

ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Учебник для 10 классов средних и средне-специальных,
профессиональных учебных заведений

Утвержден Министерством народного образования
Республики Узбекистан

Ташкент – 2017

УДК 004(075.32)

ББК 32.81я722

Т 14

Тайлаков Норбек Исакулович
Ахмедов Акром Бурхонович
Пардаева Мехринисо Данияровна
Абдуганиев Абдували Абдулхаевич
Мирсанов Уралбой Мухаммадиевич

Информатика и информационные технологии. Учебник для 10-х классов средних и средне-специальных профессиональных учебных заведений. – Ташкент:Extremum-press. 2017. – 160.

Под общей редакцией доктора педагогических наук, профессора Н.И.Тайлакова.

Рецензенты: А.Абдукодиров – доктор педагогических наук, профессор кафедры “Методика преподавания информатики” Ташкентского педагогического университета;

С.Жуманазаров – кандидат технических наук, доцент, заведующий кафедрой ”Преподавание естественных и точных наук” ТИПК и ПРНО;

С.Ештемиров – кандидат физико-математических наук, доцент кафедры “Информатика” Самаркандского государственного университета;

Б.Каримов – учитель высшей категории средней общеобразовательной школы №90 города Ташкента, преподаватель информатики и информационных технологий.

Условные обозначения:



– начало урока;



– запомните;



– вопросы и задания;



– домашнее задание.

Утверждено Министерством народного образования Республики
Узбекистан

Издано за счёт средств целевого книжного фонда республики.

ISBN: 978-9943-4860-1-0

© Extremum-press

ПРЕДИСЛОВИЕ

Дорогие учащиеся!

Настоящий учебник создан на основе государственного образовательного стандарта общего среднего образования, утверждённого Постановлением Кабинета Министров Республики Узбекистан "Об утверждении государственных образовательных стандартов общего среднего и средне-специального и профессионального образования" №187 от 6 апреля 2017 года, а также квалификационных требований общего среднего образования по дисциплине "Информатика и информационные технологии".

Продолжая Ваше обучение по работе с пользовательскими программами Microsoft Office, учебник научит Вас действиям с формулами MS Excel 2010, а также ознакомит с библиотекой функций и выполнением действий, используя логические и математические методы.

Во второй главе учебника приводятся сведения о создании базы данных MS Access 2010, её системе управления, в частности, о соединении в ней таблиц, о поиске данных по известному шаблону и приёмах их переработки.

Учебник ознакомит Вас с основами создания приложений в программной среде Delphi. В частности, Вы будете осваивать возможности создания окна приложений, размещения в нём готовых объектов управления и создания интерактивных приложений.

Таким образом, усвоив материал учебника, Вы постоянно будете стремиться развивать не только интеллектуальный, но духовный потенциал, самостоятельно повышая познавательные навыки и оценивая свои действия.

Авторы

ГЛАВА I. ДЕЙСТВИЯ И ФОРМУЛЫ В MS EXCEL 2010

ВНИМАТЕЛЬНО ПРОЧИТАВ ЭТУ ГЛАВУ, ВЫ СМОЖЕТЕ ОВЛАДЕТЬ СЛЕДУЮЩИМИ ЗНАНИЯМИ, УМЕНИЯМИ И КОМПЕТЕНЦИЯМИ:

- Ознакомиться с возможностями электронных таблиц Microsoft Excel 2010;
- Выполнять различные действия, вычислять выражения;
- Ссылки (относительная, абсолютная и смешанная) к ячейкам;
- С помощью разных ссылок пользоваться преимуществами копирования при выполнении действий;
- Строить графики простых и сложных функций;
- Обращаться к другому листу или книге;
- Пользоваться библиотекой функций;
- Пользоваться окном аргумента функций и строкой формул;
- Пользоваться возможностями автоотбора при написании имени функции и размещении;
- Владеть сведениями о приёмах решения примеров и задач, логических, математических и статистических функций.

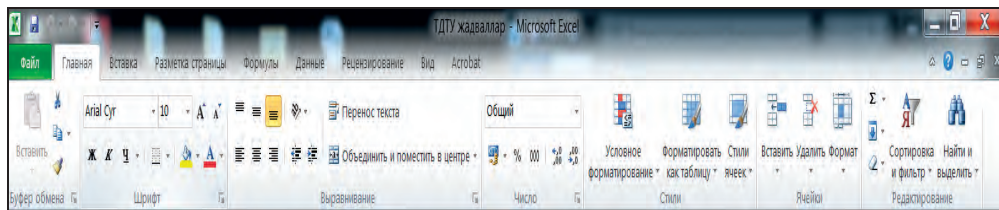




1 УРОК.

ВЫЧИСЛЕНИЕ ПРОСТЫХ ВЫРАЖЕНИЙ

В электронной таблице MS Excel 2010 в верхней части основного окна на основе нового интерфейса расположена многостраничная область, называемый **лентой** (на английском языке “Ribbon”)



В каждой части ленты MS Excel 2010 расположены кнопки, нацеленные на выполнение определённых задач:

- **Главная** – предназначена для ввода и редактирования табличных данных;
- **Вставка** – для размещения в таблицу объектов, (рисунков и диаграмм);
- **Разметка страницы** – для обозначения границ и других параметров страниц при распечатывании листов графиков;
- **Формулы** – предназначена для использования различных формул при осуществлении расчётов в таблицах;
- **Данные** – предназначена для фильтрации, отбора данных, создания и удаления копий, введения новой информации в столбцах и строках таблиц;
- **Рецензирование** – для исправления ошибок в тексте и редактирования документов;
- **Вид** – для удобного размещения таблиц на экране с целью различных демонстраций.

В верхнем левом углу главного окна имеется часто используемая и облегчающая обращение панель, в которой можно быстро и легко производить действия по сохранению в памяти и отмене последних операций.

Возможности этой панели можно расширить, если добавить к ней новые функции.

Внизу ленты расположена рабочая область электронной таблицы.

Файл, заданный в виде таблицы, в MS Excel 2010 именуется **книгой**. Книга же образована из **листов**. В стандартном состоянии их три, при необходимости количество может быть увеличено либо уменьшено.

Стандартный лист MS Excel 2010 представляет собой 16384 столбцов буквенного обозначения, начиная с буквы А и заканчивая XFD, и 1 048 576 строк числового обозначения. Каждая отдельно взятая ячейка листа имеет свой адрес: **A3, C45**.

Активизация рассматриваемой ячейки осуществляется нажатием левой кнопки мыши на ней. Внизу таблицы имеется строка состояния, в которой отображаются сведения о рабочей области.

После завершения ввода введения в таблицу значений, помощи мыши (например, **B2:F5**) можно обозначить **блок ячеек**.

Упражнение 1. Введя в ячейку **B1** число 231, в ячейку **B2** число 569, получите их сумму в **B3**.

Выполнение: Посредством написания знака “=” в ячейку **B3**, MS Excel 2010 переходит в режим ввода данных. Затем, при нажатии левой кнопки мыши на ячейку **B1**, данный адрес фиксируется в **B3**.

Следующий шаг- вводится “+”, затем также нажимается левая кнопка мышки на ячейку **B2**. В результате в **B3** появится запись **=B1+B2**. После нажатия кнопки **Enter** результат выполненного действия появляется в **B3**, в строке формул будем иметь **=B1+B2**.

Другие действия, присущие MS Excel 2010, будут рассмотрены в качестве упражнений в соответствующих главах.

В процессе работы в MS Excel 2010 могут возникнуть различные ошибки.

Остановимся на характеристике часто встречающихся ошибок:

- ##### – данные не поместились в клетку.
- #ДЕЛ/0 – в формуле расчётов имеются случаи деления на 0.
- #ЗНАЧ! – в формуле содержатся невозможные выражения, например: применяются переменные на кириллице.

- **#ИМЯ?** – невозможно распознать названия переменных в формуле MS Excel.
- **#ЧИСЛО!** – неправильное применение переменных или значений в формуле.
- **#ССЫЛКА!** – неправильное обращение к ячейке.
- **#ПУСТО!** – значение в ячейке введено в неправильном промежутке.

Упражнение 2. Выполните следующие действия:

- В ячейку **A1** введите число 23, в **B1** 0, в ячейке **C1** выполните действие **A1/B1** и проанализируйте результат.
- В ячейку **A1** введите “Ошибочный” текст, в **A2** 10 и в ячейке **A3** произведите действие **A1*A2** и выясните причину полученной ошибки.
- В ячейку **A1** введите цифру 1, в **B1** 1 и в ячейке **A3** произведите действие **A1A/B1** и выясните причину полученной ошибки.

C1		fx =A1A/B1										
	A	B	C	D	E	F	G	H	I	J	K	
1		1	#ИМЯ?									
2												

- В ячейку **A1** введите цифру 1, в **B1** 2, в **C1** посчитайте **A1+B1**, в ячейку **D1** введите 15, в **E1** введите 12, в **F1** посчитайте **D1–E1** и в **G1** посчитайте **C1*F1**;

G1		fx =C1*F1										
	A	B	C	D	E	F	G	H				
1	1	2	3	15	12	3	9					
2												

Как видно из примеров, при помощи MS Excel 2010 можно быстро и легко осуществлять произвольные арифметические вычисления.

В отличие от калькулятора электронная таблица MS Excel 2010 располагает возможностями обобщения при вычислении расширенных выражений.

Эти возможности мы рассмотрим на следующих уроках.



ЗАПОМНИТЕ !

Стандартный лист MS Excel 2010 представляет собой 16384 столбцов буквенного обозначения, и 1 048 576 строк числового обозначения.



ВОПРОСЫ И ЗАДАНИЯ

1. Какие виды данных имеются в таблицах MS Excel?
2. Какая ошибка содержится в записи **A2D:H9** ?
3. Посчитайте сумму данных трёх чисел.
4. Посчитайте произведение данных двух чисел и результат возведите в четвертую степень.



ДОМАШНЕЕ ЗАДАНИЕ

1. С помощью электронной таблицы MS Excel 2010 посчитайте переменные выражения $(a+b) \cdot (c+d)$ в произвольных значениях.
2. Найдите частное двух пар чисел, образованных путём выделения из четырёх чисел и взаимно умножьте полученные результаты.




2 УРОК. ССЫЛКИ К ЯЧЕЙКАМ: ОТНОСИТЕЛЬНАЯ, АБСОЛЮТНАЯ И СМЕШАННАЯ ССЫЛКИ

С помощью электронных таблиц Microsoft Excel 2010, кроме арифметических действий, посредством различных ссылок к ячейкам можно качественно решать практические задачи.

Упражнение 1. Определение в классах показателей успеваемости по предметам.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Определение в классах показателей успеваемости по предметам											
2	Классы	Число учеников	Предметы				Показатель усвоения					
3			Родной язык	Математика	Физика	Английский язык	Химия	Родной язык	Математика	Физика	Английский язык	Химия
4	10-а	32	27	22	25	30	27					
5	10-б	33	29	26	27	24	28					
6	10-с	31	23	25	28	21	26					
7	Всего	96	79	73	80	75	81					

Выполнение:

С помощью кнопки  Объединить и поместить в центре ▾ в первой строки все ячейки объединяются и вводится текст «**Определение в классах показателей успеваемости по предметам**».

- Объединяются ячейки **A2:A3**, и надпись “**Классы**” размещается в центр ячейки;

- Объединяются ячейки **B2:B3**, и надпись “**Число учащихся**” размещается в центр ячейки;

- Объединяются ячейки **C2:G2**, и надпись “**Предметы**” размещается в центр ячейки;

- Объединяются ячейки **H2:L2**, и надпись “**Показатель успеваемости**” размещается в центр ячейки;

- В ячейки **C3:G3** и **H3:L3** размещаются названия предметов;

- В ячейки **B4:B6** вводится количество учащихся в классах;

- С **C4:C6** по **G4:G6** вводится количество успевающих учащихся;

- **B4:B6** выделяется левой кнопкой мыши, в ленте MS Excel 2010 нажимается клавиша **Σ**, в результате в **B7** выводится общее количество учащихся;

- Для облегчения определения общего количества успевающих учащихся по всем предметам в ячейках **C7:G7**, нажимая на правый нижний угол ячейки **B7** левой кнопкой мыши, и, смещая его, получаем нужные значения. Метод данной ссылки к рассматриваемой ячейке называется **относительной ссылкой**;

- Для определения показателя успеваемости по родному языку в **10-а** классе в ячейку **H4** вводим “=” и, поделив количество успевающих учащихся **C4** на общее количество учащихся в классе **B4**, умножаем результат на 100 и округляем. По причине того, что показатель успеваемости всегда делится на количество учащихся в классе, для того чтобы **B4** сделать постоянным, ссылка в данный адрес делается в виде **\$B\$4**, то есть в виде **абсолютной ссылки**;

- Для облегчения определения показателей успеваемости в других классах и по другим предметам нажимаем на левую кнопку мыши, перемещая по столбцу **B** и 7 строке.

Таким образом, используя относительные и абсолютные ссылки, рассматриваемая задача легко решилась. В результате получилась следующая электронная таблица:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Определение в классах показателей успеваемости по предметам											
2			Предметы					Показатель усвоения				
3	Классы	Число учащихся	Родной язык	Математика	Физика	Английский язык	Химия	Родной язык	Математика	Физика	Английский язык	Химия
4	10-а	32	27	22	25	30	27	84,4	68,8	78,1	93,8	84,4
5	10-б	33	29	26	27	24	28	87,9	78,8	81,8	72,7	84,8
6	10-в	31	23	25	28	21	26	74,2	80,6	90,3	67,7	83,9
7	Всего	96	79	73	80	75	81	82,3	76	83,3	78,1	84,4

Если относительная и абсолютная ссылки идут совместно, получается **смешанная ссылка**.

Для лучшего обеспечения другими альтернативными практическими программами MS Excel 2010 для сохранения документов в памяти разработан новый формат XML (eXtensible Markup Language – расширенный язык обозначения). За счет применения технологии сжатия уменьшаются параметры файлов и появляется возможность использования данных таблиц в любой операционной системе.



ЗАПОМНИТЕ !

Файлы в MS Excel 2010 имеют расширение “.xlsx”



ВОПРОСЫ И ЗАДАНИЯ

1. Объясните разницу между абсолютной и относительной ссылками к ячейке.
2. Какая новая технология применяется в MS Excel 2010 для хранения в памяти электронных таблиц?
3. С помощью электронной таблицы MS Excel 2010 подготовьте справку для мониторинга успеваемости вашего класса по предметам.



ДОМАШНЕЕ ЗАДАНИЕ

1. Составьте задачу по абсолютной и относительной ссылке и решите ее.
2. Создайте таблицу, определяющую посещаемость учащихся класса.



3 УРОК. ПРЕИМУЩЕСТВО КОПИРОВАНИЯ ПРИ ВЫПОЛНЕНИИ ДЕЙСТВИЙ С ИСПОЛЬЗОВАНИЕМ ССЫЛОК

При выполнении действий с использованием ссылок посредством копирования появляется возможность облегчить выполняемые действия. В следующих упражнениях рассмотрим преимущество копирования при использовании ссылок.

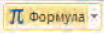
Упражнение 1. Расчет числа a до 9-степени.

Выполнение:

- Создаем таблицу из девяти столбцов и шести строк;

▪ Ячейки второй строки объединяются с помощью кнопки  и вводится текст “**Расчет степеней данного числа**”;

▪ В столбец **A** вводятся значения числа **a** ;

Выбрав последовательность **Вставка - вставка формулу**  MS Excel 2010, впишем следующие значения **a**:

	A	B	C	D	E	F	G	H	I
1									
2	Расчет степеней данного числа								
3	a	a ²	a ³	a ⁴	a ⁵	a ⁶	a ⁷	a ⁸	a ⁹
4	2								
5	3								
6	4								
7	5								

▪ В ячейке **B4** выполняется действие возведения в квадрат данного числа $=\$A4*A4$. Подводим курсор мыши на правый нижний угол данной ячейки, держим левую кнопку нажатой, по столбцу передвигаем вниз и останавливаемся в ячейке **B7**;

▪ Подводим курсор мыши на правый нижний угол ячейки **B7**, держим левую кнопку нажатой, передвигаем в промежутке **B7:I7**. В результате получается следующее:

	A	B	C	D	E	F	G	H	I
1									
2	Расчет степеней данного числа								
3	a	a ²	a ³	a ⁴	a ⁵	a ⁶	a ⁷	a ⁸	a ⁹
4	2	4	8	16	32	64	128	256	512
5	3	9	27	81	243	729	2187	6561	19683
6	4	16	64	256	1024	4096	16384	65536	262144
7	5	25	125	625	3125	15625	78125	390625	1953125



ЗАПОМНИТЕ !

Если вводимый текст в MS Excel 2010 не помещается в ячейках, используется кнопка **Перенос текста**.



ВОПРОСЫ И ЗАДАНИЯ

1. Объясните особенности копий электронных таблиц MS Excel 2010.
2. Как переносится текст в электронной таблице MS Excel 2010?
3. Создайте электронную таблицу для возведения числа 3 в 10 степень.



ДОМАШНЕЕ ЗАДАНИЕ

1. Создайте таблицу закупок учебных принадлежностей.
2. Вычислите выражение $1+2^2+3^3+4^4$ в MS Excel 2010.



4 УРОК. ГРАФИКИ ПРОСТЫХ И СЛОЖНЫХ ФУНКЦИЙ

В MS Excel 2010 графики простых и сложных функций выполняются посредством пункта **График** раздела **Вставка** меню. Процесс создания графиков простых и сложных функций рассмотрим в следующих упражнениях.

Упражнение 1. Вывод графика функции $z=a+bx+cx^2+dx^3$

Выполнение:

▪ В ячейки **A1:D1** вводятся следующие значения коэффициента **a,b,c,d**:

	A	B	C	D
1	-1	5	-35	65

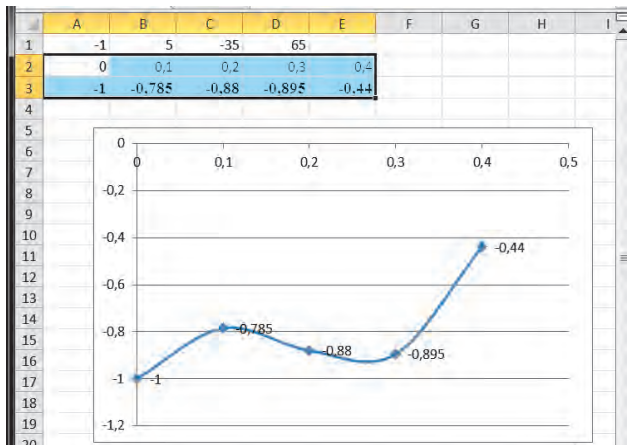
▪ В ячейки **A2:E2** вводятся значения $x=0; 0,1; 0,2; 0,3; 0,4$;

▪ В ячейках **A3:E3** соответственно значениям x , к примеру в **A3** для рассматриваемой формулы $=\$A1+\$B1*A2+\$C1*A2*A2+\$D1*A2*A2*A2$ получится значение функции при $x=0$. Для остальных точек – процесс продолжается с помощью относительной ссылки;

▪ Выделяется блок ячеек **A2:E3**;

После выделения ячеек **A2:E3** выполняется последовательность **Вставить-Диаграммы-Точечная** из ленты MS Excel 2010.

Для вывода значений x на полученный график над ним нажимается правая кнопка, выбирается пункт **Добавить подписи данных**. В результате видно следующее окно:



Упражнение 2. Получение графиков функций $z=a+bx+cy$ и $z=ax^2+bxу+cy^2$.

Выполнение:

- В ячейки **A1:C1** вводятся значения коэффициентов a, b, c

Для чертежа слева

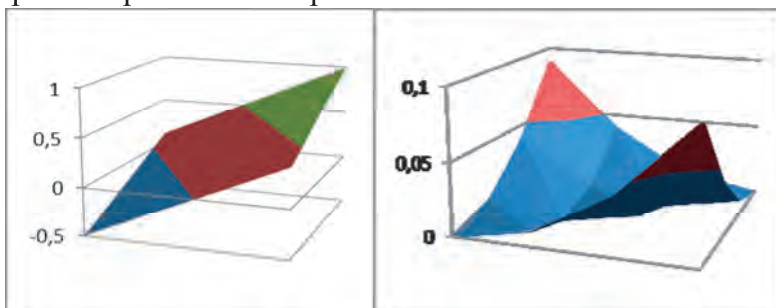
	A	B	C
1	-1	2	3

Для чертежа справа

	A	B	C
1	1	-2	1

- В ячейки **A2:E2** вводятся значения $x=0; 0,1; 0,2; 0,3; 0,4$;
- В ячейки **A2:A6** вводятся значения $y=0; 0,1; 0,2; 0,3; 0,4$;
- В ячейках **A3:E7** соответственно значениям x , к примеру в **A4** для рассматриваемой формулы $=\$A1*A2*A2+\$B1*A2*A4+\$C1*A4*A4$ получится значение функции при $x=0, y=0,1$. Для остальных точек процесс продолжается с помощью **относительной** ссылки; Здесь специальный знак **\$** означает **абсолютную** (фиксированную) ссылку к данной ячейке.

- После выделения ячеек **A3:E7** в панели MS Excel 2010 в «ленте» раздела **Вставить**, нажав на клавишу **Поверхность**, получаем чертежи рассматриваемых выражений:



ЗАПОМНИТЕ!

Образование графиков функции в MS Excel 2010 осуществляется в пункте **Диаграммы** раздела **Вставить**.



ВОПРОСЫ И ЗАДАНИЯ

1. Расскажите о возможностях составления графика в MS Excel 2010.

2. Какой метод применяется для сглаживания графиков функции?
3. Составьте в виде графика показатели успеваемости учащихся вашего класса по предмету «Информатика и информационные технологии».



ДОМАШНЕЕ ЗАДАНИЕ

1. Составьте график функции $y=3x+4$.
2. Составьте в виде графика показатели успеваемости учащихся вашего класса по предмету «Математика».



5 УРОК. ССЫЛКА НА ДРУГОЙ ЛИСТ ИЛИ КНИГУ

Еще одно из удобств MS Excel 2010 заключается в том, что сделав ссылку на другой лист или книгу, можно связаться с их данными. Методы ссылки на лист или книгу мы рассмотрим на примере следующих упражнений.

Упражнение 1. Создание обобщенной таблицы по продаже оргтехники согласно данным торговых точек.

Выполнение:

- Вводим данные по каждой торговой точке:

1-й маркет – 1 лист

	A	B	C	D
1	Продажа оргтехники Маркет 1			
2	Наименование товара	Стоимость	Число	Сумма
3	монитор	200	12	2400
4	процессор	60	31	1860
5	клавиатура	25	28	700
6	винчестер	95	14	1330
7	Всего			6290

2-й маркет – 2 лист

	A	B	C	D
1	Продажа оргтехники Маркет 2			
2	Наименование товара	Стоимость	Число	Сумма
3	монитор	200	15	3000
4	процессор	60	25	1500
5	клавиатура	25	35	875
6	винчестер	95	13	1235
7	Всего			6610

3-й маркет – 3 лист

	A	B	C	D
1	Продажа оргтехники Маркет 3			
2	Наименование товара	Стоимость	Число	Сумма
3	монитор	200	22	4400
4	процессор	60	20	1200
5	клавиатура	25	21	525
6	винчестер	95	23	2185
7	Всего			8310

- Для получения обобщенной информации на 4 листе в ячейке В3 создается надпись =Лист1!В3+Лист2!В3+Лист3!В3. В ячейку первого

листа **B3** после знака “=” нажимается левая кнопка мыши, в результате **Лист1!B3**, затем + и из 2- листа получается **Лист2!B3** и с 4 листа копирование с ячеек **B4, B5, B6, C3, C4, C5, C6, D3, D4, D5, D6, D7** осуществляется нажатием левой кнопки мыши на правый нижний угол. В результате получается следующее окно:

	A	B	C	D
1	Продажа оргтехники фирм			
2	Наименование товара	Стоимость	Число	Сумма
3	монитор	200	49	9800
4	процессор	60	76	4560
5	клавиатура	25	84	2100
6	винчестер	95	50	4750
7	Всего			21210

Упражнение 2. Получение оценок учащимися по предметам с помощью книги таблиц.

Выполнение:

■ Для составления книги таблиц оценок учащихся по предметам по каждому предмету составляется отдельная рабочая книга (математика, физика, информатика):

	A	B	C		A	B	C
1	П/н	Ф.И.О. учеников	Математика	1	П/н	Ф.И.О. учеников	Физика
2	1	Алиев Алишер Анвар уг'ли	3	2	1	Алиев Алишер Анвар уг'ли	5
3	2	Мирсанов Уралбой Мухаммад уг'ли	5	3	2	Мирсанов Уралбой Мухаммад уг'ли	5
4	3	Ходжабоев Фарход Давлатбой уг'ли	5	4	3	Ходжабоев Фарход Давлатбой уг'ли	5
5	4	Давлатбоев Бехруз Фарход уг'ли	5	5	4	Давлатбоев Бехруз Фарход уг'ли	5
6	5	Мухаммадов Асадбек Урол уг'ли	5	6	5	Мухаммадов Асадбек Урол уг'ли	5

	A	B	C
1	П/н	Ф.И.О. учеников	Информатика
2	1	Алиев Алишер Анвар уг'ли	4
3	2	Мирсанов Уралбой Мухаммад уг'ли	5
4	3	Ходжабоев Фарход Давлатбой уг'ли	5
5	4	Давлатбоев Бехруз Фарход уг'ли	5
6	5	Мухаммадов Асадбек Урол уг'ли	5

Три данные таблицы объединяются в отдельную таблицу, отображающую оценки учащихся в рабочей книге **общих предметов**:

	A	B	C	D	E
1	П/н	Ф.И.О. учеников	Математика	Физика	Информатика
2	1	Алиев Алишер Анвар уг'ли			
3	2	Мирсанов Уралбой Мухаммад уг'ли			
4	3	Ходжабоев Фарход Давлатбой уг'ли			
5	4	Давлатбоев Бехруз Фарход уг'ли			
6	5	Мухаммадов Асадбек Урол уг'ли			

■ Открывается рабочая книга, в которую введён отдельный предмет(файл таблиц общих предметов), выделяются и копируются ячейки, в которые введены оценки (Копировать). Выделяются соответствующие ячейки предметов в книге общих предметов, выбирается функция специальной вставки (**Специальная вставка**) и нажимается клавиша вставки связи (**Вставить связь**) :

	A	B	C
1	П/н	Ф.И.О. учеников	Математика
2	1	Алиев Алишер Анвар уг'ли	
3	2	Мирсанов Уралбой Мухаммад уг'ли	
4	3	Ходжабоев Фарход Давлатбой уг'ли	
5	4	Давлатбоев Бехруз Фарход уг'ли	
6	5	Мухаммадов Асадбек Урол уг'ли	
7	6	Икромова Нафиса Жасур кизи	
8	7	Ражабова Гулхайо Даврон кизи	
9	8	Маликов Жахонгир Гайрат уг'ли	
10	9	Мирзохидова Салима Комилжон кизи	
11	10	Рахимова Анора Акбар кизи	
12			
13			
14			
15			

Специальная вставка

Вставить

все с исходной темой

формулы без рамки

значения ширины столбцов

форматы формулы и форматы чисел

примечания значения и форматы чисел

условия на значения

Операция

цвет умножить

сложить разделить

вычесть

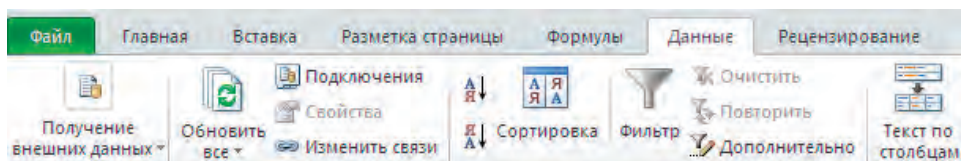
пропускать пустые ячейки транспонировать

Вставить связь OK Отмена

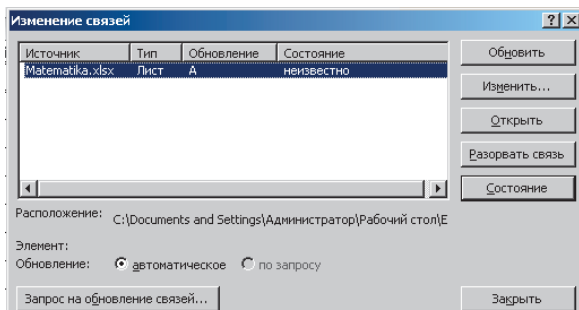
В результате получается следующая связь:

	A	B	C	D	E
1	П/н	Ф.И.О. учеников	Математика	Физика	Информатика
2	1	Алиев Алишер Анвар уг'ли	3		
3	2	Мирсанов Уралбой Мухаммад уг'ли	5		
4	3	Ходжабоев Фарход Давлатбой уг'ли	5		
5	4	Давлатбоев Бехруз Фарход уг'ли	5		
6	5	Мухаммадов Асадбек Урол уг'ли	5		
7	6	Икромова Нафиса Жасур кизи			
8	7	Ражабова Гулхайо Даврон кизи			
9	8	Маликов Жахонгир Гайрат уг'ли			
10	9	Мирзохидова Салима Комилжон кизи			
11	10	Рахимова Анора Акбар кизи			

▪ После получения связи между всеми рабочими книгами из строки рабочей книги полученных связей выбирается раздел **Данные** и нажимается последовательность **Подключения – изменить связи** :



▪ Из полученного окна выбирается клавиша **Обновить**, для окончания операции нажимается клавиша **Закреть** и совершается выход из данного окна.



ЗАПОМНИТЕ!

Для связывания книг из строки меню выбирается раздел **Данные** и **Подключения – изменить связи** и нажимается последовательность **Подключения – изменить связи**.



ВОПРОСЫ И ЗАДАНИЯ

1. Разъясните методы связывания листов.
2. Разъясните методы связывания книг.
3. Составьте отдельные таблицы для оценок учащихся 10 класса по всем предметам и объедините их в одну таблицу.



ДОМАШНЕЕ ЗАДАНИЕ

1. В ячейке третьего листа **B4** вычислите сумму чисел ячейки **C3** первого листа и ячейки **D6** второго листа.
2. Оценки учащихся по предметам история, химия, физика и математика сохраните в отдельных файлах и объедините в файле «Общий».



6 УРОК. БИБЛИОТЕКА ФУНКЦИЙ MS EXCEL

Выше мы рассказали о том, что информационные таблицы MS Excel 2010 заполняются текстом или числами. Иногда возникает необходимость в выполнении расчётов по данным ячеек, в таких случаях используются формулы.

Для этого делается ссылка в раздел **Формулы** ленты MS Excel 2010. В данном разделе собраны значения, выражения и библиотека функций, использующиеся в выполнении действий над ячейками. Рассмотрим некоторые функции, предназначенные для выполнения различных задач в библиотеке.

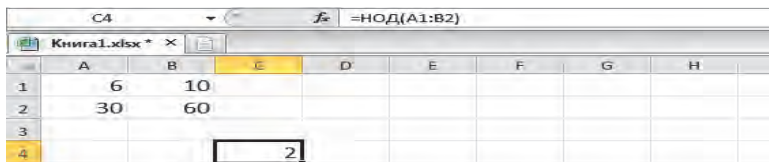
Математические функции

t/r	Функция	Предназначение
1.	ФАКТР	Рассчитывает факториал целого числа, данного в качестве аргумента
2.	ABS	Рассчитывает модуль значения аргумента числа.
3.	LN	Рассчитывает натуральный логарифм числа.
4.	EXP	Рассчитывает экспоненту числа.
5.	SIN	Рассчитывает синус числа.
6.	COS	Рассчитывает косинус числа.
7.	TAN	Рассчитывает тангенс числа.
8.	НОД	Рассчитывает наибольшим делитель среди двух или большего количества целых чисел.
9.	НОК	Рассчитывает наименьший делитель среди двух или большего количества целых чисел.
10.	ОКРУГЛ	Округляет натуральное число.
11.	СУММ	Рассчитывает сумму двух или более чисел.
12.	ПРОИЗВЕД	Рассчитывает произведение двух или более чисел.
13.	ЗНАК	Выводит -1, если число отрицательное, -если число положительное, 0, если число – 0.
14.	КОРЕНЬ	Рассчитывает квадратный корень числа.

Упражнение 1. Рассчитайте наибольший общий делитель данных чисел: 6, 10, 30, 60.

Выполнение:

- Вводим в ячейку **A1** число 6, в ячейку **B1** число 10, в ячейку **A2** число 30, в ячейку **B2** число 60;
- Вводим в ячейку **C4** следующую формулу: **=НОД(A1:B2)**
- Нажимается клавиша **Enter**. В результате появится следующее окно:



	A	B	C	D	E	F	G	H
1	6	10						
2	30	60						
3								
4			2					



ЗАПОМНИТЕ !

В MS Excel 2010 *алгебраические знаки действий* следующие:
+ (сложение); - (вычитание); * (умножение); / (деление); ^ (возведение в степень).



ВОПРОСЫ И ЗАДАНИЯ

1. Используя функцию **СУММ ПРОИЗ**, вычислите сумму произведения следующих пар чисел (5;10), (15;20), (25;30).
2. Из данных чисел -99, -23, -6, 0, 10, 56, 77, 89, 123, 345, 678, 2345 вычислите сумму и произведение самого большого и малого числа.
3. Вычислите произведение чисел -7,8; -9,67865; -4,6; -3,8; 7,8 и с помощью функции **ОКРУГЛ** округлите до 2 цифр после запятой.



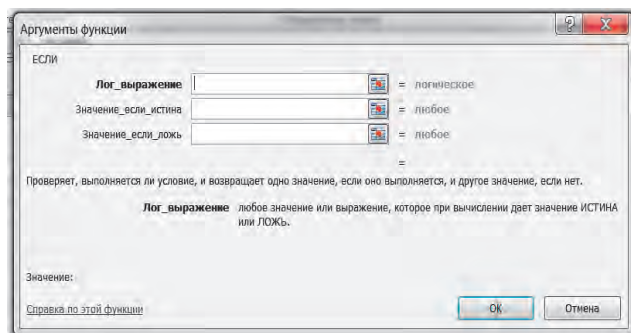
ДОМАШНЕЕ ЗАДАНИЕ

1. Вычислите сумму квадратных корней чисел 1, 2, 3, 4, 5, 6, 8.
2. Найдите наименьшее общее делимое чисел 15, 20, 45, 62, 94, 112.
3. Вычислите факториалы чисел 4, 10.



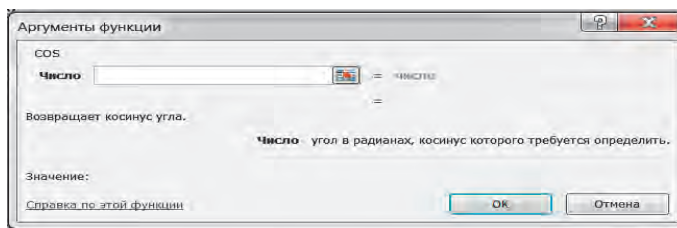
7 УРОК. ОКНО АРГУМЕНТА ФУНКЦИИ

Ввиду того, что используемые в MS Excel 2010 функции отличаются друг от друга по содержанию, их аргументы также, в свою



очередь, могут быть даны в различных видах.

К примеру, в разделе **Математические функции** в функции **cos(x)**

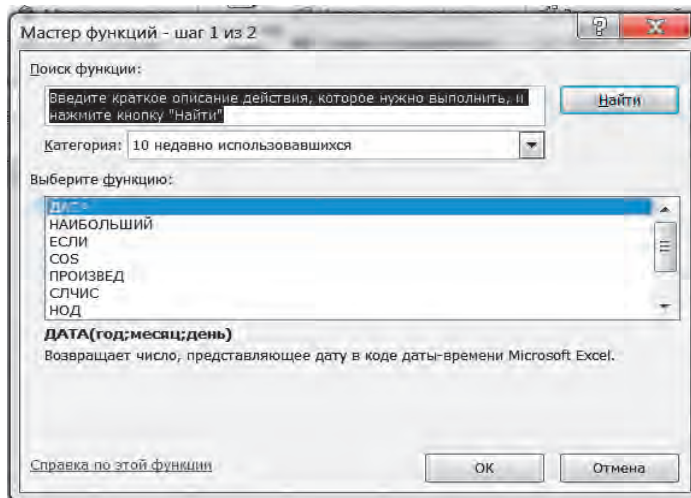


аргумент состоит из одного действительного значения и угловое значение дается в радианах:


В функции **ЕСЛИ** раздела **Логические функции** аргумент является логическим выражением, судя по двум его значениям («истина» или «ложь») можем иметь два вида результата:

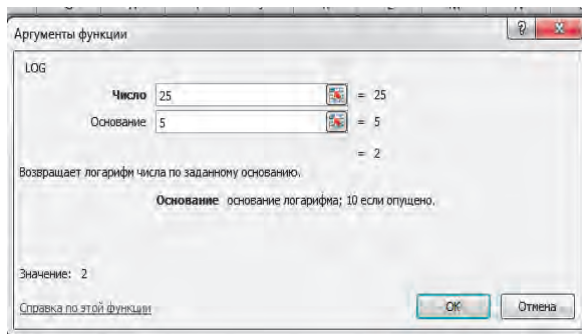
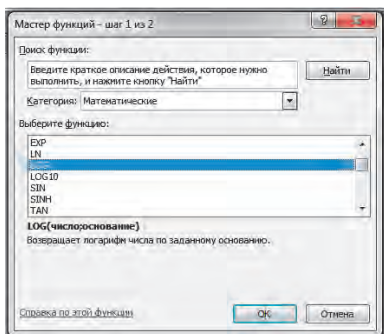
Шаг первый– в окно **Мастер функций** вводится нужная функция, с помощью клавиши **Найти** находится нужное, второй шаг – обращается к данной функции и даются значения к его аргументам. Затем выбранная функция размещается в нужную ячейку:

Упражнение 1. Вычислите значение выражения $\log_5 25$.




Выполнение:

- Из строки формул MS Excel 2010 выбирается кнопка ;
- Из окна **Мастер функций** выбирается функция расчёта логарифма **Log**;
- В ряд **Число** вводится число 25, а в строку **Основание** число 5 и нажимается кнопка **ОК**:



ЗАПОМНИТЕ!

Для пользования функциями в MS Excel 2010 из строки формул выбирается кнопка .



ВОПРОСЫ И ЗАДАНИЯ

1. Какую задачу осуществляет **Мастер функций**?
2. Дайте информацию об аргументах функций?
3. Вычислите сумму натуральных логарифмов чисел 5, 10, 15, 20, 25, 30 и округлите результат до 3 цифр.
4. Используя **Мастер функций**, вычислите экспоненты чисел 4, 6, 10, 23.
5. Используя **Мастер функций**, вычислите наименьшую общую кратное для чисел 15, 20, 45, 62, 94, 112.



ДОМАШНЕЕ ЗАДАНИЕ

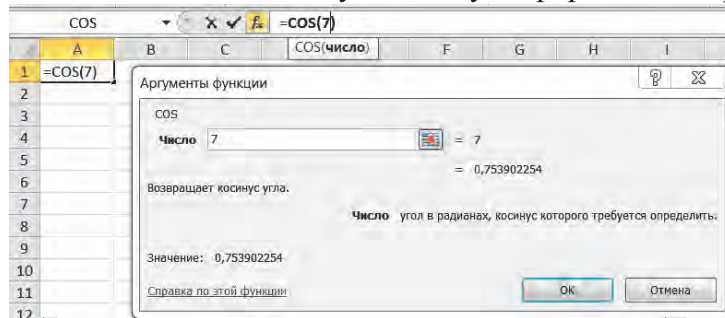
1. Используя **Мастер функций**, вычислите наибольший общий делитель для чисел 15, 20, 45, 62, 94, 112.
2. Используя **Мастер функций**, вычислите факториал для числа 10.
3. Используя **Мастер функций**, вычислите синус 2,4567.



8 УРОК.

ПОЛЬЗОВАНИЕ СТРОКОЙ ФОРМУЛ

Строка формул предназначена для просмотра формул значений текущей ячейки или ввода в данную ячейку информацию и формулы:



Кнопка - в окне обозначает отклонение, кнопка - выполнение действий, кнопка - выполняет ссылку к библиотеке.

Упражнение 1. Составьте следующую информационную таблицу. Выполните расчёты, используя строку формул. **Расходы предприятия на командировки (в тыс. сум):**

	В	С	Д	Е	Ф	Г
1	П/н	Место поездки	Стоимость дороги	Количество дней	Число людей	Всего израсходовано
2	1.					
3	2.					
4	3.					
5	4.					

Для данной задачи пункт «Все расходы» вычисляется с помощью следующей формулы:

$$\text{“Все расходы”} = (2 * \text{“Стоимость проезда”} + \text{“Количество дней”} * \text{“Ежедневный расход”}) * \text{“Число людей”}$$

Выполнение:

- Если ширина столбцов и строк недостаточна, расширяем их до необходимой величины;
- Заполняем ячейки необходимой информацией;

	А	В	С	Д	Е	Ф	Г
1	П/н	Место поездки	Стоимость дороги	Количество дней	Ежедневный расход	Число людей	Всего израсходовано
2	1.	Ташкент	80	5	80	4	
3	2.	Бухара	86	4	50	5	
4	3.	Киев	1860	12	160	4	
5	4.	Москва	1780	10	400	6	

▪ Из раздела **Файл** выбираем пункт **Сохранить как** и в строку **Имя файла** вводим название файла, например, “comras.xlsx” и нажимаем кнопку **Сохранить**.

▪ Для загрузки таблицы из диска в разделе **Файл** выбираем пункт **Открыть**. Из списка **Файл** выбираем необходимый документ (например, “comras.xlsx”), нажимая кнопку мыши;

- В ячейку **G2** вводим следующую формулу: $= (2 * C2 + D2 * E2) * F2$
- Для остальных строк также вводятся схожие формулы:

	А	В	С	Д	Е	Ф	Г
1	П/н	Место поездки	Стоимость дороги	Количество дней	Ежедневный расход	Число людей	Всего израсходовано
2	1.	Ташкент	80	5	80	4	$= (2 * C2 + D2 * E2) * F2$
3	2.	Бухара	86	4	50	5	$= (2 * C3 + D3 * E3) * F3$
4	3.	Киев	1860	12	160	4	$= (2 * C4 + D4 * E4) * F4$
5	4.	Москва	1780	10	400	6	$= (2 * C5 + D5 * E5) * F5$

В результате получаем следующую таблицу:

	A	B	C	D	E	F	G
1	Пн	Место поездки	Стоимость дороги	Количество дней	Ежедневный расход	Число людей	Всего израсходовано
2	1	Ташкент	80	5	80	4	2240
3	2	Бухара	86	4	50	5	1860
4	3	Киев	1860	12	160	4	22560
5	4	Москва	1780	10	400	6	45360



ЗАПОМНИТЕ !

Строка формул предназначена для просмотра формулы значения текущей ячейки или ввода в данную ячейку информации и формул.



ВОПРОСЫ И ЗАДАНИЯ

1. Объясните функцию строки формул.
2. Используя строку формул, вычислите сумму кубов чисел, данных в трёх ячейках.
3. В строку формул на первом листе введите выражение $= (2 * C3 + D4 * E2) * \text{Лист2!F2}$ и, обозначив нужными значениями ячейки, вычислите результат.



ДОМАШНЕЕ ЗАДАНИЕ

1. Используя строку формул, вычислите произведение данных четырёх чисел.
2. Используя строку формул, запишите формулу $= A3 - B2 * D4 + (C3 / D2)$.



9 УРОК.

ТЕКСТОВЫЕ ФУНКЦИИ

Существуют специальные функции для работы с текстами в ячейках MS Excel 2010. Они приведены ниже:

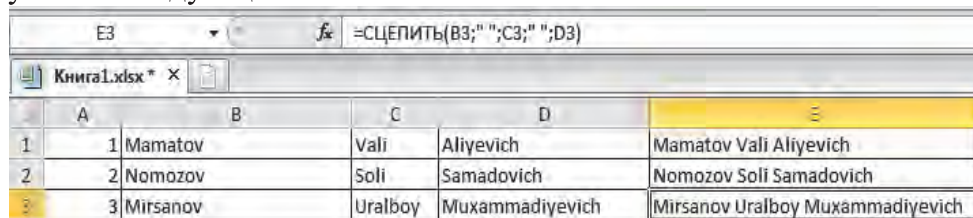
1.	Функция	Предназначение
2.	БАТТЕКСТ	Меняет число на текст.
3.	ДЛСТР	Определяет количество знаков в тексте строки.
4.	ЗАМЕНИТЬ	Заменяет выделенную часть текста на новую.
5.	ЗНАЧЕН	Преобразует число в виде текста в число.
6.	ПОВТОР	Повторяет текст в количестве раз n .
7.	СЦЕПИТЬ	Несколько текстов объединяет в один.

Упражнение 1. Объединение в одну ячейку фамилии, имени и отчества учащихся, введённых в три ячейки.

Выполнение: Существует два способа решения данной задачи.

Первый способ:

- В столбец **B** вводится фамилия ученика, в столбец **C** имя, в столбец **D** - отчество;
- В ячейку **E1** вводится формула **=СЦЕПИТЬ(B1;" ";C1;" ";D1)** и нажимается клавиша **Enter**;
- Данная формула копируется в остальные ячейки. В результате получается следующее окно:



	A	B	C	D	E
1	1	Mamatov	Vali	Aliyevich	Mamatov Vali Aliyevich
2	2	Nomozov	Soli	Samadovich	Nomozov Soli Samadovich
3	3	Mirsanov	Uralboy	Muxammadiyevich	Mirsanov Uralboy Muxammadiyevich

Второй способ:

- Заполним таблицу по выше приведённому способу;
- В ячейку **E1** вводится формула **=B1&" "&C1&" "&D1** и нажимается клавиша **Enter**.

Если фамилия ученика введена в ячейку **B1** листа **Лист1**, имя в ячейку **B1** листа **Лист2**, отчество в ячейку **B1** листа **Лист3**, для получения данной информации в листе **Лист4** в любую ячейку вводится формула **=СЦЕПИТЬ(Лист1!B1;" ";Лист2!B1;" ";Лист3!B1)** и нажимается клавиша **Enter**.



ЗАПОМНИТЕ !

В электронной таблице MS Excel выражение **B2:C5** обозначает блок ячеек. Включает в блок 8 ячеек. Если имеет вид **B2;C5**, выделяет две ячейки.



ВОПРОСЫ И ЗАДАНИЯ

1. Объясните функцию текстовых функций.
2. В столбце C даны фамилии, имена, отчества учащихся. В столбце D даны даты рождения учащихся. Объедините их в столбце E.



ДОМАШНЕЕ ЗАДАНИЕ

1. Определите количество знаков в тексте «Узбекистан – независимое

- государство» с помощью функции ДЛСТР.
2. Самостоятельно составьте и решите задачу с помощью функции **ЗАМЕНИТЬ**.



10 УРОК. ЛОГИЧЕСКИЕ ФУНКЦИИ

Вычисления при решении некоторых практических задач могут зависеть от тех или иных условий. В данной ситуации можно использовать функции с логическими условиями.

Логические функции

П/Н	Функция	Предназначение
1.	ЕСЛИ	Если значение логического выражения является ИСТИНОЙ , выполняется первое условие, иначе, то выполняется второе условие.
2.	И	Если все значения логических выражений являются истиной, значение функции является ИСТИНОЙ , в противном случае значение функции является ЛОЖЬЮ .
3.	ИЛИ	Если одно из логических выражений является истиной, значение функции является ИСТИНОЙ , в противном случае значение функции является ЛОЖЬЮ .
4.	НЕ	Обращает истинное значение в ложное и ложное в истинное.

Упражнение 1. Создание таблицы, анализирующей показатель успеваемости класса по пяти предметам по 20 балльной шкале.

Выполнение: Алгоритм выполнения данной задачи имеет следующий вид:

- В столбец **В** вводятся фамилии и имена учащихся класса;
- В столбец **С** – оценки по литературе;
- В столбец **Д** – оценки по алгебре;
- В столбец **Е** – оценки по физике;
- В столбец **Ф** – оценки по химии;
- В столбец **Г** – оценки по информатике;
- В столбец **Н** – общее количество баллов;

- В ячейки **I3** и **I5** вводится следующая формула:

$$=ЕСЛИ(С17<55;"неудовлетворительно";ЕСЛИ(С17<71;"удовлетворительно";ЕСЛИ(С17<86;"хорошо";"отлично")))$$

	A	B	C	D	E	F	G	H	I
1		Предметы							
2	П/н	Фамилия, имя	Литература	Алгебра	Физика	Химия	Информатика	Всего	Оценка
3	1	Азимов С	20	12	18	19	18	87	Отлично
4	2	Ахмедов Н	6	7	2	3	9	27	Неудовлетворительно
5	3	Вахобов А	15	10	18	18	13	74	Хорошо
6	4	Мухсумов Х	20	6	7	19	18	70	Неудовлетворительно
7	5	Нуримов Х	20	3	7	19	18	67	Неудовлетворительно



ЗАПОМНИТЕ!

Если вычисления при решении некоторых практических задач зависят от тех или иных условий, можно использовать функции с логическими условиями.



ВОПРОСЫ И ЗАДАНИЯ

1. Объясните предназначение функции **И**.
2. Объясните предназначение функции **ИЛИ**.
3. В торговой точке перед продавцами поставлены следующие условия:
 - Если торговля продавца составит объём, равный 5 размерам минимальной заработной платы, он получает 5 % премии;
 - Если торговля продавца составит объём, равный 10 размерам минимальной заработной платы, он получает 7,5 % премии;
 - Если торговля продавца составит объём, равный 20 размерам минимальной заработной платы, он получает 10 % премии;
 - Если торговля продавца составит объём свыше 20 - кратного размера минимальной заработной платы, он получает 12 % премии;
 Составьте таблицу логического решения данной задачи.



ДОМАШНЕЕ ЗАДАНИЕ

1. Составьте таблицу, анализирующую показатель успеваемости класса по десяти предметам по 10 - балльной шкале.
2. Самостоятельно составьте задачу по функции **ИЛИ** и решите её.



11 УРОК.

ПРАКТИЧЕСКИЙ УРОК ДЛЯ ЗАКРЕПЛЕНИЯ

Выполните следующие задания:

1. Расчёт командировочных расходов отделов организации (в тыс. сум.).

П/н	Отделы	Годы			Всего
		2015	2016	2017	
1.	Техническое обеспечение	131,5	14,6	152,6	298,7
2.	Бухгалтерия	141,6	112,7	114,6	368,9
3.	Отдел кадров	128,4	153,3	143,8	425,5
Всего		401,5	280,6	411	1093,1

2. Расчет налогов с заработной платы сотрудников организации (в тыс. сум.).

П/н	Фамилия, имя, отчество	Оклад	15 %ый налог	1 %ый Страховой фонд	Получение
1.	Ахадов С.	1680			
2.	Сидоров П.	1360			
3.	Усманов К.	1460			
Всего		4500			

3. Расчёт налогов с операций купли-продажи, произведённых организацией (в тыс. сум.).

П/н	Название операции	Общая сумма	1 %ное страхование	18 %ый налог на добавленную стоимость	Остаток
1.	Заработная плата	26500			
2.	Закупка	32500			
3.	Продажа	41800			
Всего		100800			

4. Определение себестоимости продукции домостроительной организации (в тыс. сум.).

П/н	Использованная продукция	Себестоимость	18 %ый налог на добавленную стоимость	Остаток
1.	Цемент	214,3		
2.	Щебень	216,4		
3.	Вода	48,9		
Всего		479,6		

5. Расчёт налогов с реализованной продукции предприятия (в тыс. сум.).

П/н	Наименование продукции	Количество	Стоимость	18 %ый налог на добавленную стоимость
1.	Кирпич	2790	0,6	
2.	Блок	6170	1,2	
3.	Двери	1070	700	
Всего		10030	701,8	



12 УРОК. КОНТРОЛЬНАЯ РАБОТА

Решите следующие задачи с использованием навыков, приобретённых в пройденных темах, полученные результаты изобразите в виде графика:

1 вариант

- Вычисление доходов физических лиц от суммы, вложенной в сберегательный банк (в тыс. сум.).

П/н	Фамилия, имя, отчество	Вложенная сумма	9 %ый доход	Всего
1.	Суяров А.	11500		
2.	Бозоров А.	9500		
3.	Сергеев А.	14400		
4.	Мустафин С.	13500		

- Вычисление общих расходов (в тыс. сум.).

П/н	Цель расхода	Сумма суточного расхода	Сумма расхода на 2 дня	Рост в %
1.	Завтрак	12	15	
2.	Обед	15	17	
3.	Ужин	14	18	
Всего				

2 вариант

1. Выполнение годовой нагрузки школьных учителей.

П/н	Фамилия, имя, отчество	План	Выполнено	В %
1.	Алимов А.	780	780	
2.	Яковлев Н.	806	798	
3.	Судейманов А.	810	804	
4.	Азимов А.	678	678	

2. Вычисление заработной платы сотрудников предприятия с учётом местного коэффициента и вредности (в тыс. сум).

п/н	Фамилия, имя, отчество	Оклад	Местный коэфф.	Вредность	Всего
1.	Назаров У.	1400	0,05	0,2	
2.	Хайдаров Р.	1300	0,05	0,2	
3.	Якушев О.	1280	0,05	0,2	
4.	Сагатшин В.	1170	0,05	0,2	
Всего					

3 вариант

1. Квартальный отчёт отдела организации (в тыс. сум.).

п/н	Квартал	Объём товара, в тоннах	Себестоимость 1 тонны	Всего
1.	I квартал	205,6	22400	
2.	II квартал	207,7	21800	
3.	III квартал	208,8	22400	
Средняя стоимость				

2. Расчёт отчислений предприятия в местный фонд с общего оборота по месяцам (в тыс. сум.).

П/н	Месяцы	Оборотная сумма	5 %ное отчисление в местный фонд
1.	Январь	12000000	
2.	Февраль	22600000	
3.	Март	32600000	

4 вариант

1. Показатели успеваемости в школе по дисциплине «Информатика и информационные технологии».

П/н	Класс	Количество учеников	Сдавшие	Успеваемость в %
1.	9-а класс	18	14	
2.	9- б класс	19	15	
3.	10- а класс	17	15	

2. Доход фирмы от продаж за 1-квартал текущего года (в тыс. сум.).

П/н	Содержание операции	Месяцы			Всего
		Январь	Февраль	Март	
1.	Продажа	205,7	304,8	215,6	
2.	Покупка	218,7	189,6	151,9	
3.	Транспортные расходы	40,5	21,4	24,5	
4.	Доход				



13 УРОК. МАТЕМАТИЧЕСКИЕ ФУНКЦИИ

Изучим математические функции в электронных таблицах MS Excel 2010, широко использующиеся в решении экономических и инженерных задач.

Упражнение 1. С использованием математических функций MS Excel 2010 вычислите аргументы функций $y=\arccos x$, $y=\cos x$, $y=e^x$, $y=\cos e^x$ с различными значениями.

Выполнение:

- В ячейки **A1:C1** введём значения;

	A	B	C
1	-0,9	0,2	-0,9

▪ Обратившись к функциям **acos**, **cos**, **exp**, в разделе **Формулы** получаем выражения, соответствующие ячейкам **A2:A5**.

	A	B	C
1	-0,9	0,2	-0,9
2	2,690566	1,369438	2,690566
3	0,62161	0,980067	0,62161
4	0,40657	1,221403	0,40657
5	0,918483	0,342328	0,918483

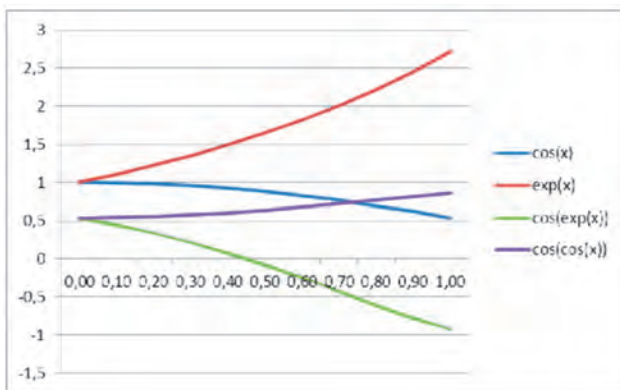
К примеру: поскольку **cos^{e^x}** является сложной функцией, в **A4** вводится **=EXP(A1)**, а в **A5** вводится **=EXP(A4)**.

▪ Используя метод **Относительной ссылки**, продолжим действия в остальных ячейках:

Таким образом, имеем возможность вычисления функции любой сложности.

Упражнение 2.

Получение графиков функций **y=cosx**, **y=e^x**, **y=cos^{e^x}**, **y=cos(cosx)** в значениях 0, 0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, 0,9, 1 **x**.



Выполнение:

▪ В ячейки **F3:P3** вводятся значения **x**;

▪ В ячейку **F4**

берётся функция **cos** из пункта **Математические** раздела **Формулы** и для его аргумента обращается в ячейку **F3** ;

▪ Вычисляемое в ячейке **F4** выражение осваивается в блоке **G4:P4** с помощью **относительной ссылки**;

▪ В ячейку **F5** вводится выражение **exp(F3)**, в ячейку **F6** - **cos(exp(F3))**, в ячейку **F7** - **cos(cos(F3))**, и соответственно, в блоках **F5:P5**, **F6:P6**, **F7:P7** осваивается с помощью **относительной ссылки**;

▪ Сделав ссылку в соответствующий пункт части **График** в разделе **Вставка** создаются графики рассматриваемых функций:

Упражнение 3. Получение рисунка, соответствующего функции $z = \cos x \cos y$.

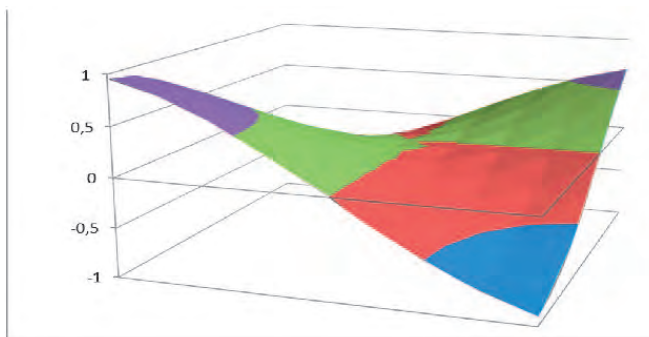
Выполнение:

- В ячейки **A1:J1** вводятся значения;
- В ячейки **A2:A10** вводятся соответствующие значения;
- В ячейку **B2** пишется выражение **=COS(\$A\$1)*COS(B1)**;
- В ячейках от **B2** до **B10** формула образуется с помощью смешанной ссылки;
- В ячейках от **B10** до **J10** выражение образуется с помощью смешанной ссылки. В результате получается следующее окно:

J10		=COS(A9)*COS(\$J\$1)									
	A	B	C	D	E	F	G	H	I	J	
1	0,00	0,30	0,60	0,90	1,20	1,50	1,80	2,10	2,40	2,70	
2	0,3	0,955336	0,825336	0,62161	0,362358	0,070737	-0,2272	-0,50485	-0,73739	-0,90407	
3	0,6	0,912668	0,788473	0,593847	0,346174	0,067578	-0,21705	-0,4823	-0,70446	-0,86369	
4	0,9	0,788473	0,681179	0,513037	0,299067	0,058382	-0,18752	-0,41667	-0,6086	-0,74616	
5	1,2	0,593847	0,513037	0,386399	0,225245	0,043971	-0,14123	-0,31382	-0,45837	-0,56198	
6	1,5	0,346174	0,299067	0,225245	0,131303	0,025632	-0,08233	-0,18293	-0,2672	-0,3276	
7	1,8	0,067578	0,058382	0,043971	0,025632	0,005004	-0,01607	-0,03571	-0,05216	-0,06395	
8	2,1	-0,21705	-0,18752	-0,14123	-0,08233	-0,01607	0,051621	0,114702	0,167537	0,205407	
9	2,4	-0,4823	-0,41667	-0,31382	-0,18293	-0,03571	0,114702	0,25487	0,37227	0,456417	
10	2,7	-0,70446	-0,6086	-0,45837	-0,2672	-0,05216	0,167537	0,37227	0,543749	0,666657	

- Выделяется блок **B2:J10**;
- Выбирается последовательность **Вставка – Другие – Поверхность**, и создаётся график рассматриваемых функций;
- Из полученного графика убираются надписи, не имеющие важности;

В результате получаем следующий рисунок:





ЗАПОМНИТЕ !

Для получения графиков функций в ячейки MS Excel 2010 вводятся значения функций и выбирается часть **График** раздела **Вставка**.



ВОПРОСЫ И ЗАДАНИЯ

1. Объясните последовательность создания графика функций.
2. Создайте график функции $y = \sin x$.
3. Создайте график функции $y = \cos^2 x$.



ДОМАШНЕЕ ЗАДАНИЕ

1. Создайте график функции $y = \sin^2 x$.
2. Создайте в одном окне графики функций $y = x^2$ и $y = x^3$.



14 УРОК.

ФУНКЦИИ ДЛЯ ВЫЧИСЛЕНИЯ ПРОИЗВЕДЕНИЙ

Произведение является распространённым действием в математических выражениях и на данном занятии мы познакомимся с методами вычислений, где участвует произведение:

1-метод. В ячейку вводится знак = и умножаются соответствующие выражения;

2-метод. Используется функция вычисления произведения.

Упражнение 1. В MS Excel 2010 имеется следующая таблица:

	A	B	C	D	E	F
1	0	10				
2	0,3	11				
3	0,6	12				
4	0,9	13				
5	1,2	14				
6	1,5	15				
7	1,8	16				
8	2,1	17				
9	2,4	18				
10	2,7	19				

- В столбце **C** вычислите произведение чисел столбцов **A** и **B** ;

▪ В столбце **D** – сумму чисел столбцов **A** и **B**;

▪ В столбце **E** – произведение чисел столбцов **A, B, C**;

▪ В столбце **F** – сумму чисел столбцов **A, B, C, D, E**;

Выполнение:

▪ В ячейке **C1** получается произведение **ПРОИЗВЕД(A1:B1)** блока **A1:B1**;

▪ В ячейке **D1** получается сумма **=A1+B1** блока **A1:B1**;

▪ В ячейке **E1** получается произведение **ПРОИЗВЕД(A1:D1)** блока **A1:D1**;

▪ В ячейке **F1** получается сумма **=СУММ(A1:E1)** блока **A1:E1**;

▪ Вышеприведённые формулы копируются соответственно в ячейки **C2:F10**. В результате получается следующее окно:

	A	B	C	D	E	F
1	0	10	0	10	0	20
2	0,3	11	3,3	11,3	123,1	149
3	0,6	12	7,2	12,6	653,2	685,6
4	0,9	13	11,7	13,9	1903	1942
5	1,2	14	16,8	15,2	4290	4337
6	1,5	15	22,5	16,5	8353	8409
7	1,8	16	28,8	17,8	14764	14828
8	2,1	17	35,7	19,1	24343	24417
9	2,4	18	43,2	20,4	38071	38155
10	2,7	19	51,3	21,7	57108	57202



ЗАПОМНИТЕ !

Для вычисления произведения данных чисел используется функция **ПРОИЗВЕД**.



ВОПРОСЫ И ЗАДАНИЯ

1. Какие методы вычисления произведения данных чисел вы знаете?
2. Вычислите произведение данных 20 чисел без помощи функции **ПРОИЗВЕД**.
3. Поделите сумму данных чисел 10, 20, 40, 80, 160 на число 40.



ДОМАШНЕЕ ЗАДАНИЕ

1. Сделайте вычитание числа 56 из произведения чисел 10, 20, 40, 80, 160.
2. В столбцы **A, B, C** введите любые 20 чисел. Из произведения чисел столбцов **A** и **C** в столбце **D** сделайте вычитание чисел столбцов **B, C**.




15 УРОК.

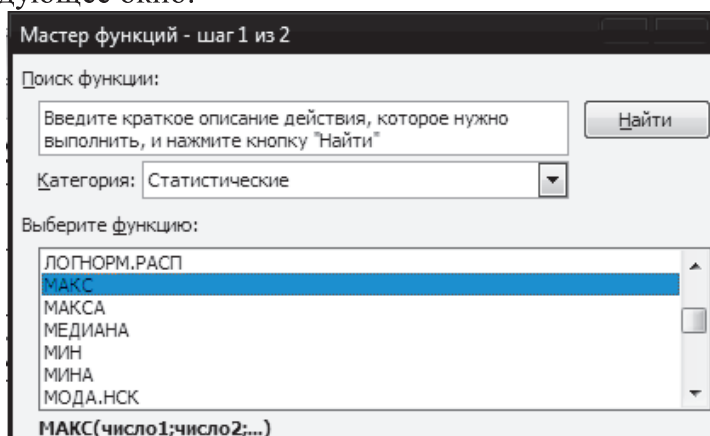
СТАТИСТИЧЕСКИЕ ФУНКЦИИ

Ещё одной из возможностей MS Excel 2010 является решение соответствующих задач при помощи данных статистических функций.

Использование статистических функций осуществляется двумя методами:

1-метод. Посредством прямого ввода функций в ячейки;

2-метод. Посредством выбора пункта  строки формул, можно открыть следующее окно:



В MS Excel 2010 статистических функций много и ниже мы рассмотрим лишь некоторые из них и их предназначение:

	Функция	Предназначение
1.	МАКС	Вычисляет самое большое число из данных.
2.	МИН	Вычисляет самое малое число из данных.
3.	СРЗНАЧ	Вычисляет среднеарифметическое значение чисел из данных.
4.	СЧЁТЕСЛИ	Считает количество заполненных ячеек условно удовлетворяющего блока.

Упражнение 1. В ячейке **D8** получите среднеарифметическое значение самого большого и самого малого числа блока **A1:C5**.

Выполнение:

- В блоки **A1:C5** вводим произвольные числа;
- В ячейку **D6** вводим следующую формулу (формула нахождения самого большого числа из данных): **=МАКС(A1:C5)**;
- В ячейку **D7** вводим следующую формулу (формула нахождения самого малого числа из данных): **=МИН(A1:C5)**;

- В ячейку **D8** вводим следующую формулу (формула нахождения среднего арифметического значения чисел из данных): **=СРЗНАЧ(D6:D7)**;


В результате получается следующее окно:

	A	B	C	D
1	4	5	44	
2	33	-8	3	
3	100	1	7	
4	11	12	-22	
5	10	8	9	
6				100
7				-22
8				39



ЗАПОМНИТЕ !

Использование статистических функций осуществляется двумя способами:

- 1-способ. Посредством прямого ввода функций в ячейки;
- 2-способ. Посредством выбора пункта  строки формул.



ВОПРОСЫ И ЗАДАНИЯ

1. Объясните выбор последовательности функции **МАКС** из списка статистических функций.
2. Найдите самое большое число из данных 10 чисел. Выполните упражнение двумя способами.



ДОМАШНЕЕ ЗАДАНИЕ

1. Вычислите среднегеометрическое значение чисел 4, 6, 8, 9, 10.
2. Самостоятельно составьте задачу с помощью функции **СЧЁТЕСЛИ** и решите её.



16 УРОК. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

В MS Excel 2010:

1. Постройте график функции $y = \ln 2x + \sin x$.

2. Из произведения чисел 1, 3, 5, 7, 9 сделайте вычитание произведения чисел 2, 4, 6, 8, 10.
3. Найдите пропорцию суммы данных 20 чисел к произведению произвольных 10 чисел.
4. Вычислите кубический корень чисел 10, 20, 30, 40.
5. Вычислите 34 процента числа 900.
6. Вычислите среднеарифметическое значение квадрата чисел 1, 2, 3, 4, 5, 6, 7, 8, 9.
7. Вычислите данное выражение: $345-45*5678+12:456-89+(34-8*9)$.
8. Вычислите $\sqrt[4]{625}-\sqrt[3]{\frac{8}{3}-2*6}$. 9. Вычислите $\frac{2}{3}+\frac{4}{5}+\frac{6}{7}+\frac{8}{9}$.
10. Вычислите $\frac{2}{3}+\frac{4}{5}+\frac{6}{7}+\frac{8}{9}$. 11. Вычислите $\frac{1}{2}-\frac{1}{3\cdot 4}+\frac{1}{5\cdot 6}-\frac{1}{7\cdot 8}+\frac{1}{9\cdot 10}$.
12. Вычислите $\frac{\log_3 9}{2}-\frac{\sqrt[3]{67}}{9}+3\cdot\frac{5}{\sqrt{36}}$.
13. Вычислите $\sqrt{2+\sqrt{2+\sqrt{2+\sqrt{2+\sqrt{2}}}}}$.



17 УРОК. РЕШЕНИЕ НЕКОТОРЫХ ЗАДАЧ В MS EXCEL

В MS Excel 2010 имеется возможность решать уравнения и систему уравнений с помощью графического метода. Для этого уравнения или система уравнений приводятся в виде функций. Посредством анализа графика определяются примерные решения уравнений.

Упражнение 1. Найдите промежуточное решение уравнения $x^4 - 2x^3 - 3x^2 + 1 = 0$.

Выполнение:

- В ячейку **A1** вводится x , в ячейку **A2** вводится $f(x)$;
- В ячейку **V1:F1** вводятся значения x ;
- В ячейку вводим данную формулу **B2:** $=B1^4-2*B1^3-3*B1^2+1$;
- В ячейки блока **C2:F2** копируется приведённая формула. В результате получается следующая таблица:

Из таблицы видно, что промежуточное решение рассматриваемого уравнения лежит в $(0;1)$ $(2;3)$. Для ещё большего уменьшения

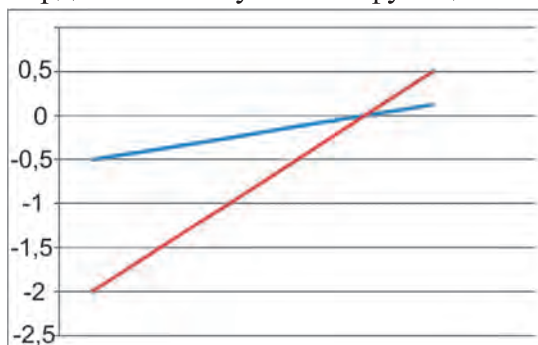
F2		fx = F1^4-2*F1^3-3*F1^2+1				
	A	B	C	D	E	F
1	x	0	1	2	3	4
2	f(x)	1	-3	-11	1	81

рассматриваемого промежутка можем достаточно приблизиться к поиску решения.

Упражнение 2. Решите систему уравнений $\begin{cases} x - 2y - 1 = 0 \\ 2x - y + 1 = 0 \end{cases}$.

Выполнение. Для решения данной системы уравнений рассмотрим два графика функций: $\begin{cases} y = (x - 1) / 2 \\ y = 2x + 1 \end{cases}$

Для решения рассматриваемой системы уравнений находятся значения x , равные ординатам полученных функций:



Из графика видно, что при $x=1, y=0$.



ЗАПОМНИТЕ !

С помощью MS Excel 2010 есть возможность примерного решения уравнений и системы уравнений посредством анализа графика.



ВОПРОСЫ И ЗАДАНИЯ

1. Объясните последовательность решения уравнений графическим методом.
2. Создайте таблицу распорядка дня жизни человека.

3. Решите квадратное уравнение.



ДОМАШНЕЕ ЗАДАНИЕ

1. Составьте диаграмму показателей успеваемости учащихся класса по предмету «Информатика и информационные технологии».
2. Составьте таблицу ваших годовых расходов.



18 УРОК. ПРАКТИЧЕСКИЕ ЗАНЯТИЯ

1. Вычислите площадь круга в MS Excel 2010.
2. Вычислите $\frac{2 - |0.2 + 3|}{1 + 2^3}$.
3. Создайте график функций $y = \sin 2x$.
4. Вычислите 15% чисел в данной таблице:

П/н	Кварталы	Оборотные средства	15%ое отчисление в местный фонд
1.	I	1175000	
2.	II	211800	
3.	III	3151000	
4.	IV	4151000	
	Всего:		

5. Вычислите отношение скорости к расстоянию и времени и создайте его график.

S	v	t
12	5	2,4
10	3	3,333333
5	4	1,25
2,5	5	0,5



19 УРОК. КОНТРОЛЬНАЯ РАБОТА

1 вариант

1. В ячейке **D9** второго листа (Лист2) вычислите частное чисел ячейки **B3** первого листа (Лист1) и ячейки **C8**.
2. Вычислите данные действия в MS Excel: $2*(6-5)/5$.
3. Вычислите длины кругов с разными радиусами в MS Excel.
4. Создайте график функции $y=\sin x$.

2 вариант

1. Вычислите в ячейке **D3** второго листа (Лист2) частное чисел ячейки **B1** первого листа (Лист1) и ячейки **B3**.
2. Вычислите объём куба в MS Excel.
3. Вычислите данное выражение: $(6:8)-9:81$.
4. Создайте график функции $y=\ln x$.

3 вариант

1. В ячейку **B1** введите фамилию ученика, в ячейку **C1** его имя, в ячейку **D1** отчество. В ячейке **E1** объедините все данные.
2. Найдите самое малое число в диапазоне **A1:D9**.
3. Вычислите данное выражение: $(6-8)*6+1$.
4. Создайте график функции $y=x^3$

4 вариант

1. Создайте график функции квадрата.
2. Вычислите произведение чисел в диапазоне **A2:E10**.
3. Вычислите данное выражение: $\sqrt{9} + \sqrt{81} - 4$.
4. Создайте график функции $y=x^3 - 1$

ГЛАВА II. БАЗА ДАННЫХ



ВНИМАТЕЛЬНО ПРОЧИТАЙТЕ ЭТУ ГЛАВУ, И ВЫ СМОЖЕТЕ ОВЛАДЕТЬ СЛЕДУЮЩИМИ ЗНАНИЯМИ, УМЕНИЯМИ И КОМПЕТЕНЦИЯМИ:

- Понятием о базе данных и системе их управления;
- Знаниями об основных элементах MS Access 2010 и особенностях полей;
- Умениями создавать базы данных в MS Access 2010;
- Навыками взаимного связывания таблиц в MS Access 2010;
- Навыками поиска и переработки информации по заданному шаблону в MS Access 2010;
 - Умениями создавать базы данных в MS Access 2010 и использовать формы в редактировании;
 - Навыками выполнения математических действий в MS Access 2010.





20 УРОК. ПОНЯТИЕ О БАЗЕ ДАННЫХ

Использование информации, то есть сбор информации по определенной технологии, хранение, переработка и передача являются важными вопросами. Информационные ресурсы нужно организовать таким образом, чтобы их использование стало максимально удобным и полезным. Новый подход в решении задач стал причиной появления новой технологии. Основу данной технологии составляют база и банки данных.

База данных (БД) – набор введённой в память компьютера взаимосвязанной и упорядоченной информации, имеющей единую структуру.

Информационная модель – набор структуры взаимосвязанной информации и проводимых над ней операций.

Известно, что для размещения во внутреннюю среду одной и той же информации можно использовать различные структуры и модели информации. Их выбор возлагается на создающего БД пользователя и зависит от многих факторов.

Основными методами создания БД являются:

- Иерархическая модель– в данной модели данные хранятся в древовидном виде. Единицей работы является надпись;
- Сетевая модель – хотя в данной модели данные хранятся и в древовидном виде, разветвление узлов не ограничивается;
- Реляционная (взаимосвязанная) модель – в данной модели данные хранятся в виде таблиц.

Данные можно получить двумя способами: неструктурированным и структурированным.

Неструктурированный – это способ получения данных, при котором отсутствует соглашение о способе изображения данных.

Структурированный – это способ получения данных, при котором имеется соглашение о способе их изображения. Примером неструктурированных данных может служить следующее: к сведениям, включающим данные об учащемся (фамилия, имя, отчество, место проживания, телефон) можно указать следующее:

Фамилия – *Назаров*, Имя – *Нодир*, Отчество – *Ахмедович*, Место проживания – *город Навои*, Номер телефона+998947777777.

Оформив эти сведения в виде таблицы, можно получить структурированные данные:

П/п	Фамилия	Имя	Отчество	Место проживания	Номер телефона
1.	<i>Назаров</i>	<i>Нодир</i>	<i>Ахмедович</i>	<i>город Навои</i>	+998947777777



ЗАПОМНИТЕ!

База данных (БД) – набор введенной в память компьютера взаимосвязанной и упорядоченной информации, имеющей единую структуру.



ВОПРОСЫ И ЗАДАНИЯ

1. Что вы понимаете под базой данных?
2. Дайте сведения о модели БД.
3. Что вы понимаете под структурированными данными?
4. Сформируйте неструктурированные данные об учащихся вашего класса с указанием фамилии, имени, отчества, даты рождения, номера телефона, оценок по предметам.



ДОМАШНЕЕ ЗАДАНИЕ

1. Сформируйте неструктурированные данные об авторах учебников для 10-го класса и дате их издания.
2. Создайте структурированные данные об учащихся вашего класса с указанием фамилии, имени, отчества, даты рождения.



21 УРОК.

СИСТЕМЫ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ

Для составления БД, введения дополнительных данных и пользования ими необходимо создать специальное программное обеспечение.

Такой программный комплекс именуется системой управления базой данных.

Система управления базой данных (СУБД) – система программных средств, предназначенная для создания, пополнения и поддержания(применения) БД пользователями.

Примерами СУБД могут быть программные средства MS Access, OpenOffice.org Base, Cache, IMS, Firebird, MySQL.

Одним из основных свойств СУБД является не только перечень процедур, используемых при введении и сохранении данных, но и описание их структур.

Файл, управляемый СУБД и сохранивший данные, сначала именуется как банк данных, затем как “База данных”.

При создании БД необходимо иметь в виду следующее:

- Типы и виды данных не должны быть связаны с поддерживающими их (применяемыми) программами;
- Для поиска нужных данных, содержащихся в базе данных, или ознакомления с ними, не должно быть необходимости создавать какую-либо программу.

Следует отметить, что почти все нынешние СУБД создаются на основе реляционных моделей. По этой причине и корпорация Microsoft Office создаёт самые массовые программные средства.

Эти программные средства позволяют высокопрофессионально готовить документы любого профиля.

Одно из них – это программа Microsoft Access, предназначенная для работы с БД.



ЗАПОМНИТЕ!

Система управления базой данных (СУБД) – система программных средств, предназначенная для создания, пополнения и поддержания(применения) БД пользователями.



ВОПРОСЫ И ЗАДАНИЯ

1. Дайте определение **СУБД**.
2. Какие программные средства создания БД вы знаете?

3. Дайте сведения о моделях БД.
4. Оформите сведения об одноклассниках - фамилия, имя, домашний адрес.



ДОМАШНЕЕ ЗАДАНИЕ

1. Оформите сведения об авторах книг по предметам, о названиях книг, о годе издания.
2. Оформите сведения об областях нашей республики, их площади, численности населения.

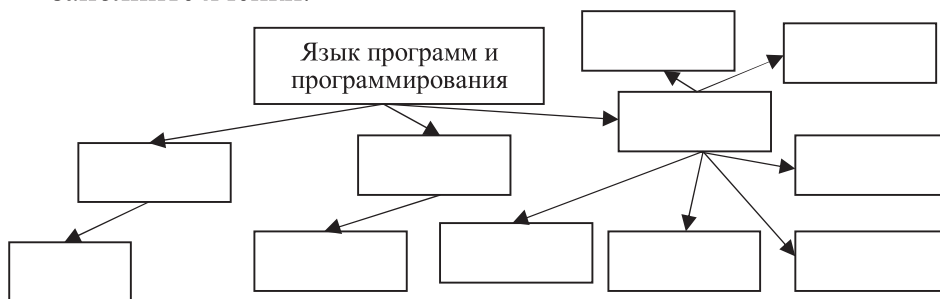


22 УРОК. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

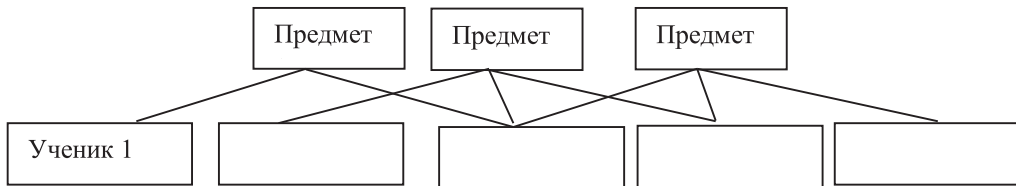
1. Сделайте модели для создания базы данных одноклассников.
2. Создайте указанную таблицу в MS Excel 2010 и заполните её сведениями об одноклассниках:

п /н	Фамилия	Имя	Оценки по предметам				
			матема- тика	физика	химия	биология	исто- рия
1.							
2.							
3.							

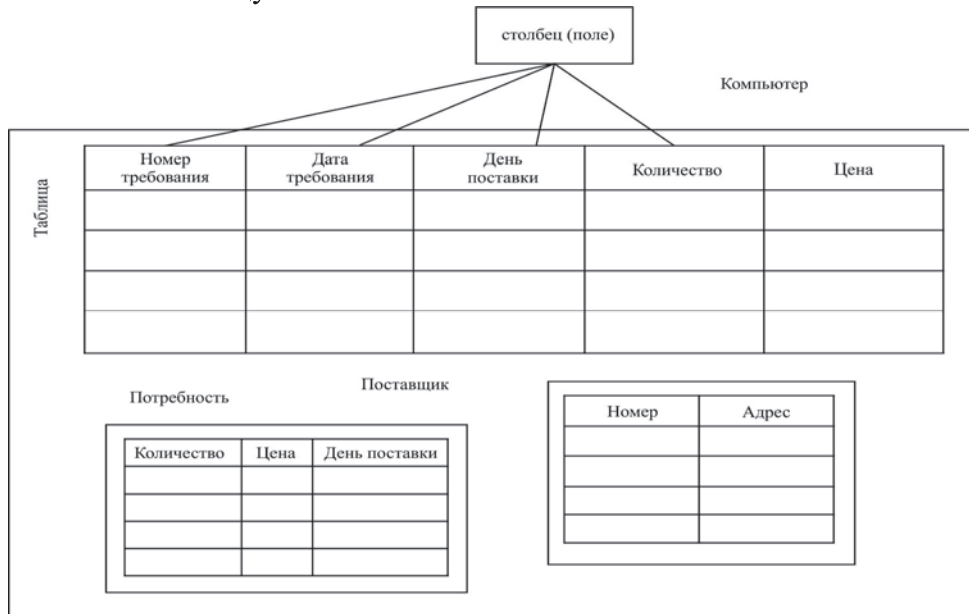
3. Распечатайте указанный рисунок иерархической модели и заполните ячейки.



4. Распечатайте указанный рисунок отраслевой модели и заполните ячейки:



5. Распечатайте указанный рисунок реляционной модели и заполните таблицу:



6. Приведите пример иерархических моделей и сделайте рисунок.
7. Приведите пример отраслевых моделей и сделайте рисунок.
8. Приведите пример реляционных моделей и сделайте рисунок.



23 УРОК. ОСНОВНЫЕ ЭЛЕМЕНТЫ И СВОЙСТВА ПОЛЕЙ В MS ACCESS 2010

Microsoft Access 2010 относится к реляционному типу СУБД, в настоящее время входит в число самых массовых систем управления базой данных и предназначен для создания БД, сохранения данных в БД, поиска и автоматической обработки.

Дополнительным преимуществом MS Access 2010 является то, что он находится в одной среде с такими программами, как MS Word, MS Excel, входящими в состав Microsoft Office. Созданные в приложениях этих программ данные можно легко импортировать или экспортировать из одного приложения в другое. С помощью MS Access 2010 обычный пользователь без громоздкого и сложного программного обеспечения может создавать и применять БД.

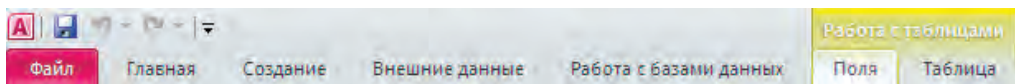
Можно рассмотреть три способа загрузки MS Access 2010:

- При помощи ярлыка на рабочем столе пользователя;
- При помощи ярлыка программы;
- При помощи главного меню.

Работа с MS Access 2010

В верхнем ряду экрана MS Access 2010 сверху данных расположено меню для выполнения различных действий.

Разделы - **Файл**, **Главная**, **Создание**, **Внешние данные**, **Работа с базами данных**, **Поля**, **Таблица**.



Файл – возможности сохранения данных в памяти, обращения к файлам, распечатки данных, настройки параметров программы.

Главная – возможности фильтрации данных таблиц, а также целый ряд действий по отбору и обработке данных.

Создание – можно производить целый ряд действий по созданию таблиц, опросов, форм отчётов, модулей, макросов.

Внешние – можно производить целый ряд действий по импорту, экспорту созданных данных, а также отправлять на электронные адреса.

Работа с базами данных – можно производить целый ряд действий по созданию макросов, соединять полученные таблицы, связывать с серверной базой SQL.

Поля – возможности пользования типами полей, изменения их свойств, редактирования форматов таблиц, математическими функциями.

Таблица – можно создавать макросы, устанавливая соединяющие свойства таблиц.

Основные элементы MS Access 2010:

- **Таблицы** – служат для хранения данных;
- **Запросы** – служат для предоставления требований отбора данных и внесения в них изменений ;
- **Формы** – служат для просмотра и редактирования данных;
- **Страницы** – с помощью браузера Internet Explorer служат для просмотра данных MS Access 2010 файлов в формате HTML;
- **Отчеты** – дают возможность обобщения и печатания данных;
- **Макрос** – обеспечивает автоматическое выполнение одного или нескольких действий.

Поля MS Access 2010 считаются элементарной единицей логического упорядочивания данных.

Типы полей, описанных в MS Access 2010

Тип	Определение
Текстовый (Текстовый)	Тип поля, образованный из текстов, длина (объём) которого не должен превышать отметки 255
Поле МЕМО (поле МЕМО)	В поле МЕМО хранятся текстовые и цифровые данные, объём которых не превышает отметку 65535.
Числовой (Числовой)	Тип поля, хранящий цифровые значения, диапазон которых определяется в параметре размера поля.
Дата/время (Дата/время)	Значение поля, состоящее из даты и времени(8 байт), в которое можно ввести годовой диапазон от 100 до 9999.
Денежный (Денежный)	Поле из цифровых значений, в котором могут храниться цифры влево от запятой – 15 разрядов и вправо от запятой – 4 разряда.
Счетчик (Счетчик)	При пополнении таблицы новыми данными автоматически происходит увеличение на одно значение.
Логический	Поле хранит значения “Да” или “Нет”. В

(Логический)	программе MS Access “1” – используется как “Да” и “0” – используется как “Нет”.
Поле объекта OLE (Поле объекта OLE)	На поле размещаются объекты, переработанные OLE - сервером.
Гиперссылка (Гиперссылка)	Это поле осуществляет гиперсвязь жёсткого диска с другим файлом.
Вложение (Вложение)	Создаётся возможность работать с математическими функциями.



ЗАПОМНИТЕ!

Файлы MS Access 2010 “.accdb” обладают свойством расширения.



ВОПРОСЫ И ЗАДАНИЯ

1. К какому типу СУБД относится MS Access 2010?
2. Объясните основные элементы объектов MS Access 2010.
3. Сообщите об основных типах полей MS Access 2010?
4. Начните работать в MS Access 2010 и в его главном окне создайте новую базу данных.



ДОМАШНЕЕ ЗАДАНИЕ

1. Создайте новую базу данных в MS Access 2010. Назовите Базу “ученики” и сохраните на диске **D**.
2. Проанализируйте типы полей, описанных в MS Access 2010.



24 УРОК.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

1. Начните работать в MS Access 2010, используя следующие приёмы:
 - С помощью ярлыка на рабочем столе пользователя;
 - С помощью значка ярлык программы MS Access 2010;
 - С помощью главного меню.
2. Объясните возможности меню программы MS Access 2010 **Файл, Главная, Создание, Внешние данные, Работа с базами данных, Поля, Таблица.**

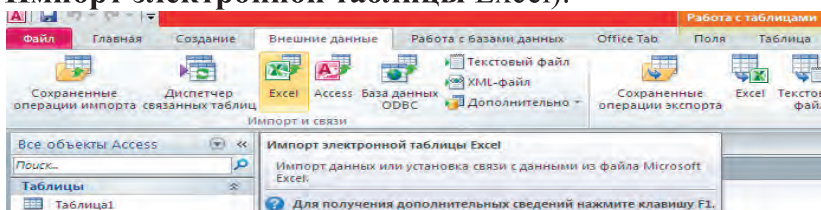
3. Объясните функции основных элементов программы MS Access 2010.

4. Объясните функции типов полей программы MS Access 2010.

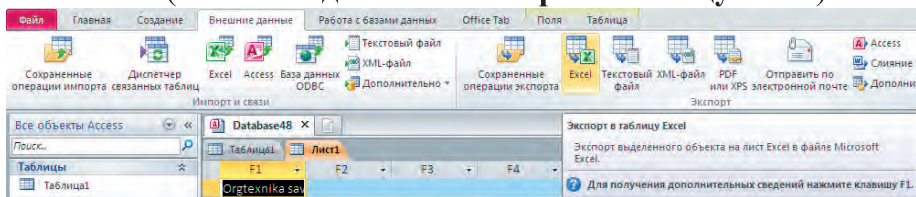
5. Нарисуйте следующую таблицу в программе MS Excel 2010:

П/н	Фамилия	Имя	Год рождения	Номер телефона
1.				
2.				
3.				
4.				

Полученную таблицу импортируйте в MS Access 2010 (**Внешние данные-Импорт электронной таблицы Excel**):



6. Загрузите созданный в MS Access 2010 файл и экспортируйте его в MS Excel 2010 (**Внешние данные-Экспорт в таблицу Excel**):



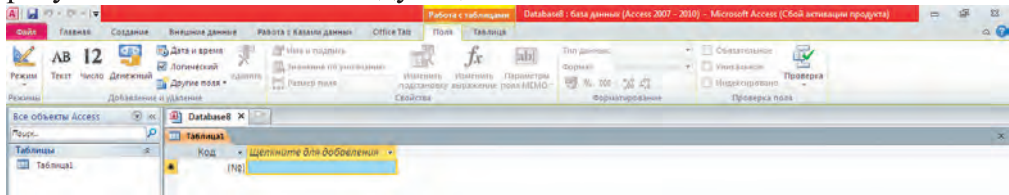
6. Загрузите созданный в MS Access 2010 какой-либо файл посредством следующей последовательности: **Файл-Открыть**.

25 УРОК. СОЗДАНИЕ БАЗЫ ДАННЫХ В MS ACCESS 2010

Для создания базы данных в MS Access 2010 из перечня программ Microsoft Office выбираем Microsoft Access 2010 и загружаем:



В образовавшемся окне выбираем раздел **Новая база данных**. В результате появляется следующее окно:



В появившуюся в MS Access 2010 таблицу можно ввести данные тремя способами:

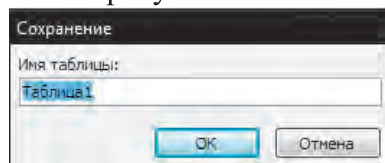
1. **Ввести данные непосредственно.**
2. **Режим-Конструктор.**
3. **Создание-Конструктор таблиц.**

Для получения одной таблицы в окне MS Access 2010 можно воспользоваться разделом **Режим-Конструктор**. Для того, чтобы создать в окне несколько таблиц и установить между ними взаимосвязь, следует выполнить эту операцию посредством раздела **Конструктор таблиц** в меню **Создание**.

Упражнение 1. Способ составления сведений об учащихя в MS Access 2010.

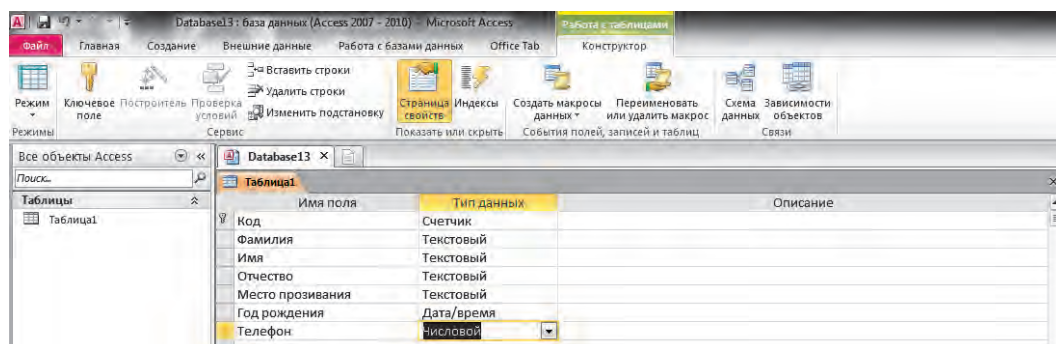
Выполнение:

Выполняется последовательность **Режим-Конструктор** программы MS Access 2010. В результате появляется следующее окно:

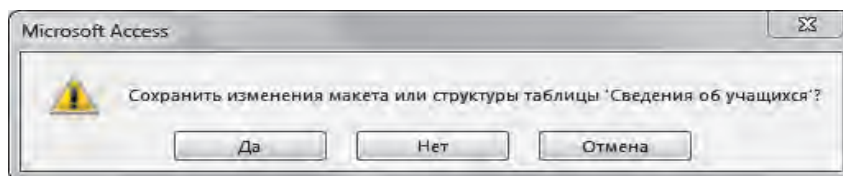


- В ряду **Имя таблицы** сохраняем под названием “Сведения об учащихся” и составляем таблицу в следующем порядке:

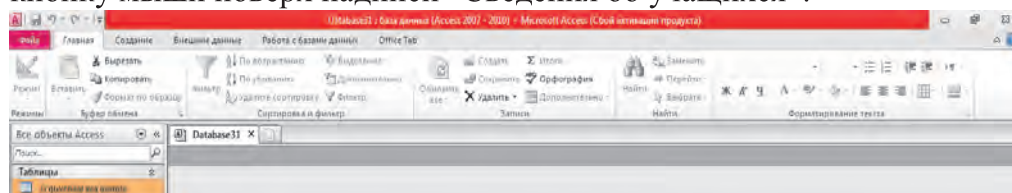
Идентификационный номер учащегося	Числовой (числовой)
Фамилия учащегося	Текстовой (текстовой)
Имя учащегося	Текстовой (текстовой)
Отчество учащегося	Текстовой (текстовой)
Адрес учащегося	Текстовой (текстовой)
Год рождения учащегося	Дата/время (Дата/время)
Телефон учащегося	Числовой (числовой)



- Выходим из этого окна (выбирается значок x в углу окна). В результате появляется следующее окно:



- В окне выбирается кнопка **Да** (получается таблица под названием “Сведения об учащихся”).
- Для создания базы данных об учащихся нужно дважды нажать кнопку мыши поверх надписи “Сведения об учащихся”:



- Полученная база данных заполняется сведениями об учащихся:

Код	Фамилия	Имя	Отчество	Место жительства	Дата рождения	Телефоны
1	Махмудов	Ахмад	Нажимович	г. Навои, ул.М.Тароби к. 117	23.09.2003	944826749
2	Махмудова	Лобар	Ахадовна	г. Навои, ул.Навои. 23.56	06.09.2004	932456789

- Созданную базу данных сохраняем в памяти компьютера (**Файл – Сохранить базу данных как**).

Есть возможность изменить надпись в столбце таблицы, созданной в MS Access 2010. Для этого выполняется действие **Главная – Режим – Конструктор**.



ЗАПОМНИТЕ!

В таблицы, созданные в MS Access 2010, можно вводить данные тремя способами: 1. Непосредственно вводить данные. 2. Режим-Конструктор. 3. Создание-Конструктор таблиц.



ВОПРОСЫ И ЗАДАНИЯ

1. Перечислите пункты меню MS Access 2010?
2. В каком меню MS Access 2010 расположен раздел **Конструктор таблиц**?
3. Выберите раздел **Режим–Конструктор** в MS Access 2010 и создайте БД учащихся, т.е. фамилия, имя, адрес.



ДОМАШНЕЕ ЗАДАНИЕ

1. Начните работу в MS Access 2010 тремя способами.
2. Выберите раздел **Создание–Конструктор таблиц** и создайте БД учащихся, т.е. фамилия, имя, адрес, номер телефона.



26 УРОК. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

1. Пользуясь разделом **Режим–Конструктор** MS Access 2010, создайте таблицу, дающую сведения о названии учебника, авторе, годе издания, объёме.

2. Выберите раздел **Создание–Конструктор таблиц**, составьте три таблицы - (в первой таблице - сведения об областях Республики Узбекистан, во второй – сведения о численности населения областей,

заводах и фабриках, в третьей таблице – сведения о площади города Ташкента, населении, заводах и фабриках).

3. Посредством раздела **Главная** пункта **Итоги** MS Access 2010 в столбцах вычислите наибольшее арифметическое, наименьшее арифметическое, среднее арифметическое значение и сумму числа 10.

4. В MS Access 2010 составьте следующую таблицу:

Код	Фамилия	Имя	Отчество	Адрес	Математика	Физика	Информати
1	Назаров	Нодир	Косимович	г. Ташкент	4	3	5
2	Маматов	Вали	Аллулович	Кибрайская обл.	4	4	4
3	Ашурсева	Васила	Ахматова	Навоийская обл.	3	3	5
4	Шодмонова	Шафога	Ташмурадовна	Самаркандская обл.	5	5	3
(№)							
Итого					4	3,75	4,25



27-УРОК. СОЕДИНЕНИЕ ТАБЛИЦ В MS ACCESS 2010

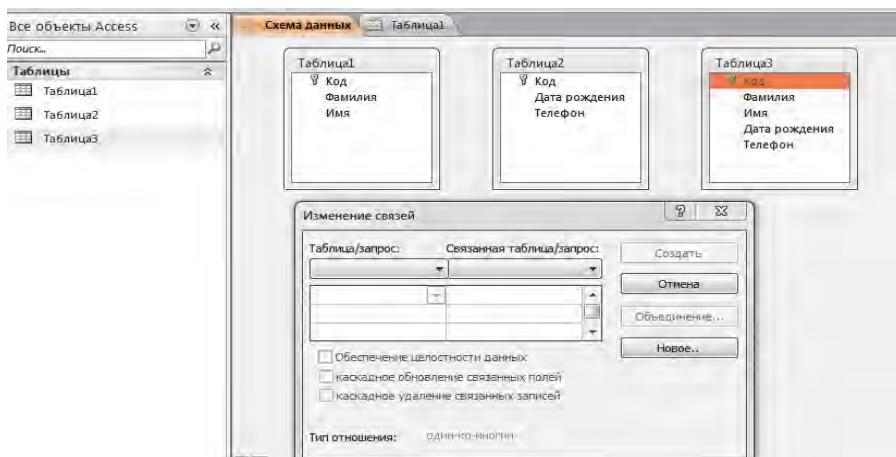
Соединение таблиц в MS Access 2010 имеет большое значение. Потому что в СУДБ используется составление нескольких таблиц и группировка нужных данных в одном месте. Соединение таблиц в MS Access 2010 осуществляется через раздел **Схема данных** в меню **Работа с базами данных**. Соединение таблиц и группировка их при помощи запроса в одну таблицу осуществляется через раздел **Конструктор запросов** в меню **Создание**.

Упражнение 1. Рассмотрим задачу соединения трёх таблиц с данными.

Выполнение:

- Составляем через раздел **Конструктор таблиц** в меню **Создание** три таблицы и сохраняем их под названием “таблица-1”, “таблица-2”, “таблица-3”. Затем выбираем раздел **Схема данных** в меню **Работа с базами данных**, в появившемся окне **Добавление таблицы** отмечаем каждую таблицу, выбираем кнопку **Добавить** ;

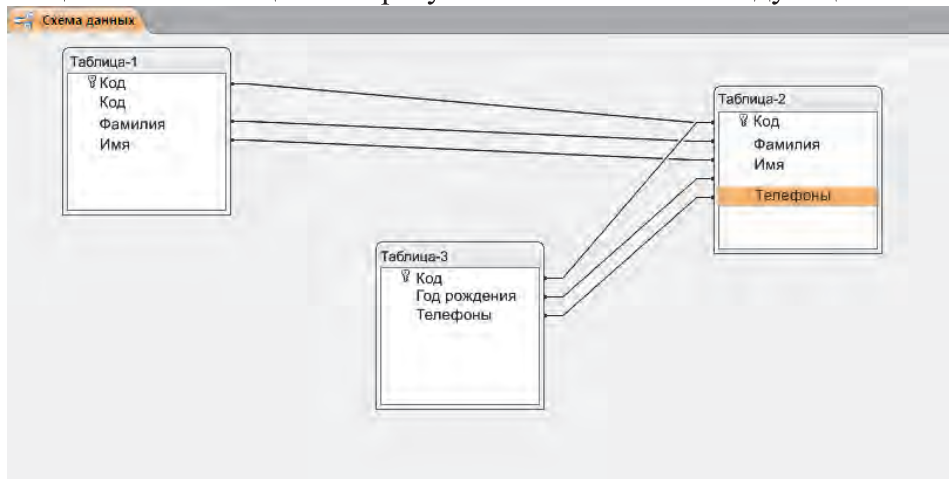
- Для соединения полученных таблиц выбирается в меню раздел **Изменить связи**. В результате появляется следующее окно:



- Выбрав из окна кнопку **Новое**, появляется новое окно под названием **Создание**. В окне из ряда **Левая таблица** выбираем “Таблицу1”, **Правая таблица** выбираем “Таблицу3”. В ряды **Левый столбец** и **Правый столбец** выбирается “Код”, затем – кнопка **ОК**;

- Из окна **Изменение связей** в первом ряду “Таблицы 1” выбирается “Код”, во втором ряду – “Фамилия”, в третьем ряду выбирается “Имя”, затем – кнопка **Создать**;

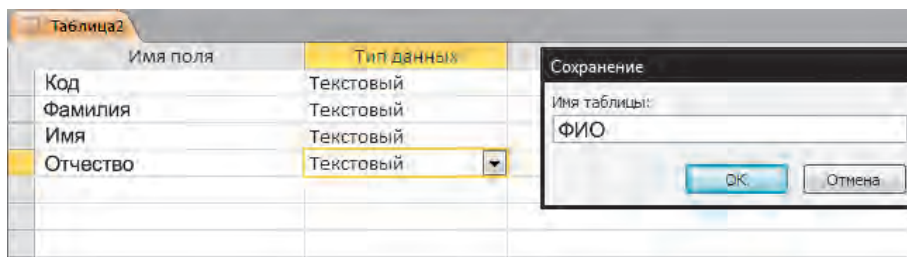
Эта последовательность действий применяется также для “Таблица-1” и “Таблица-3”. В результате появляется следующее окно:



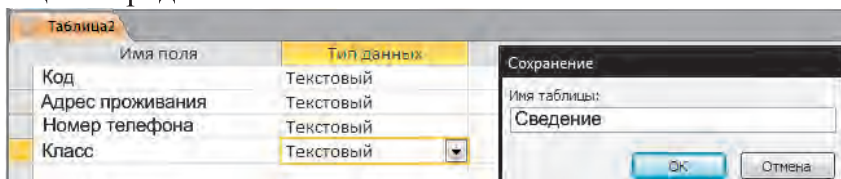
Упражнение 2. Поместить сведения об учащихся в 3 таблицы, и при помощи запросов объединить их в одну.

Выполнение:

- В первую таблицу вводятся данные об учащихся: фамилия, имя, отчество;
- Во вторую таблицу – адрес, номера телефонов учащихся, сведения о классе;
- В третью таблицу – данные о полученных оценках по информатике, математике и физике;
- Для получения первой таблицы начинаем работать в MS Access 2010 и в меню **Создание** выбираем раздел **Конструктор таблиц**. Полученную таблицу заполняем в следующем порядке:

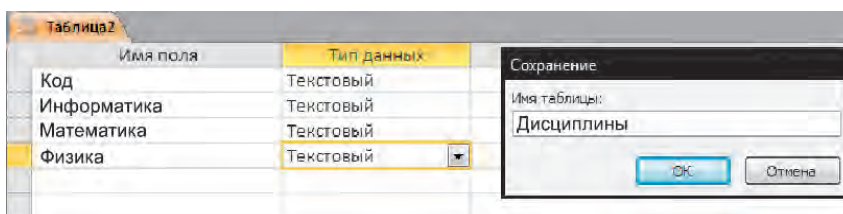


- Созданную таблицу сохраняем под названием “ФИО”;
- Для составления второй таблицы в меню **Создание** выбираем раздел **Конструктор таблиц**. Полученную таблицу заполняем в следующем порядке:



Таблицу сохраняем под названием “Сведение”;

- Порядок составления третьей таблицы осуществляется в указанной выше последовательности и полученную таблицу заполняем в следующем порядке:



■ Созданную таблицу сохраняем под названием “Дисциплины” и таблицы заполняем данными в следующем порядке:

а)

Код	Фамилия	Имя	Отчество	Щелкните для добавления
1	Назаров	Исмоил	Махмудович	
2	Хошимов	Вали	Алимович	
*	(№)			

б)

Код	Адрес проживания	Номер телефона	Класс	Щелкните для добавления
1	Навои	944826749	10 А	
2	Самарканд	4387778	10 В	
*	(№)			

с)

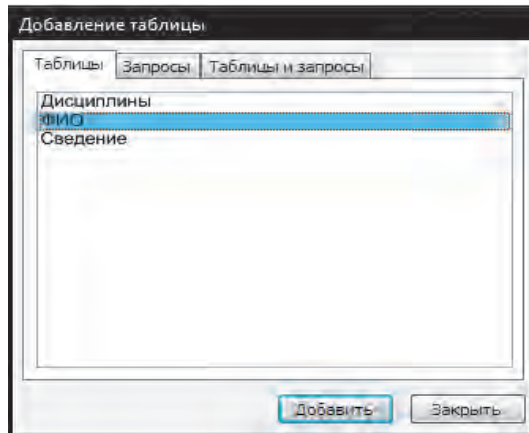
Код	Информатика	Математика	Физика	Щелкните для добавления
1	5	4	3	
2	3	4	4	
*	(№)			

■ После того, как таблицы данными заполнились, в меню **Создание** выбирается раздел **Конструктор запросов** и выполняется следующая последовательность:

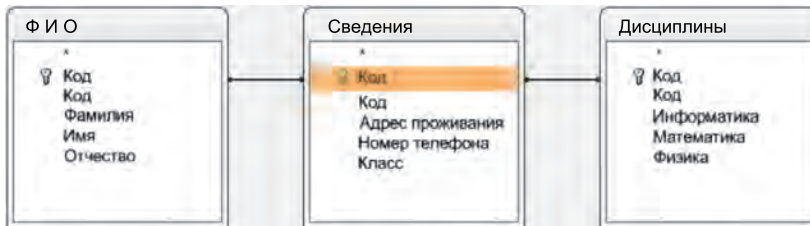
а) Выбирается раздел “ФИО”, затем выбирается кнопка **Добавить**.

б) Выбирается раздел “Сведение”, а затем – кнопка **Добавить**.

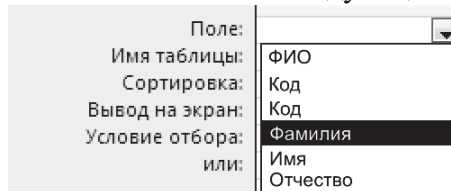
с) Выбирается раздел “Дисциплины”, далее – кнопка **Добавить** и посредством кнопки **Заккрыть** осуществляется выход.



Полученные три формы соединяем в следующем порядке (действия по передвижению осуществляются нажатием кнопки мыши):



В трёх таблицах выбираем соответствующие поля и функции. Для этого курсор мыши наводится на первый столбец раздела **Поле**, и после нажатия левой кнопки появляется следующий вид:



В полученную таблицу размещаются данные в следующем порядке:

Поле:	Фамилия	Имя	Отчество	Адрес проживания	Номер телефона	Класс	Информатика	Математика	Физика
Имя таблицы:	ФИО	ФИО	ФИО	Сведения	Сведения	Сведения	Дисциплины	Дисциплины	Дисциплины
Сортировка:									
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:								Дисциплины	
или:									

В меню выбирается команда **Выполнить**.

- В результате появляется следующее окно:

Фамилия ▾	Имя ▾	Отчество ▾	Адрес проживания ▾	Номер телеф ▾	Класс ▾	Информати ▾	Математи ▾	Физика ▾
Назаров	Исмоил	Махмудович	Навои	944826749	10 А	5	4	3
Хошимов	Вали	Алимович	Самарканд	4387778	10 В	3	4	4

Если обратить внимание на таблицу, то видно, что данные трёх таблиц посредством запросов объединились в одну.



ЗАПОМНИТЕ!

Для соединения таблиц в программе MS Access 2010 используется раздел **Схема данных** в меню **Работа с базами данных**.



ВОПРОСЫ И ЗАДАНИЯ

1. Сколько способов составления таблиц в MS Access 2010 Вы знаете ?
2. Объясните способы соединения таблиц в MS Access 2010?
3. Можно ли при помощи запросов изменить данные в составленных в MS Access 2010 таблицах?
4. Составьте данные об учащихся в трёх таблицах и объедините их в одну при помощи запросов.



ДОМАШНЕЕ ЗАДАНИЕ

1. Посредством раздела **Схема данных** соедините четыре таблицы с различными данными.
2. Составьте данные об учащихся в четырёх таблицах и объедините их в одну при помощи запросов.



28 УРОК. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

1. Создайте БД литературы по предмету “Информатика и информационные технологии”, авторов, год издания и, посредством запроса, произведите поиск соответствующих данных в БД.

2. Создайте БД операционных систем. Сохраните в БД наименование операционных систем, время создания, принадлежность к классу систем и т.п.

3. Создайте БД процессора. В созданной БД сохраните данные о процессоре – год выпуска, название фирмы-изготовителя, оперативная скорость, тип и т.п.

4. Создайте БД типов памяти. В созданной БД сохраните данные о типах памяти, скорости писания и чтения данных, вместимости и т.п.

5. Создайте БД вспомогательных программ. В созданной БД сохраните данные о названиях программ, их функциях, работе в операционной системе.

6. Создайте БД антивирусных программ. В БД сохраните название программ, их функции, порядок работ в операционной системе, данные о рабочем состоянии.

7. Составьте сведения о себе в пяти таблицах и объедините их.

8. Составьте данные о книгах в трёх таблицах и с помощью запросов объедините в одну.



29 УРОК. ПОИСК И ОБРАБОТКА ДАННЫХ ПО УКАЗАННОМУ ШАБЛОНУ В MS ACCESS 2010

Для работы в MS Access 2010 по обработке и переработке сведений можно из раздела **Выделение** меню **Главная**, либо посредством нажатия правой кнопки мыши на таблицу и в последовательности **Текстовые – фильтры – Ровно...** осуществить действия. Сведения в таблицах можно искать по таким данным, как фамилия, имя, адрес. Для удаления найденных сведений выбирается раздел **Фильтр** меню **Главная**.

Упражнение 1. Создание базы данных учащихся 10 класса (фамилия, имя, адрес, номер телефона, год рождения), обработка их и переработка.

Выполнение:

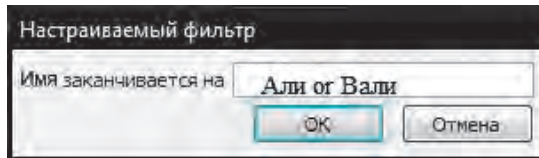
▪ Оформляем сведения об учащихся в MS Access 2010 по указанным параметрам:

Код	Фамилия	Имя	Адрес	Номер телефо	Класс	Дата рожде
1	Усманов	Али	г. Навои, ул. М.Таробий, д.120, кв 2	944826749	10 А	12.07.2003
2	Косимов	Вали	г. Навои, ул. Навои, д.28, кв 23	934567890	10 В	03.03.2003
3	Кувондилов	Фарход	Самаркандская обл. р-н Пахтачи	912346789	10 А	07.02.2002
4	Шодмонов	Даврон	Самаркандская обл. р-н Пахтачи	907865434	10 А	06.10.2003
5	Нажимов	Толиб	Самаркандская обл. р-н Пахтачи	935677742	10 А	07.09.2002
6	Асадов	Феруз	Самаркандская обл. р-н Пахтачи	945671245	10 А	12.04.2004
7	Номозова	Нишонова	г. Навои, ул. Навои, дом 29, кв 19	912345963	10 А	04.05.2003
8	Литфуллаева	Лазиза	г. Навои, ул. Навои, дом 34, кв 2	903452585	10 А	08.09.2002
9	Нишонов	Ахмад	г. Навои, ул. Навои, дом 36, кв 5	945678912	10 А	05.03.2003
10	Ахмедов	Ахрор	г. Навои, ул. Навои, дом 1, кв19	907865432	10 В	04.07.2004
11	Хошимов	Али	г. Навои, ул. Навои, дом 1, кв 6	917652345	10 А	08.04.2004

Для поиска в этой таблице учащихся по имени “Али” после выделения надписи “Али” выполняется следующая последовательность в меню **Главная - Выделение – Равно – Али**. В результате программа находит в таблице сведения об учащихся по имени “Али”. Если требуется найти в таблице учащихся по имени “Али и Вали”, нажатием правой кнопки мыши в ячейку имён учащихся выбирается последовательность **Текстовые фильтры – Заканчивается на..** и в ряд **Настраиваемый фильтр** вводится **“Али от Вали”** и выбирается кнопка **ОК**.

Код	Фамилия	Имя	Адрес	Номер телефо	Класс	Дата рожде
1	Усманов	Али	г. Навои, ул. М.Таробий, д.120, кв 2	944826749	10 А	12.07.2003
2	Косимов	Вали	г. Навои, ул. Навои, д.28, кв 23	934567890	10 В	03.03.2003
3	Кувондилов	Фарход	Самаркандская обл. р-н Пахтачи	912346789	10 А	07.02.2002
4	Шодмонов	Даврон	Самаркандская обл. р-н Пахтачи	907865434	10 А	06.10.2003
5	Нажимов	Толиб	Самаркандская обл. р-н Пахтачи	935677742	10 А	07.09.2002
6	Асадов	Феруз	Самаркандская обл. р-н Пахтачи	945671245	10 А	12.04.2004
7	Номозова	Нишонова	г. Навои, ул. Навои, дом 29, кв 19	912345963	10 А	04.05.2003
8	Литфуллаева	Лазиза	г. Навои, ул. Навои, дом 34, кв 2	903452585	10 А	08.09.2002
9	Нишонов	Ахмад	г. Навои, ул. Навои, дом 36, кв 5	945678912	10 А	05.03.2003
10	Ахмедов	Ахрор	г. Навои, ул. Навои, дом 1, кв19	907865432	10 В	04.07.2004
11	Хошимов	Али	г. Навои, ул. Навои, дом 1, кв 6	917652345	10 А	08.04.2004

В результате в нижеследующем окне появляются сведения об учащихся по имени Али и Вали:



Для удаления упорядоченных данных в меню **Главная** выбирается команда **Фильтр**.

Код	Фамилия	Имя	Адрес	Номер телефо	Класс	Дата рожден
1	Усманов	Али	г.Навои, ул. М.Таробий, д.120, кв 2	944826749	10 А	12.07.2003
2	Косимов	Вали	г.Навои, ул. М.Навои, д.28, кв 23	934567890	10 В	03.03.2003
3	Хошимов	Али	г.Навои, ул. М.Навои, д.1, кв 6	917652345	10 А	08.04.2004

Вместе с этим в MS Access 2010 есть возможность поиска данных при помощи запросов. Этот процесс можно осуществить посредством раздела **Конструктор запрос** в меню **Создание**. При помощи **Конструктор запрос** из одной таблицы можно составить несколько запросов(таблиц). Каждый запрос отделяет какую-то часть данных основной таблицы.

В результате запросов из основной таблицы образуется итоговая таблица. Эта таблица из общих данных включает в себя запрашиваемую часть.

В ходе запросов полученные данные могут упорядочиваться, фильтроваться, объединяться, разделяться, изменяться, в то время как данные таблиц не изменяются. Это считается одним из главных свойств запросов. Ещё одним свойством запросов является выполнение ими итоговых расчётных операций.

В MS Access 2010 содержится несколько типов запросов:

- Избирательный запрос, образцовый запрос;
- Параметральный запрос;
- Вышеуказанный, распространённый запрос - избирательный запрос.

▪ Целью использования этих запросов является составление итоговой таблицы по результатам запросов.

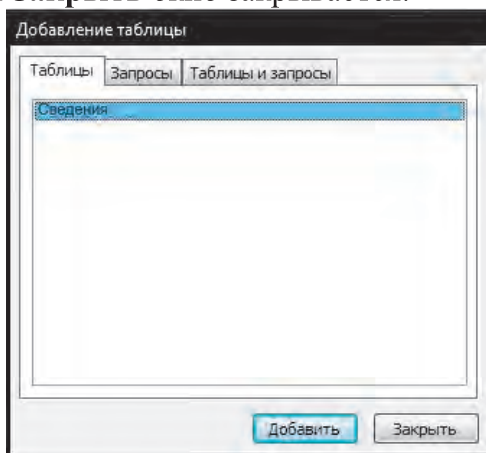
Упражнение 2. Сделайте поиск имени какого-нибудь учащегося из приведённой выше базы данных, указанной в упражнении 1.

Выполнение:

- Составьте базу данных учащихся в указанном виде:

Код	Фамилия	Имя	Адрес	Номер телефо	Класс	Дата рожде
1	Усманов	Али	г. Навои, ул. М.Таробий, д.120, кв 2	944826749	10 А	12.07.2003
2	Косимов	Вали	г. Навои, ул. Навои, д.28, кв 23	934567890	10 В	03.03.2003
3	Кувондиқов	Фарход	Самарқандская обл. р-н Пахтачи	912346789	10 А	07.02.2002
4	Шодмонов	Даврон	Самарқандская обл. р-н Пахтачи	907865434	10 А	06.10.2003
5	Нажимов	Толиб	Самарқандская обл. р-н Пахтачи	935677742	10 А	07.09.2002
6	Асадов	Феруз	Самарқандская обл. р-н Пахтачи	945671245	10 А	12.04.2004
7	Номозова	Нишонова	г. Навои, ул. Навои, дом 29, кв 19	912345963	10 А	04.05.2003
8	Литфуллаева	Лазиза	г. Навои, ул. Навои, дом 34, кв 2	903452585	10 А	08.09.2002
9	Нишонов	Ахмад	г. Навои, ул. Навои, дом 36, кв 5	945678912	10 А	05.03.2003
10	Ахмедов	Ахрор	г. Навои, ул. Навои, дом 1, кв19	907865432	10 В	04.07.2004
11	Хошимов	Али	г. Навои, ул. Навои, дом 1, кв 6	917652345	10 А	08.04.2004

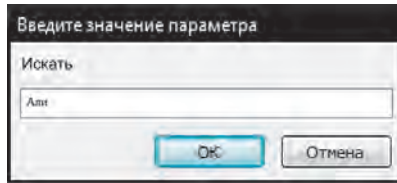
- Выбирается раздел **Конструктор запрос** в меню **Создание**;
- Выбирается кнопка **Добавить** в окне **Добавление таблицы** и посредством кнопки **Закреть** окно закрывается:



- Появившееся окно заполняем в следующем порядке:

Поле:	Код	Отчество	Имя	Адрес	Номер телефона	Класс	Дата рождения
Имя таблицы:	Сведение	Сведение	Сведение	Сведение	Сведение	Сведение	Сведение
Сортировка:							
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:							
или:							

- Если в этом окне производится поиск по какому-либо параметру, в этом столбце отмечается ряд **Условие отбора** и затем вводится какое-либо предложение в кавычках “[]”.
- Например, для создания поискового запроса по имени учащихся вводится слово “[Поиск]” в столбец “Имя”, в меню выбирается команда **Выполнить**. В результате появляется следующее окно:



Здесь в окно мы вводим имя искомого учащегося, выбираем кнопку **ОК**. В результате, в таблице появляется сообщение об учащихся по имени “Али”:

Код	Фамилия	Имя	Адрес	Номер телефони	Класс	Дата рожде
1	Усманов	Али	г. Навои, ул. М.Таробий, д.120, кв 2	944826749	10 А	12.07.2003
11	Хошимов	Али	г. Навои, ул. Навои, дом 1, кв 5	917652345	10 А	08.04.2004



ЗАПОМНИТЕ !

В MS Access 2010 для поиска и обработки базы данных по шаблону выбирается раздел **Выделение** в меню **Главная**.



ВОПРОСЫ И ЗАДАНИЯ

1. Как осуществляется поиск в базе данных по указанному шаблону?
2. Как удаляются отобранные данные?
3. Сделайте сортировку и поиск трёх фамилий в таблице одновременно.
4. Создайте базу данных учащихся и воспользуйтесь следующей последовательностью:
 - а) Создайте поисковую БД по фамилии;
 - б) Создайте поисковую БД по номеру телефона;
 - в) Создайте поисковую БД по классу.



ДОМАШНЕЕ ЗАДАНИЕ

1. Создайте базу данных по учебникам для 10-го класса и создайте поисковую программу по авторам.
2. Загрузите таблицу данных учебников для 10-го класса и посредством команды **Найти** в меню найдите название учебника, год издания, авторов.



30 УРОК. СОЗДАНИЕ БАЗЫ ДАННЫХ В MS ACCESS 2010 И ИСПОЛЬЗОВАНИЕ ФОРМ ПРИ РЕДАКТИРОВАНИИ

Формы используются для ввода новых данных и просмотра имеющихся.

Форма имеет вид электронного бланка, имеющего поля для ввода данных. Введенные в эти поля данные соединяются непосредственно с таблицей БД.

В целом, даже не образуя **формы**, можно вводить данные в основную таблицу. Однако введение данных в таблицу посредством **формы** намного удобнее. Для введения данных в таблицу составляются различные **формы**.

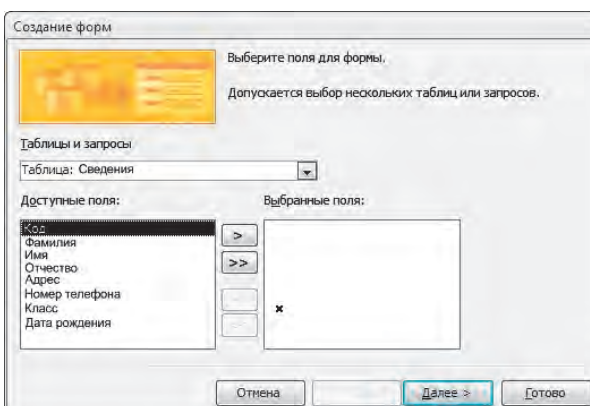
Имеется два способа ввода новых данных в БД MS Access 2010:

- Непосредственный ввод в таблицу БД;
- Ввод посредством специально созданных **форм**.

Упражнение 1. Создайте **форму** для базы данных об учащихся и введите данные посредством **формы**.

Выполнение:

- Создается таблица под названием «Справка», выделяется и из меню **Создание** выбирается пункт **Мастер форм**;
- Из открывшегося окна выбирается знак “>>” посредством кнопки **Далее** переходим к следующему окну;



▪ В окне имеется возможность изменить окно формы до четырех видов:

- в один столбец;
- ленточный;
- табличный;
- выровненный.

▪ Из данного окна отмечается пункт **в один столбец** и затем нажимается кнопка **Далее**;

▪ Из следующего окна выбирается пункт **Открыть форму для просмотра и ввода данных**;

▪ Нажимаем кнопку **Готов**.

В результате получается следующее окно:

The screenshot displays a software interface with a form titled 'Сведения' (Information) and a dialog box titled 'Создание форм' (Form Creation). The 'Сведения' form contains the following fields:

Код	1
Фамилия	Аслонов
Имя	Ахмад
Адрес	Шодмонович
Номер телефона	944826749
Класс	10 А
Дата рождения	14.07.2004

The 'Создание форм' dialog box prompts the user to 'Выберите внешний вид формы:' (Select the external view of the form:). It offers four options:

- В один столбец (One column)
- Ленточный (Ribbon)
- Табличный (Table)
- Выровненный (Aligned)

At the bottom of the dialog box, there are buttons for 'Отмена' (Cancel), '< Назад' (Back), 'Далее >' (Next), and 'Готово' (Done).

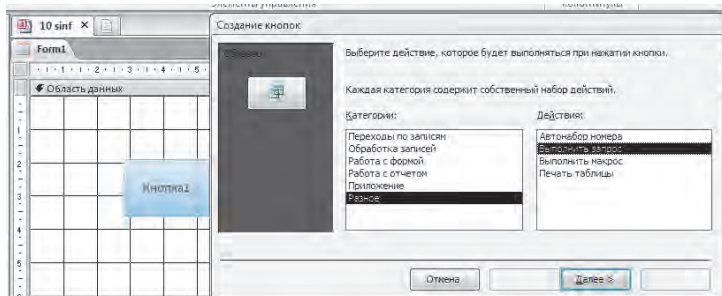
С помощью полученного окна формы имеется возможность ввода данных в таблицу «Справка».

Упражнение 2. Посредством формы сделайте поиск сведений об учащихся, данный в 29-занятии.

Выполнение:

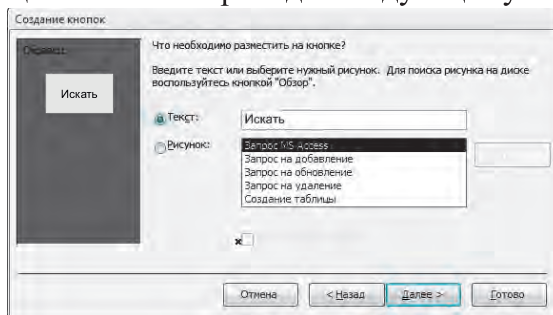
▪ Выделяем таблицу, сохранённую под именем «Искать с помощью имени», и из строки меню выбираем раздел **Конструктор-форма**;

- Из раздела **Конструктор** выбираем команду **Кнопка** и размещаем его в окне формы. В результате получится следующее окно формы:

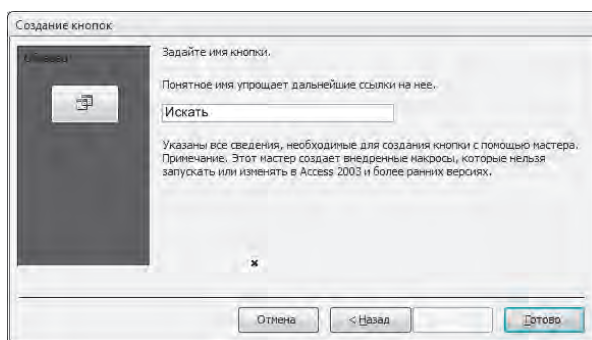


- Выбирается команда **Разное** из первого окна и, выделив команду **Выполнить запрос** из второго окна, нажимаем кнопку **Далее**;

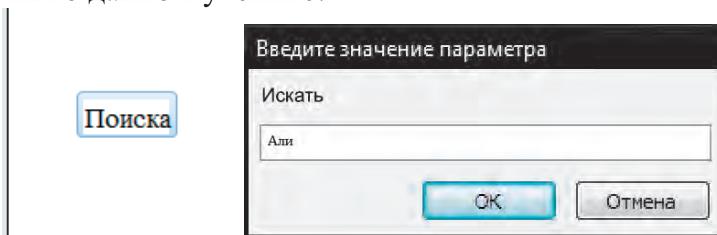
- В строку **Текст** вводится надпись «Поиск» и с помощью выбора кнопки **Далее** осуществляется переход к следующему окну:



- В строку **Понятное имя упрощает дальнейшие ссылки на нее** вводится текст «Поиск» и выбирается кнопка **Готов**;



Для запуска созданной программы из раздела **Режим** выбирается пункт **Режим формы**. В результате этого созданная программа переходит в рабочий режим. Посредством выбора кнопки «Поиска» в окне с помощью имени нужного ученика в таблице можно получить сведения о данном ученике:



В результате на экране в следующем виде отображаются данные об ученике по имени **Али**:

Код	Фамилия	Имя	Адрес	Номер телефо	Класс	Дата рожде
1	Усманов	Али	г. Навои, ул. М.Таробий, д.120, кв 2	944826749	10 А	12.07.2003
11	Хошимов	Али	г. Навои, ул. Навои, дом 1, кв 6	917652345	10 А	08.04.2004

Если необходимо ввести дополнительные изменения в окно формы, то это можно осуществить посредством выбора пункта **Конструктор** из раздела **Режим**.



ЗАПОМНИТЕ!

Имеется два способа ввода новых данных в БД MS Access 2010:

1. Непосредственный ввод в таблицу БД;
2. Ввод посредством специально созданных **форм**.



ВОПРОСЫ И ЗАДАНИЯ

1. Объясните последовательность создания форм в MS Access 2010.
2. Сколько способов ввода новых данных в БД существует в MS Access 2010?
3. Как размещаются кнопки в окне формы?
4. Создайте форму, используя **Мастер форм**.
5. Посредством созданного окна формы введите данные в таблицу.



ДОМАШНЕЕ ЗАДАНИЕ

1. Создайте БД литературы по математике, их авторов, времени издания и посредством запроса найдите соответствующие данные в БД.
2. Создайте кнопку для распечатывания на бумагу данных таблицы посредством формы.



31 УРОК. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

1. Создайте таблицу об учащихся класса (фамилия, имя, телефон).
2. В созданной таблице осуществите сортировку по фамилиям, именам, номерам телефонов.
3. Созданную таблицу упорядочите по фамилиям учащихся.
2. Создайте поисковые запросы для упражнения, приведенного в 29-30-занятиях по следующим разделам: фамилия; адрес; дата рождения; номер телефона; класс.
3. Создайте форму, используя **Конструктор форм**.
4. Вставьте диаграмму в окно созданной формы.
5. Вставьте рисунок в форму.



32 УРОК. ВЫПОЛНЕНИЕ МАТЕМАТИЧЕСКИХ ОПЕРАЦИЙ В MS ACCES 2010

Создав в MS Access 2010 базу данных, можно совершать некоторые математические действия с данными. Данные действия можно совершить с помощью математических и статистических функций. Математические действия можно совершить посредством раздела **Конструктор** меню **Создание**.

И в MS Access 2010, также как в MS Excel 2010, можно совершить математические действия, используя при этом математические и статистические функции, логические действия.

Математические действия в программе MS Access 2010

1.	+	Сложение
2.	-	Вычитание
3.	*	Умножение
4.	/	Деление
5.	^	Возведение в степень

Действия математических отношений в MS Access 2010

1.	>	Больше
2.	<	Меньше
3.	=	Равно
4.	>=	Больше или равно
5.	<=	Меньше или равно
6.	<>	Не равно

Математические функции в MS Access 2010

п/н	Функция	Предназначение
1.	Abs	Вычисляет модуль числа
2.	sqr	Вычисляет квадратный корень числа
3.	cos	Вычисляет косинус числа
4.	sin	Вычисляет синус числа
5.	tan	Вычисляет тангенс числа
6.	atn	Вычисляет арктангенс числа
7.	log	Вычисляет логарифм числа
8.	Rnd	Выбор любого числа в промежутке от 0 до 1
9.	int	Целая часть натурального числа

Статистические функции в MS Access 2010

п/н	Функция	Предназначение
1.	max	Находит самое большое значение в столбце

2.	min	Находит самое меньшее значение в столбце
3.	Sum	Вычисляет сумму значений столбца

Логические действия в MS Access 2010

t/r	Функция	Предназначение
1.	NOT	Отрицание
2.	AND	И
3.	OR	Или

Упражнение 1. Получите сумму чисел, данных в трёх столбцах в четвёртом столбце.

Выполнение:

- Таблица создается следующим образом:

Имя поля	Тип данных
A	Числовой
B	Числовой
C	Числовой
A+B+C	Числовой

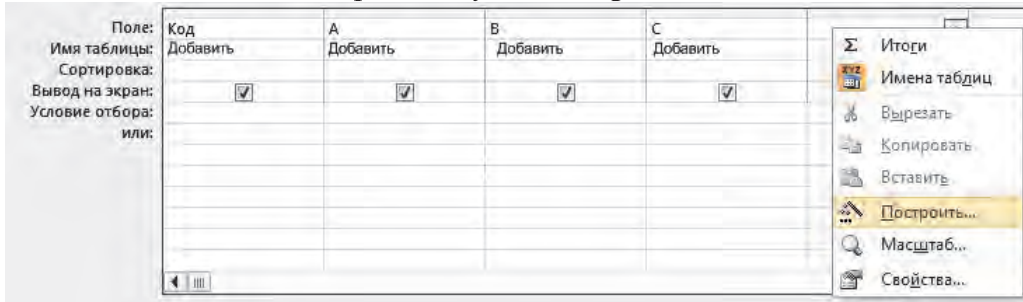
- Сохраним созданную таблицу под именем «Сложить»;
- Загружается сохранённая таблица и заполняется в следующем порядке:

Код	A	B	C	A+B+C
1	7	7	7	7
2	5	5	5	5
3	4	4	4	4
4	3	3	3	3
5	2	2	2	2
6	1	1	1	1

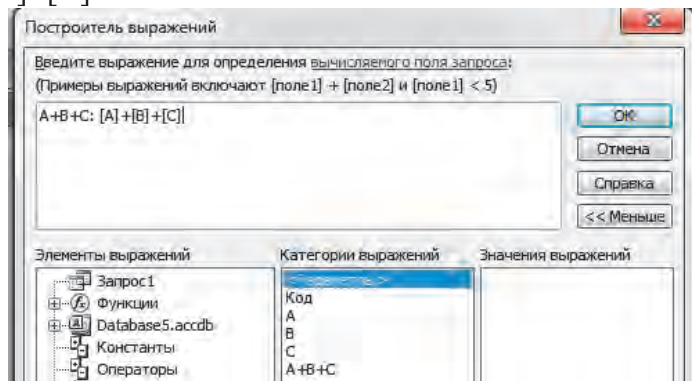
- Закрыв заполненную таблицу, из строки меню выбирается последовательность **Создания-Конструктор запросов**;
- Выделяется таблица под именем «Сложение» и нажимается кнопка **Добавить**.
- Рабочее окно заполним в следующем порядке:

Поле:	Код	A	B	C	
Имя таблицы:	Добавить	Добавить	Добавить	Добавить	
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:					
или:					

■ Нажимая правую кнопку мыши на следующем столбце таблицы из контекста меню выбирается пункт **построить**:



■ Полученное окно состоит из двух частей, в первом окне вводятся формулы математических расчётов. Во втором окне – математические и статистические функции и методы использования логических действий. Для получения в четвёртом столбце суммы чисел, данных в трёх столбцах, вводим в первое окно следующий код: $A+B+C: [A]+[B]+[C]$



■ Сохранив выполненные действия под именем «Сложение чисел» и загрузив таблицу, получим следующее:

Код	A	B	C	A+B+C
1	7	7	7	21
2	5	5	5	15
3	4	4	4	12
4	3	3	3	9
5	2	2	2	6
6	1	1	1	3

Сумма чисел трёх столбцов отражена в четвёртом.



ЗАПОМНИТЕ!

В MS Access 2010 математические расчёты можно совершить посредством раздела **Конструктор запросов** меню **Создание**.



ВОПРОСЫ И ЗАДАНИЯ

1. Какие математические действия имеются в MS Access 2010?
2. Какие относительные действия имеются в MS Access 2010?
3. Запишите математические функции в MS Access 2010.
4. Запишите статистические функции в MS Access 2010.
5. Запишите логические действия в MS Access 2010.
6. Создайте таблицу для вычисления произведения данных четырёх чисел.
7. Создайте таблицу для вычисления корней чисел.



ДОМАШНЕЕ ЗАДАНИЕ

1. Создайте таблицу для вычисления косинуса чисел столбца.
2. Создайте таблицу для вычисления среднего значения чисел столбца.



33 УРОК. ЗАДАНИЯ ДЛЯ ПОВТОРЕНИЯ

1. Расскажите о базе данных в MS Access 2010.
2. Расскажите о составляющих БД.
3. Составьте в MS Access 2010 БД, хранящую данные об учащихся класса.
4. Из таблицы MS Access 2010 об учащихся класса найдите четырёх учащихся по фамилиям.

5. Упорядочите данные об учащихся в таблице MS Access 2010 по датам рождения.
6. Составьте три таблицы с данными о книгах и по запросу соедините в одну таблицу.
7. Составьте пять таблиц с данными о книгах и по запросу соедините в одну таблицу.
8. Составьте три таблицы с данными о себе и по запросу соедините в одну таблицу.
9. Создайте БД для поиска данных об учащихся класса в MS Access 2010.
10. Создайте программу MS Access 2010 для определения учащихся, пропускающих большое количество занятий.
11. Создайте таблицу для вычисления вычитания двух чисел по столбцу.
12. Создайте таблицу для вычисления квадрата чисел в столбце.
13. Создайте таблицу для вычисления синуса чисел столбца.
14. Выполните следующие действия:
 а) $3*(5-7+8*6+9)$. б) $4*(7-5+12*3-4)$.



34 УРОК. КОНТРОЛЬНАЯ РАБОТА

1 вариант

1. Создайте базу данных о девочках вашего класса и сделайте сортировку данных.

2. Создайте в MS Access 2010 данную цветную таблицу:

Код	Фамилия	Имя	Инфор- матика	Физи-ка	Мате- матика	Химия	Исто- рия
1.	Каримов	Асад	3	5	4	3	5
2.	Мустафин	Дамир	4	4	3	4	4
3.	Сергеев	Алексей	2	3	4	4	2
4.	Ли	Александр	4	3	3	2	5

3. Объясните способ использования функции **max** в MS Access 2010.

4. Выполните следующие действия в MS Access 2010:

$$5^2 + 6 - 4 * 6 + \frac{9}{4} .$$

2 вариант

1. Создайте четыре таблицы и соедините их.

2. Создайте в MS Access 2010 данную цветную таблицу:

Код	Фамилия	Имя	Дата рождения	Адрес	Номер телефона
1.	Каримов	Асад	25.06.2002	Навоийская область	678-34-09
2.	Мустафин	Дамир	6.09.2001	Город Ташкент	234-56-01
3.	Сергеев	Алексей	5.03.2003	Бухарская область	234-12-32
4.	Ли	Александр	6.09.2001	Самаркандская область	345-00-12

3. Объясните способ использования функции **min** в MS Access 2010.

4. Выполните следующие действия в MS Access 2010: $4 - 8 * 5^2 + 9$.

3 вариант

1. Создайте практическую поисковую программу с помощью фамилий учащихся класса.

2. Создайте в MS Access 2010 данную таблицу:

Код	Фамилия	Имя	Дата рождения	Адрес	Номер телефона
1.	Каримов	Асад	25.06.2002	Навоийская область	678-34-09
2.	Мустафин	Дамир	6.09.2001	Город Ташкент	234-56-01
3.	Сергеев	Алексей	5.03.2003	Бухарская область	234-12-32
4.	Ли	Александр	6.09.2001	Самаркандская область	345-00-12

3. Объясните способ использования функции **tan** в MS Access 2010.

4. Выполните следующие действия в MS Access 2010:

$$5^3 + 7 + 4 * 6 - \frac{5}{7} .$$

4 вариант

1. Создайте окно формы для ввода данных об учащихся.
2. Создайте в MS Access 2010 данную таблицу:

Код	Фамилия	Имя	Информатика	Физика	Математика	Химия	История
1.	Каримов	Асад	3	5	4	3	5
2.	Мустафин	Дамир	4	4	3	4	4
3.	Сергеев	Алексей	2	3	4	4	2
4.	Ли	Александр	4	3	3	2	5

3. Объясните способ использования функции **log** в MS Access 2010.

4. Выполните следующие действия в MS Access 2010:

$$2^3 - 6 + 4 * 6 + \frac{7}{4} .$$

Borland

Delphi[™]
Enterprise

ГЛАВА III. СОЗДАНИЕ ПРИЛОЖЕНИЙ В СРЕДЕ DELPHI 7





**ВНИМАТЕЛЬНО ПРОЧИТАВ ЭТУ ГЛАВУ, ВЫ ОВЛАДЕЕТЕ
СЛЕДУЮЩИМИ ЗНАНИЯМИ, УМЕНИЯМИ И КОМПЕТЕНЦИЯМИ:**

- Современные методы создания приложений;
- Интерфейс среды Delphi 7;
- Создание окна приложения в среде Delphi 7;
- Использование кнопки управления в приложении;
- Окно **ShowMessage** в приложениях;
- Вставка данных в окно приложения;
- Доступность и видимость объектов управления;
- Ввод данных в приложение;
- Изменение типа данных;
- Использование флажков в приложении;
- Использование группы радиокнопок в приложении;
- Объекты **ListBox** и **ComboBox**;
- Объект управления **Memo**;
- Работа с графикой в среде Delphi 7;
- Объект **Timer** и работа с ним;
- Вставка других объектов в рисунок;
- Применение графики и анимационных возможностей в приложении.

Borland Delphi 7



35 УРОК. СОВРЕМЕННЫЕ МЕТОДЫ СОЗДАНИЯ ПРИЛОЖЕНИЙ

1. Современные приложения. Известно, что современные компьютеры оснащены операционной системой. С их помощью приложения операционной системы выполняют практические задачи. В качестве примера можно привести офисные приложения: **MicroSoft Word** (кратко  – MS Word),  – MS Excel,  – MS Access,  – MS Power Point.

Особенностью данных приложений является интерактивность и с их помощью можно выполнить сотни и тысячи операций. Как создаются такие сложные приложения?

Программные средства создания приложений на сегодняшний день обладают огромными возможностями. Знакомство с ними мы начнём с истории программирования.

2. Этапы развития программирования. Историю развития средств разработки приложений (практических программ) можно разбить на следующие этапы (см. рисунок):

В первых компьютерах программы непосредственно в виде последовательности команд микропроцессора (машинного кода). Это в свою очередь усложняло процесс программирования, требовало много сил и времени, затрудняло устранение ошибок в программах.

Чтобы облегчить эту задачу, вводились короткие имена для команд микропроцессора и специальные программы переводили их в машинные коды (команды микропроцессора). Полученный код программы выполнялся непосредственно компьютером. Такое программирование носило название программирования на языке Assembler.



После начала массового выпуска компьютеров (компьютеры третьего поколения) у них появилась **операционная система**. Для таких компьютеров были созданы **высокоуровневые языки программирования**. Теперь программирование уже не зависело от машинного кода компьютера. Язык программирования всё больше становился схожим с языком человеческого общения. К примеру, в этих языках можно было использовать предложения типа **если $x > 0$, то $y = \ln(x)$** . Перевод текста программы в машинный код, понятный компьютеру, осуществляла специально разработанная программа, которая называлась **транслятором**. В результате программирование стало более простым и расширился круг задач, решаемых при помощи компьютеров.

На следующем этапе программирования задачи, решаемые с помощью компьютера, были разбиты на несколько меньших и лёгких задач. При необходимости и их можно было разбить на более мелкие задачи. Это дало возможность заняться разработкой одной программы одновременно несколькими (иногда десяткам) программистам. Качество создаваемых программных средств, то есть приложений, возросло, программы усложнились, приобрели больше возможностей. Такое программирование носило название **структурного программирования**. Знакомый нам язык программирования **Turbo Pascal** входит в их число. Разлагая задачи на более мелкие, программирование осуществлялось с помощью процедур и функций программирования.

Следующий этап программирования называется **объектно-ориентированное программирование**. Это в первую очередь связано с появлением графических операционных систем (в них также входит широко распространённая сейчас Windows), созданными для упрощения и облегчения работы на персональных компьютерах.

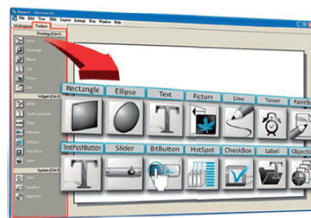
Мы знаем, что информация – это данные с известным способом их разработки. В Turbo Pascal информация делится на переменную и постоянную. Способы их обработки (алгоритмы обработки) могут быть в виде отдельных процедур и функций.

В объектно-ориентированном программировании появилась возможность объединить эти два элемента, что ранее не удавалось. Разница между данными и их обработкой исчезла. В результате, обработка информации стала ещё легче. Это дало возможность заниматься одной программой одновременно десяткам, иногда сотням и даже тысячам программистов. Возможности созданных программных средств резко возросли.

Разрабатываемое программное средство можно рассматривать как информацию. Значит, при создании программного кода также можно использовать объекты. К примеру, каждая программа имеет своё окно. Окно программы имеет свои параметры, как, например, высота и ширина (параметры объекта окна). С помощью таких методов обработки (методы объекта окна) как создание, закрытие, изменение места и параметров можно облегчить работу с окном программы.

Если создан объект – окно, работа с окном программы будет лёгкой и в виде использования способов объекта в необходимой степени, а также установления свойств данного объекта будет очень доступной. В результате программирование ещё больше облегчилось, качество создаваемых программ повысилось.

Теперь для создания приложений стало достаточно того, чтобы ввести созданные другими готовые объекты в программу и установить их свойства в требуемом виде. Данное программирование носит название **визуального программирования** и программирование стало более похожим на сборки игрушек Лего.



В настоящее время все больше распространяется **обобщённое программирование**. Суть этого метода заключается в следующем. Пусть в программе имеются такие объекты, как треугольник, квадрат, многоугольник и их свойства, площадь. Вместо того, чтобы использовать эти свойства по отдельности, мы можем использовать такое свойство, как площадь геометрической фигуры. Сама программа определяет: о какой геометрической фигуре идет речь и, используя

необходимое свойство соответствующей геометрической фигуры, находит ее свойство. В итоге программирование стало ещё легче и дешевле. Теперь разрабатываемую программу можно одновременно создавать для любой операционной системы, а также с интерфейсом на любом языке. Их адаптация не потребует много времени и труда.

3. Современные среды программирования. Сейчас в мире насчитывается более десяти миллионов программистов, из них два миллиона – профессионалы, остальные – программисты-любители. Конечно же, их инструменты, то есть используемые ими среды программирования, различаются друг от друга.

В распространённых сегодня средах программирования в основном используются три языка программирования: **Ci**, **BASIC**, **Pascal**. Если профессионалы и опытные любители в основном используют **C** (Си) и её последующие варианты **C++** и **C#**, то большинство пользуется языком **Pascal** для решения собственных задач. **BASIC** (вернее, **Visual BASIC**) используется для автоматизации работы в графических редакторах, в офисных приложениях компании **Майкрософт**, а также в ряде продукций других компаний.

В условиях сильной конкуренции и с целью привлечения пользователей все распространённые среды программирования дают возможность одновременного программирования в нескольких языках. Большинство создаваемых в последнее время языков программирования ориентированы на тех, кто пользуется одним из вышеприведённых языков. К примеру, широко распространённые языки программирования **Java**, **Java Script**, **ASP**, **PHP**, **Python** похожи на **C+** и **C#**, языки программирования пакетов компьютерной математики **MatLab**, **MathCAD**, **Maple** на **Pascal**, язык программирования приложения **MacroMedia Flash** – для создания анимационной графики **Action Script** на **Visual BASIC**.

4. Среда программирования Delphi. **Delphi** (читается как Дельфи) это название греческого города, где располагался храм, в котором жил известный в древнем мире прорицатель. Как и он, среда программирования стала такой же известной. За ее четверть века было создано 25 вариантов, которые подразделяются на следующие три группы:

Delphi 1 – Delphi 8 (1995-2003гг.);
Delphi 2005 – Delphi 2010 (2005-2009гг.);
Delphi XE 1 – Delphi XE 10 (2010-2017гг.).

Хотя у них всех основным (иногда единственным) языком программирования является **Object Pascal**, во многих можно писать программы на **C++**, **Assembler**, в последних вариантах в **Java**, в отдельных версиях даже **PHP**. Последние версии дают возможность разрабатывать наряду с приложениями **Windows** также и мобильные для приложения **Android** и **IOS**.



ЗАПОМНИТЕ !

1. Визуальное программирование считается самым современным методом программирования.
2. В мире несколько миллионов программистов пользуется средой программирования **Delphi**.



ВОПРОСЫ И ЗАДАНИЯ

1. Перечислите этапы развития программирования.
2. Что вы знаете о современных средах программирования?
3. Создайте в **Turbo Pascal** программы с участием а) оператора присвоения;
б) условного оператора.
4. Создайте в **Turbo Pascal** программы с участием а) оператора разветвления;
б) оператора цикла.



ДОМАШНЕЕ ЗАДАНИЕ

1. Сколько вариантов среды программирования **Delphi** разработано и на сколько групп они делятся?



36 УРОК.

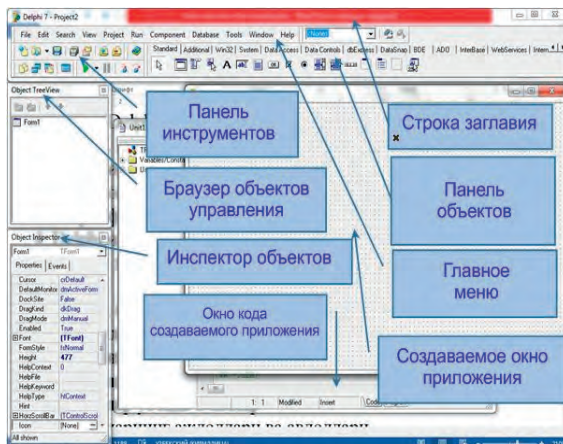
СРЕДА ПРОГРАММИРОВАНИЯ DELPHI

1. Интерфейс Delphi состоит из следующих разделов:

1. Строка заголовка.
2. Основное меню.
3. Панель инструментов.
4. Панель объектов (компонентов) управления.

5. Браузер объектов управления.
6. Инспектор объектов (элементов) управления.
7. Окно (форма) разрабатываемого приложения.
8. Окно кода приложения.

Строка заголовка главного окна среды **Delphi** имеет стандартный вид. Где указаны названия приложения, названия проекта, создаваемого в нём, расположены три кнопки для скрытия, изменения размера и закрытия окна приложения.



2. Основное меню Delphi состоит из следующих 11 разделов:

- | | |
|----------------------------------|---|
| 1. File (Файл). | 7. Component (Объекты управления). |
| 2. Edit (Редактирование). | 8. Database (База данных). |
| 3. Search (Поиск). | 9. Tools (Инструменты). |
| 4. View (Вид). | 10. Window (Окна). |
| 5. Project (Проект). | 11. Help (Помощь). |
| 6. Run (Ввод в действие). | |

3. **Панель инструментов и объектов (компонент) Delphi.** В панели инструментов расположены наиболее часто используемые команды, что дает возможность использовать их не входя в главное меню. В панели инструментов находятся более десяти кнопок, их можно дополнить новыми или убрать ненужные.

Компонентами являются различные элементы (объекты) управления, предназначенные для вставки в создаваемое приложение. Визуальное программирование осуществляется посредством вставки данных элементов управления в окно приложения и изменения его свойств. В **Delphi** имеется более сотни элементов управления и они распределены в двадцати с лишним вкладках панели компонент.

4. **Окна Delphi.** Кроме главного окна, в Delphi имеется ряд окон, расположение которых можно изменить внутри главного окна, скрыть,

вывести на экран, изменить свойства. Первое окно называется **окном формы** и в нём отображается вид главного окна создаваемого приложения. Изменением свойств данного окна можно изменять свойства окна приложения. С помощью мыши в него переносятся нужные компоненты из панели компонент и формируется их взаимное расположение. Данное окно мы назовём окном нового приложения или, просто, **окном приложения**. Работа с ним предусматривает изменение свойств окна, вставку в него новых элементов управления и изменения их свойств, а также изменение расположения этих элементов.

Вторым важным окном является окно кода программы (на самом деле – текста программы) и изменения в окне приложения автоматически отображаются на нём. К примеру, если мы изменим свойства окна приложения, осуществляющий эти изменения оператор программы включается в его текст. Если в окно приложения вставлены новые элементы управления, соответствующие этим элементам операторы автоматически вставляются в текст программы.

Главная функция программиста в окне кода программы состоит в указании необходимости соответствующего ответа программы элементами управления при соответствующих ситуациях. Для каждой ситуации создаётся отдельная процедура и написание текста данной процедуры возлагается на программиста.

В визуальном программировании главной задачей предназначения программиста является использование элементов управления, то есть необходимых свойств и методов этих объектов. Здесь главное – быстрое найти нужное из десятков свойств и методов более ста объектов. В этом ему поможет **Инспектор объекта**.

При выборе в окне приложения какого-то объекта, все доступные свойства данного объекта отображаются в инспекторе объекта. В результате нет необходимости в поиске и запоминании нужных свойств и их значений.

В создании больших и сложных приложений непомерно увеличивается количество используемых объектов. Более простой способ нахождения нужного объекта приложения – использование

специального окна **Object TreeView** (Список объектов в виде дерева). Это окно также называют **Браузером объектов**. Один объект может включать в себя другой. Именно из-за этого список объектов в виде дерева является самым удобным.

В **Delphi** существует ещё несколько главных и достаточно много диалоговых окон. С ними мы познакомимся позже, по мере необходимости.

5. Создание и сохранение проектов в Delphi. Создаваемые в **Delphi** приложения называются **проектами** и хранятся в специальной папке **Projects**. Они являются более сложными, чем приложения, создаваемые в **Pascal** и, для их сохранения понадобится несколько файлов.

Для хранения программы, созданной в **Pascal**, достаточно одного файла и все программы можно хранить в одной папке. Но в **Delphi** это невозможно, потому что одноимённые файлы разных проектов сохраняются вместо других. В результате проекты работают неправильно и возникают конфликты между ними. Поэтому для работы с новым проектом необходимо создать новую папку для данного проекта.

При создании нового проекта в **Delphi** на основе существующего нужно быть осторожным. При внесении изменений в проект данные изменения автоматически будут сохраняться в файлах проекта и возможности восстановления предыдущего проекта не будет. Именно поэтому при создании нового проекта на основе существующего необходимо копировать существующий проект в папку с новым именем.

Договоримся называть проекты **Delphi** приложениями.



ЗАПОМНИТЕ !

В **Delphi** существуют сотни готовых объектов управления.



ВОПРОСЫ И ЗАДАНИЯ

1. Что расположено в панели компонент **Delphi**?
2. Какую задачу выполняет окно приложения **Delphi**?

3. Чем отличаются приложения **Delphi** от программ в **Pascal**?
4. Создайте две программы в **Turbo Pascal** со стандартными функциями и процедурами.
5. Создайте две программы в **Turbo Pascal** с функциями и процедурой.



ДОМАШНЕЕ ЗАДАНИЕ

1. Запишите разделы интерфейса **Delphi**.
2. Запишите разделы главного меню **Delphi**.



37 УРОК.

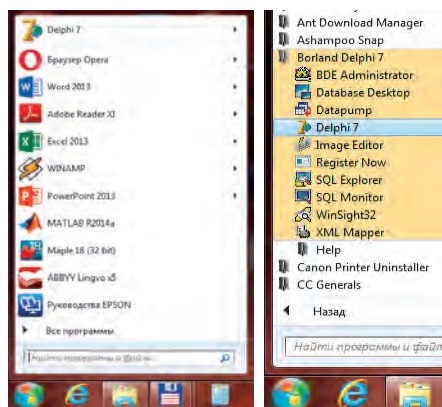
ПРИЛОЖЕНИЕ И ЕГО ОКНО

1. Запуск Delphi и создание в нем приложения. Для запуска **Delphi** наводим курсор мыши на **Пуск** на рабочем столе **Windows** и нажимаем. Из появившегося меню выбираем пункт **Все программы**. Из всплывшего меню активизируется пункт **Borland Delphi 7**.

В результате в списке появятся новые пункты, относящиеся к **Delphi**. Из них выбираем **Delphi 7**. На экране появятся окна среды программирования **Delphi** и создается новое приложение.

Если в пункте **Delphi 7** основного меню нажать вместо левой кнопки мыши правую кнопку и вызвав меню контекста, выбрать пункты **Закрепить в меню Пуск** или **Закрепить на панели задач** появится ярлык **Delphi 7**, что облегчит его запуск. Если пункт **Delphi 7** меню перетащить с помощью мыши, а оставить на пустом месте рабочего стола и отпустить кнопку мыши, ярлык останется на рабочем столе.

Для создания нового приложения в **Delphi** выбираем пункт **New** в разделе **File** его основного меню и из полученного раздела меню выбираем пункт **Application**. Если предыдущее приложение не сохранено, **Delphi** предупреждает об этом и предложит сохранить его. Можно сохранить старое приложение или перейти на новое, не сохранив предыдущее.



2. Загрузка и сохранение приложения в Delphi. Существует несколько методов сохранения приложения в Delphi. Все эти команды расположены в разделе **File** главного меню.

Ими являются:

1. **Save**
2. **Save as ...**
3. **Save Project as ...**
4. **Save All.**

Первая команда **Save** служит для сохранения приложения под текущим именем. Вторая сохраняет под новым именем. Третья сохраняет приложение в новом месте, а четвертая служит для сохранения всех файлов приложения.

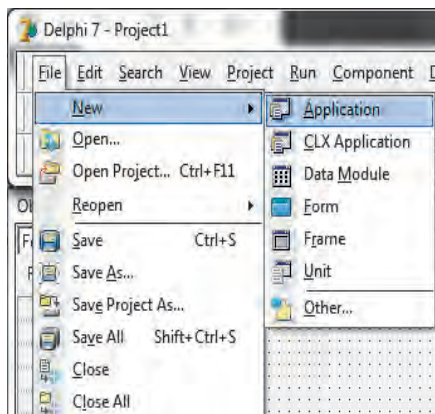
3. Установка окна приложения и его свойств. Основными свойствами окна приложения являются его размеры и место на экране компьютера. Свойства окна приложения **Delphi** можно изменить с помощью мыши, передвигая нижний правый угол окна приложения. Но есть и другой путь.

Наводим мышь на окно приложения (**Form1**) и нажимаем левую кнопку, в окне Инспектора объекта (**Object Inspector**) меняем значения нужных параметров. В данном окне приведено в алфавитном порядке более 50 свойств. Если нужное свойство не видно на экране, его можно вывести на экран сдвигом вертикальной прокрутки с помощью мыши.

Ширину окна приложения меняем, вводя значение 400 в свойства **Width**. Для этого выбираем с помощью мыши свойство **Width**, вводим число 400. Таким же образом высоту окна приложения с помощью свойства **Height** приравниваем на 300. Данные действия кратко обозначим следующим образом и впоследствии их широко используем: `Form1.Width:=400;`

`Form1.Height:=300;`

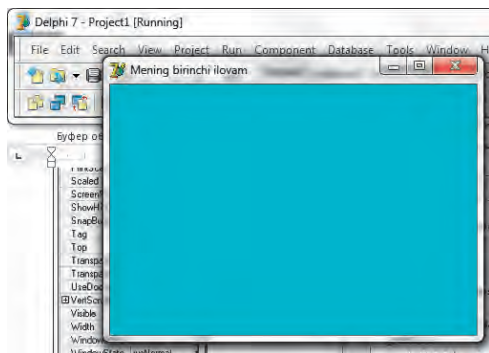
Чтобы изменить название заголовка из стандартного **Form1** в **Мое первое приложение** выполним:




`Form1.Caption:= 'Мое
первое приложение;`


С помощью `Form1.
Color:='ClAqua'` ИЗМЕНИМ
фон окна приложения на голубой.

С помощью свойств `Form1.
Left:=100` и `Form1.Top:=50`
устанавливаем полосу слева и
сверху окна приложения равным
100 и 50.



Для просмотра выполненных действий открываем приложение.
Это можно сделать тремя способами.

1. Выбор пункта **Run** раздела **Run** главного меню.
2. Нажатие на кнопку  в панели инструментов.
3. Нажатие клавиши **F9**.

Для окончания работы с приложением нажимаем на кнопку  в его верхнем правом углу.



ЗАПОМНИТЕ !

Form1 – это окно создаваемого приложения, имеет более 50 свойств.



ВОПРОСЫ И ЗАДАНИЯ

1. Перечислите основные свойства окна приложения **Delphi**.
2. С помощью каких методов можно открыть приложение **Delphi**?
3. Введите свое имя в заголовок приложения **Delphi**.
4. Создайте в **Delphi** окно приложения с параметрами 600 на 800.
5. Создайте в **Delphi** окно приложения со светло-жёлтым фоном.



ДОМАШНЕЕ ЗАДАНИЕ

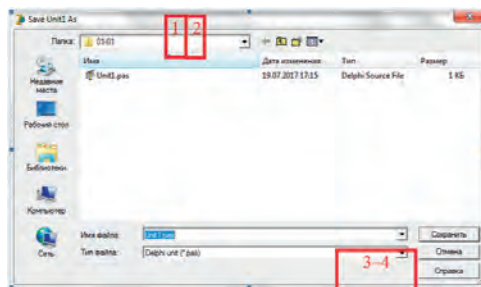
1. Как создается новое приложение в **Delphi**?
2. Какие методы имеются для сохранения приложения в **Delphi**?



38 УРОК. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

1. Запустите **Delphi**.

1.1. Создайте новое приложение и с помощью свойства **Caption** измените его заголовок на **01_02**. Сохраните приложение с помощью выбора пункта **Save** раздела **File** главного меню. Так как приложение сохраняется в первый раз, на экране появляется диалоговое окно.



1.1.1. В диалоговом окне в качестве текущей папки указана папка **01_01** предыдущего приложения. Нажимаем на кнопку 1, указанную на рисунке и выходим из него.

1.1.2. Для создания новой папки для приложения нажимаем на кнопку 2, указанную на рисунке.

1.1.3. В текущей папке появляется новая папка под именем **Новая папка**, которая выделяется. Дадим ей имя **01_02**, нажимаем на кнопку **Открыть**, появившуюся вместо кнопки **Сохранить**.

1.1.4. Для сохранения приложения с предложенным именем в новой папке, нажимаем на кнопку **Сохранить**.

1.2. Открыв приложение, проверяем его работу и закрываем окно приложения.

1.3. Выбрав пункт **Close all** в разделе **File** главного меню закрываем приложение.

2. Выполните на компьютере заданные на предыдущих занятиях упражнения.

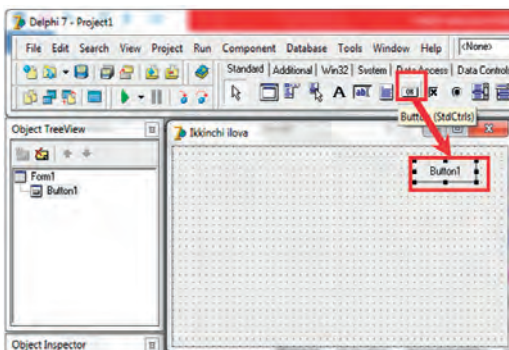


39 УРОК. КНОПКА УПРАВЛЕНИЯ

1. Вставка кнопки управления в окно приложения. Наше первое приложение, созданное на первом занятии, никакой функции не выполняет. Теперь мы устраним этот недостаток.

Создадим новое приложение с параметрами 225 на 400. Заголовок меняем на **Второе приложение**. В окне приложения установим объект управления: кнопку управления.

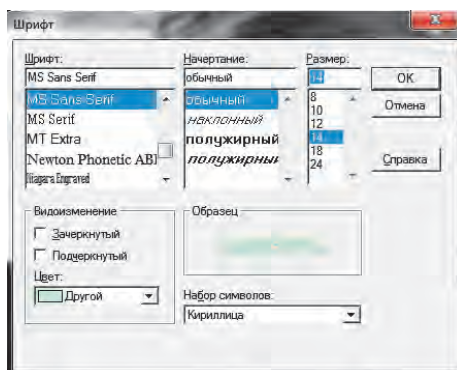
Выбираем мышью кнопку управления, расположенную в стандартном томе панели инструментов, затем наводим мышь на место кнопки управления окна приложения **Form1** и нажимаем еще раз. Если нужно изменить его место, можно сдвинуть его на новое место с помощью мыши.



2. Определение свойств кнопки управления. Как окно приложения имеет свои свойства, так и кнопка управления, расположенная на нём, имеет свои свойства. Их тоже можно менять. Это облегчается тем, что многие свойства объекта кнопка аналогичны свойствам объекта **Form**.

Сначала: выбираем кнопку **Button1** путём наведения на него мыши и нажатия его левой кнопки. В Диспетчере Объектов (**Object TreeView**) **Button1** выделяется и в окне Инспектора Объекта (**Object Inspector**) появляется список свойств объекта **Button1**. В свойстве **Caption** меняем заголовок кнопки на '++'. В соответствии с договорённостью, принятой на предыдущем занятии, это пишется следующим образом:

```
Button1.Caption:='++'
```



Так как название кнопки получилось маленьким, увеличим его размер. Для этого используем свойство **Font**. При его выборе на экране появляется диалоговое окно **Шрифт**. Выбрав из списка под его полем **Размер** число **14**, нажимаем кнопку **OK** (см. Рисунок выше):

3. Связывание событий с кнопкой управления. Кнопка **Button1** должна выполнять определённую задачу в приложении. Пусть при нажатии на неё окно меняет размеры своего окна на 16 и 9.

Сначала сделаем двойной щелчок на кнопке. На экране появляется окно кода программы. На месте курсора вводим строки:

```
Form1.Height:=Form1.Height+9;
```

```
Form1.Width:=Form1.Width+16;
```

Нажав кнопку **F9**, начинаем работу с приложением и проверяем правильность его работы, нажав несколько раз кнопку **Button1**.



ЗАПОМНИТЕ !

При нажатии кнопки управления **Button1** в приложении включается процедура **TForm1.Button1Click** и выполняются его операторы.



ВОПРОСЫ И ЗАДАНИЯ

1. Как переносится кнопка управления в окно приложения?
2. Где изменяются свойства кнопки управления?
3. Где отображаются функции кнопки при выборе управления ею?
4. Создайте новое приложение в **Delphi**. Введите в него четыре кнопки управления. Выведите на экран первую под именем **Создай**, вторую **Загрузи**, третью **Сохрани** и четвёртую **Заверши**.



ДОМАШНЕЕ ЗАДАНИЕ

1. Выпишите основные свойства кнопки управления.



40 УРОК.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

1. Создайте в **Delphi** новое приложение и сохраните его в папке **01_03**.

1.1. Измените заголовок приложения на **Здравствуй, мир!**

1.2. В окне приложения разместите в одну колонну три кнопки управления.

1.3. Нажмите первую кнопку дважды. В появившуюся на экране процедуру **TForm1.Button1Click** после **begin** введите **TForm1.Button1.Caption:='Salom, '+TForm1.Caption;**

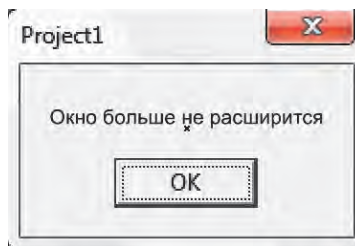
1.4. Дважды нажав на вторую кнопку, приведите процедуру в следующий вид:

```
procedure TForm1.Button2Click(Sender: TObject);  
begin
```

```
TForm1.Left:= TForm1.Left+10;
end;
```

1.5. Дважды нажав на третью кнопку, приведите процедуру в следующий вид:

```
procedure
TForm1.Button3Click(Sender:
TObject);
begin
TForm1.Top:= TForm1.Top+10; end;
```



1.6. Начав работать с приложением, наблюдайте за тем, какие функции выполняют кнопки. Закрыв окно приложения, сохраните все его файлы. Закройте все файлы приложения.

2. Выполните на компьютере упражнение, заданное на предыдущем занятии.



41 УРОК. ОКНО SHOWMESSAGE

1. Недостаток второго приложения. Начав работать со вторым приложением, нажимаем на кнопку “++”. Ширина экрана приложения превысит 16000 пикселей. Это непростительная ошибка для программиста. Для устранения ошибки договоримся не превышать ширину экрана приложения за 600 пикселей. Но если при нажатии пользователем кнопки ничего не изменится, может возникнуть сомнение в правильности работы программы.

2. Процедура ShowMessage. Чтобы предупредить это приложение должно выводить на экране информацию о том, что его ширина превысила 600.

Мы это осуществим посредством **ShowMessage**. Дважды нажав на кнопку **Button1** второго приложения, вместо двух операторов введем следующие операторы:

```
If Form1.Width<600 then begin
Form1.Width:=Form1.Width+16;
Form1.Height:=Form1.Height+9; End else
ShowMessage('Окно больше не расширится');
```

Запустив приложение и нажав несколько раз на кнопку, получим на экране, показанную на рисунке. Пока мы не нажмём кнопку **ОК**, приложение приостановит свою работу.

3. Функция IntToStr. Улучшим приложение: для нас пока чтобы оно выводило на экран текущий размер окна при изменении размеров. Можно ли размер экрана вывести на окно приложения? Пока мы знаем только один способ: вывести размеры окна на заголовок окна. Для этого используем оператор

```
Form1.Caption:=Form1.Height+' *'+Form1.Width;
```

Но данный оператор неверный, в нем предпринята попытка ввести в один тип переменное значение другого типа. В **Delphi** это приведёт к ошибке.

Для перевода величины целого типа в величину типа строки текста используем функцию **IntToStr** (**integer to string** – от целого к строке):

```
Form1.Caption:=' ('+IntToStr(Form1.Height)+' *'+IntToStr(Form1.Width)+' )';
```

Добавим в него истинное имя приложения:

```
Form1.Caption:=' третье приложение'+Form1.Caption;
```

Нажав дважды на кнопку **Button1**, выводим окно кода программы на экран и добавим в него последние две строки:

```
If Form1.Width<600 then begin  
Form1.Width:=Form1.Width+16;  
Form1.Height:=Form1.Height+9; End else  
ShowMessage('Окно больше не расширится');  
Form1.Caption:=' ('+IntToStr(Form1.Height)+' *'+IntToStr(Form1.Width)+' )';  
Form1.Caption:=' Третье приложение'+ Form1.Caption;
```

Открыв приложение, наблюдаем изменение его размера и заголовка:





ЗАПОМНИТЕ !

ShowMessage создаёт новое диалоговое окно и пока оно не закроется, работа приложения приостанавливается.



ВОПРОСЫ И ЗАДАНИЯ

1. В чём заключается недостаток второго приложения?
2. Какую функцию выполняет процедура **ShowMessage**?
3. Какую функцию выполняет функция **IntToStr**?
4. Измените третье приложение таким образом, чтобы оно выводило на экран предупредительную надпись при превышении а) ширины окна приложения более чем на 800, б) высоты окна приложения более чем на 450 пикселей.



ДОМАШНЕЕ ЗАДАНИЕ

1. Вспомните основные свойства окна приложения и перепишите их в свою тетрадь.



42 УРОК.

КОНТРОЛЬНАЯ РАБОТА

Ниже приведены образцы вариантов контрольной работы:

1 вариант

1. Программирование на машинном коде и ассемблере.
2. Главное меню **Delphi**.
3. Свойство **Caption** объектов.

2 вариант

1. Высокоуровневые языки программирования.
2. Интерфейс **Delphi**.
3. Свойства **Left** и **Top** объектов.

3 вариант

1. Визуальное программирование
2. Окна **Delphi**.
3. Свойства **Width** и **Height** объектов.

4 вариант

1. Обобщённое программирование.
2. Создание приложения в **Delphi** и его сохранение.
3. Процедура **ShowMessage**.



43 УРОК.

РАЗМЕЩЕНИЕ ДАННЫХ В ОКНЕ ПРИЛОЖЕНИЯ

1. Объект управления Label и его свойства. Любое приложение должно выводить некоторую информацию для пользователя в окно приложения. Для этого есть ряд объектов управления и самым простым из них является объект **Label** (знак, надпись). Данный объект расположен в пятой вкладке **Стандарт панели** объектов.

Этот объект является наследником объекта окна приложения **Form** и многие его свойства позаимствованы им. На примере, такие свойства **Label** как **Width** (ширина), **Height** (высота), **Left** (левый), **Top** (выше), соответственно, определяют ширину, высоту, расстояние объекта от левой и верхней границ.

Основным свойством **Label** является **Caption** (Заголовок). Им определяется строка текста, выводимая объектом на экран.

2. Изменение свойств объекта управления Label. У объекта **Label**, также как и у других объектов, есть свое имя и оно хранится в свойстве **Name**. Это свойство используется при обращении к объекту и его свойствам. Поэтому его изменение при работающем приложении приводит к проблеме. Свойство **Name** меняется только один раз при вставке объектов в окно приложения.

При размещении объектов в приложении (это ещё называются созданием копии объекта) им автоматически присваиваются имена. К примеру, копиям объектов **Label** последовательно присваиваются имена **Label1**, **Label2**, **Label3**. Вспомните объекты под именами **Form1**, **Button1** из предыдущих приложений.

Значение свойства объекта **Name** используется при создании двойных имён для обращения к другим свойствам: имя объекта, имя свойства. К примеру, **Form1.Width**, **Button1.Caption**, **Label1.Left**. Чтобы избежать проблем в этой области мы, будем некоторое время использовать имена автоматически присвоенные программой.

3. Приложение параметров окна. Заново создадим приложение с предыдущего занятия. Для увеличения и уменьшения размеров окна

приложения требуется два **Button** для отображения размеров окна используем два **Label** сначала установим свойства окна приложения:

```
Form1.Caption:='Четвертое приложение;  
Form1.Height:=225; Form1.Width:=400;  
Form1.Left:=200; Form1.Top:=100;
```

Все четыре объекта управления вставим в окно приложения и установим их свойства:

```
Button1.Caption:='++';  
Button2.Caption:='--';  
Label1.Caption:='Высота окна: ' ;  
Label2.Caption:='Ширина окна: ' ;
```

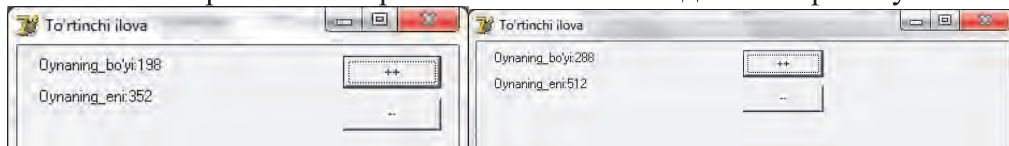
В окне приложения дважды нажимаем на **Button1**, на месте курсора в окне кода программы на экране вводим следующие операторы:

```
Form1.Width:=Form1.Width+16;  
Form1.Height:=Form1.Height+9;  
Label1.Caption:='Высота окна:'+IntToStr(Form1.Height);  
Label2.Caption:='Ширина окна:'+IntToStr(Form1.Width);
```

В окне приложения дважды нажимаем на **Button1**, на месте курсора в окне кода программы на экране вводим следующие операторы:

```
Form1.Width:=Form1.Width+16;  
Form1.Height:=Form1.Height+9;  
Label1.Caption:='Высота окна:'+IntToStr(Form1.Height);  
Label2.Caption:='Ширина окна:'+IntToStr(Form1.Width);
```

Начинаем работать с приложением и наблюдаем его работу:



ЗАПОМНИТЕ !

Объект **Label** используется для вывода строки текста в окно приложения и данный текст сохраняется в свойстве **Caption**.



ВОПРОСЫ И ЗАДАНИЯ

1. Как в приложении можно изменить свойства объекта управления **Label**?

3. Измените четвёртое приложение таким образом, чтобы высота и ширина его окна изменились с шагом в 5 и 8.

3. Измените четвёртое приложение таким образом, чтобы значение его ширины изменилось от 500 до 800.

4. Надписи на кнопках четвёртого приложения измените на **a→A** и **A→a**.



ДОМАШНЕЕ ЗАДАНИЕ

1. Объясните предназначение объекта управления **Label**.

2. Перечислите основные свойства объекта управления **Label**.



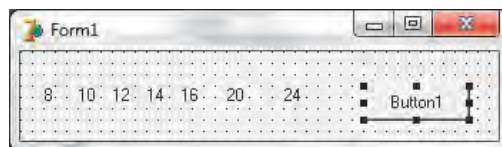
44 УРОК. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

1. Создайте новое приложение и сохраните его в папке **03_01**.

1.1. Вставьте в приложение семь объектов **Label** и один объект **Button**.

1.2. Измените надписи на **Label** с помощью свойства на 8, 10, 12, 14, 16, 20 и 24.

1.3. Соответственно дважды нажмите на кнопку **Button1**, на месте курсора в окне кода



программы на экране введите следующие операторы:

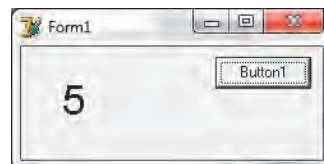
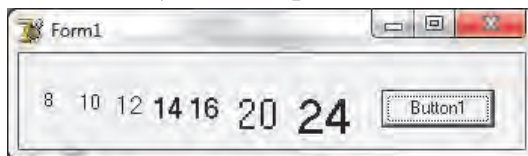
```
Label1.Font.Size:=8; Label2.Font.Size:=10;  
Label3.Font.Size:=12; Label4.Font.Size:=14;  
Label5.Font.Size:=16; Label6.Font.Size:=20;  
Label7.Font.Size:=24;
```

1.4. Откройте приложение и нажав кнопку, посмотрите на результат:

2. Создайте новое приложение и сохраните его в папке **03_02**.

Вставьте в него по одному объекту **Label** и **Button**. При нажатии кнопки **Button** каждый раз размер шрифта надписи на **Label** должен увеличиться на две единицы:

3. Выполните на компьютере упражнения, заданные на предыдущем уроке.





45 УРОК. ДОСТУПНОСТЬ И ВИДИМОСТЬ ОБЪЕКТОВ УПРАВЛЕНИЯ

1. **Свойство Visible объектов управления.** Почему сложно находить компьютерные вирусы? Потому что они скрыты, у них нет своего окна. Можно ли сделать скрытыми создаваемые приложения? Конечно же, можно.

Для этого нужно будет использовать свойство **Visible** (видимый), которое есть у каждого управляющего объекта, в том числе у окна приложения. Это свойство является логическим и принимает только значения **True** (Истина) или **False** (Ложь).

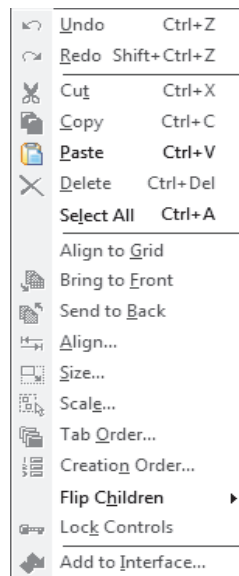
Свойство **Visible** объекта первично бывает Истиной и объект виден в окне. Если это свойство установить как Ложь, объект перестанет быть видимым в окне. Его место можно использовать в других целях.

Если два или более объектов управления не используются одновременно, можно сделать истиной свойство **Visible** только одного из них, остальные установить как ложь, тем самым можно сэкономить место на экране. В визуальном программировании это практикуется часто.

2. **Свойство Enabled объектов управления.** На рисунке изображён раздел **Edit** (Редактирование) главного меню **Delphi 7**. Можно увидеть, что многие его пункты тусклые. Это означает, что они не активны и их нельзя использовать.

К примеру, для использования четвёртого пункта меню **Copy** (копирование) прежде нужно выделить какую-то часть текста. С выделением данной части текста появится возможность её копирования и активируется пункт меню **Copy**, он станет темным и можно будет выполнить данное действие.

Для этой цели используется Свойство объектов управления **Enabled** (доступный). Это свойство также принимает логические значения. Если его значение Истина, то объект доступен, если Ложь –



объект недоступен. Первично данное свойство является истиной для всех объектов и они доступны. Если данное свойство установить как Ложь, хотя объект и виден в окне, он станет тусклым. Он временно станет недоступным для использования.

3. Использование свойств Visible и Enabled. Используя данные свойства объекта, улучшим работу последнего приложения. Если ширина его окна превысит 600, установим свойство **Enabled** кнопки **Button1** равной **false**. В результате дальнейшее расширение окна станет невозможным. Пусть сначала вторая кнопка **Button2** в окне приложения будет не видна (**Button2.Visible=False**).

Нажатием первой кнопки расширяем окно приложения. Как только его ширины достигнет значение 600, в окне появится вторая кнопка (**Button2.Visible=True**), и первая кнопка станет тусклой (**Button1.Enabled=False**). Теперь можно только уменьшить ширину окна приложения с помощью второй кнопки. Как только ширина окна станет меньше значения 600, первая кнопка вновь станет доступной (**Button1.Enabled=True**). С уменьшением ширины окна меньше 400, вторая кнопка вновь исчезнет, и окно больше не уменьшится.

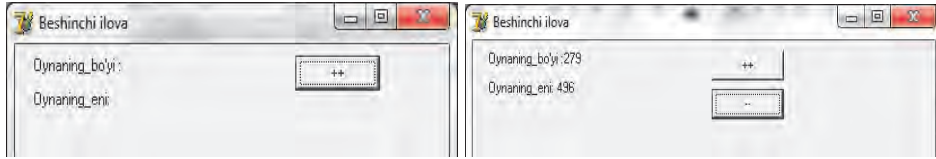
Чтобы показать программе, что нужно делать при нажатии первой кнопки, нажимаем ее дважды и на место курсора вводим следующий код:

```
If Form1.Width<600 then
begin
Form1.Width:=Form1.Width+16;
Form1.Height:=Form1.Height+9; End else
begin
Button2.Visible:=True;
Button1.Enabled:=False; end;
```

Нажав дважды вторую кнопку, введём следующую часть программы:

```
If Form1.Width>400 then
begin
Form1.Width:=Form1.Width-16;
Form1.Height:=Form1.Height-9; End else
```

```
Button2.Visible:=False;  
if Form1.Width<600 then Button1.Enabled:=True;
```



ЗАПОМНИТЕ !

Каждый визуальный объект может быть доступным или недоступным, скрытым или визуальным.



ВОПРОСЫ И ЗАДАНИЯ

1. Какую функцию выполняет свойство **Visible**?
2. В каких объектах встречается свойство **Enabled**?
3. Измените пятое приложение таким образом, чтобы шаг изменения его высоты и ширины было равным на 5 и 8, соответственно.
4. Измените пятое приложение, чтобы значение его ширины составлялось между 500 и 800.
5. Надписи на кнопках пятого приложения измените на **a->A** и **A->a**.



ДОМАШНЕЕ ЗАДАНИЕ

1. В каких объектах встречается свойство **Visible**?
2. Какую функцию выполняет свойство **Enabled**?



46 УРОК.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

1. Создайте новое приложение в **Delphi** и сохраните его в папке **04_01**.

1.1. Вставьте три объекта **Label** и три объекта **Button** в окно приложения.

1.2. Не изменяйте свойства **Enabled** (доступный) и **Visible** (видимый) у первых из них, а у третьих свойство **Visible** (видимый) установите как **False** (Ложь).

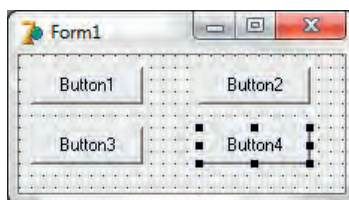


1.3. Откройте приложение и наблюдайте за изменениями в нём.

1.4. Сохраните приложение и закройте его.

2. Создайте новое приложение и сохраните его в папке **04_02**.

2.1. Вставьте в него четыре объекта **Button** в два столбца.



2.2. Пусть первый из них изменит доступность второго столбца, второй – доступность первого, а третий и четвертый – каждый раз изменит видимость на обратное.

2.3. Дважды нажав первую кнопку, введите в его процедуру операторы:

```
Button2.Enabled:=not Button2.Enabled;  
Button4.Enabled:=not Button4.Enabled;
```

а в процедуру второй кнопки:

```
Button1.Enabled:=not Button1.Enabled;  
Button3.Enabled:=not Button3.Enabled;
```

в процедуру третьей кнопки:

```
Button2.Visible:=not Button2.Visible;  
Button4.Visible:=not Button4.Visible;
```

и процедуру четвертой кнопки:

```
Button2.Visible:=not Button2.Visible;  
Button4.Visible:=not Button4.Visible;
```

2.4. Откройте приложение и наблюдайте за изменениями в нём.

2.5. Сохраните приложение и закройте его.

3. Выполните на компьютере упражнения, заданные на предыдущем уроке.



47 УРОК.

ВВОД ДАННЫХ В ПРИЛОЖЕНИЕ

1. Ввод в приложение строки текста. Объект управления Edit и его свойства. Любое приложение должно иметь возможность наряду с выводом и ввод в него данные. В среде **Delphi** самым простым способом является ввод объекта управления **Edit** (Поле ввода) в окно приложения. Данный объект расположен шестым в, **Стандартный панели** объектов управления. Его ярлычком является

Многие свойства данного объекта аналогичны со свойствам объекта **Form** окна приложения. В нём также имеются свойства **Left**, **Top**, **Width**, **Height**. Свойства **Caption** в нём нет. Поэтому он обычно используется вместе с объектом **Label**. Главным его свойством является **Text** (Текст), введённая в него строка текста хранится в данном свойстве. Если данному свойству присвоить значение определённой строки текста, данная строка отразится на поле ввода.

Ещё одним важным свойством поля ввода является **ReadOnly** (только для чтения). Значение данного свойства является логической величиной, принимает только значения **True** (Истина) или **False** (Ложь). По умолчанию значение данного свойства установлено как **False** (Ложь).

Если значение данного свойства установить как **True** (Истина), пользователь не сможет ввести в него данные. Но приложение может изменять в нём данные. Возникает естественный вопрос – для чего это нужно? В отличие от **Label** объект **Edit** вставлен в рамку и введённые в него данные смотрятся лучше. Если мы хотим использовать его для вывода данных, то необходимо, чтобы эти данные не были изменены пользователем.

2. Другие свойства объекта управления Edit. Ещё одним свойством объекта **Edit** является **MaxLength**. С его помощью можно ограничить длину вводимого в строку текста. Например, если в поле необходимо ввести год рождения ученика, нужно, чтобы **MaxLength** равнялся 4, что предохранит от ошибки ввода других данных.

Ещё одним важным свойством объекта **Edit** является **Font**, с помощью которого можно установить шрифт, размер шрифта, цвет и ещё ряд других параметров шрифта текста, вводимого в поле. Это свойство есть и у других объектов, например у объекта **Form**. Если мы установим шрифт для окна приложения, он окажет воздействие на все объекты окна приложения. Для предотвращения этого необходимо значение свойства объекта **ParentFont** (шрифт родителей) окна приложения перевести с Истины на Ложь.

Ещё одним полезным логическим свойством **Edit** является **AutoSize** (автоматический выбор размера), который обеспечивает увеличение

размера поля по мере увеличения вводимого текста. Если при вводе нужно, чтобы объекты рядом с полем не перекрывались, необходимо значение данного свойства изменить с Истины на Ложь.

Если поле ввода предназначено для ввода пароля пользователя для того, чтобы другие не видели вводимый, нужно заменить пароль, он заменяется другим знаком, например “*”. В **Edit** эта возможность тоже существует. Его свойство **PasswordChar** означает, что номер заменяемого знака в поле ввода (32–127). Первоначально оно равно #0 и означает незаменяемость вводимые буквы не заменяются. Если мы заменим его на #42, вместо введённых в поле букв появляется буква “х”. Вместо #42 также можно ввести непосредственно нужный знак. Например, “-“, “+“, “*“, “.”.

3. Приложение приветствия пользователя. Пусть пользователь в приложении введёт своё имя и общий для всех пароль: “12345”. Для этого создадим в **Delphi** новое приложение и вставим в его окно два объекта **Label**, три **Edit**, один **Button**. Изменим их свойства в Инспекторе объекта (**Object Inspector**) в следующем порядке:

```
Form1.Caption:='Шестое приложение';
Label1.Caption:='';
Label2.Caption:='';
Button1.Caption:='OK';
Edit1.Text:='';Edit2.Text:='';
Edit3.Text:='';
Edit2.Passwordchar:=#42;
Edit3.ReadOnly:=True;
```

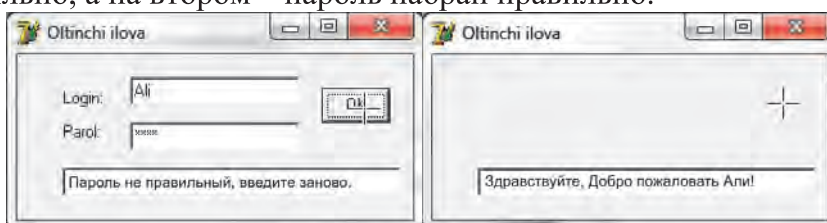
Выполнив всё это, дважды нажимаем на **Button1** и в окне текста программы, появившемся на экране, на месте курсора, введём следующие операторы без ошибки:

```
if Edit2.Text='12345' then begin
Edit3.Text:='Здравствуйте,'+Edit1.Text+'!Добро
пожаловать!';
Edit1.Visible:=False;Edit2.Visible:=False;
Label1.Visible:=False;Label2.Visible:=False;
Button1.Visible:=False; end else
```

```
Edit3.Text:='Пароль неправильный, введите заново.';
```

Данные операторы сработают при нажатии на кнопку **Button1**. Первый из них проверяет правильность введённого в поле **Edit2** пароля. Если пароль правильный, оператор со второй строки выводит на третье поле строку приветствия для пользователя. Операторы следующих трёх строк скрывают с экрана ставшие уже ненужными два объекта **Label1**, **Label2**, два **Edit1**, **Edit2** и **Button1**. В противном случае, на третье поле выводится информация о неправильности введённого пароля и можно будет сделать следующую попытку.

На первом рисунке изображен вариант, когда пароль набран неправильно, а на втором – пароль набран правильно.



ЗАПОМНИТЕ!

Объект **Edit** используется для ввода в приложение данных в виде строки текста.



ВОПРОСЫ И ЗАДАНИЯ

1. Какую функцию выполняет объект управления **Edit**?
2. Что хранится в свойстве **Text** объекта **Edit**?
3. Какую функцию выполняет свойство **ReadOnly** объекта **Edit**?
4. Измените пароль в шестом приложении на а) “I7:7200U”; б) “123-12-20”.
5. В шестом приложении вместо команды **Edit3** используйте **ShowMessage**.



ДОМАШНЕЕ ЗАДАНИЕ

1. Какую функцию выполняет свойство **MaxLength** **Edit**?
2. Какую функцию выполняет свойство **PasswordChar** объекта **Edit**?



48 УРОК. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

1. Пусть в шестом приложении неправильно набран логин или пароль. Надпись в **Edit3** оповещает об этом. Если пользователь совершит ошибку ещё раз, надпись в **Edit3** не изменится. Если в приложении ничего не изменится, пользователь останется в недоумении. Для предотвращения этого, с началом нового ввода данных в **Edit1** или **Edit2**, надпись в **Edit3** должна исчезнуть.

1.1. Откройте **Delphi** и с помощью пункта **Open Project** раздела **File** основного меню загрузите шестое приложение.

1.2. Дважды нажмите мышью на **Edit1**. Введите оператор **Edit3.Text:=**"; на процедуре **TForm1.Edit1Change**, появившейся на экране.

1.3. Дважды нажмите мышью на **Edit2**. Введите оператор **Edit3.Text:=**"; на процедуре **TForm1.Edit2Change** появившейся на экране.

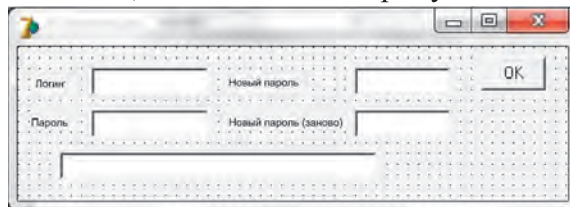
1.4. Откройте приложение и убедитесь, что изменение имеет место.

1.5. Сохраните приложение и закройте его.

2. Создайте приложение, меняющее старый пароль пользователя на новый.

2.1. Вновь загрузите шестое приложение и сохраните его в папке **05_02**.

2.2. В окно приложения добавьте ещё два объекта **Label** и **Edit**. Измените их свойства, как показано на рисунке:



2.3. Измените процедуру кнопки **OK**.

2.4. Запустите приложение и проверьте правильность его работы.

2.5. Сохраните приложение и закройте его.

3. Выполните на компьютере заданное на предыдущем уроке упражнение.



49 УРОК. ИЗМЕНЕНИЕ ТИПА ДАННЫХ

1. Изменение данных из строки текста на целые числа и наоборот. Если помните, в языке программирования **Pascal** одному типу переменного нельзя присвоить значение другого типа. В **Delphi** также действует этот запрет. Как его можно обойти?

Например, если ширина объекта **Button1** (свойство **Width**) зависит от свойства **Text** объекта **Edit** (тип строки текста), использование оператора **Button1.Width:=Edit1.Text** приведёт к ошибке. В результате приложение остановится и предупредит об ошибке.

В **Object Pascal** существует ряд стандартных функций для перевода переменных из одного типа в другой. Приведём их далее:

1) **StrToInt (string to integer)** – переводит строку текста в целое число;

2) **IntToStr (integer to string)** – целое число переводит в строку текста;

3) **StrToFloat (string to float)** – переводит строку текста в дробное число;

4) **FloatToStr (float to string)** – переводит дробное число в строку текста;.

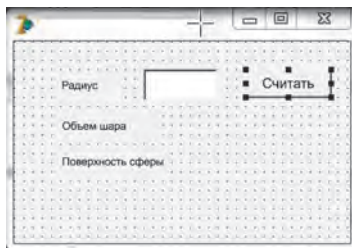
Вышеуказанный оператор вводим с использованием данных функций:
Button1.Width:=StrToInt(Edit1.Text);

2. Приложение вычисления объёма шара и площади поверхности сферы.

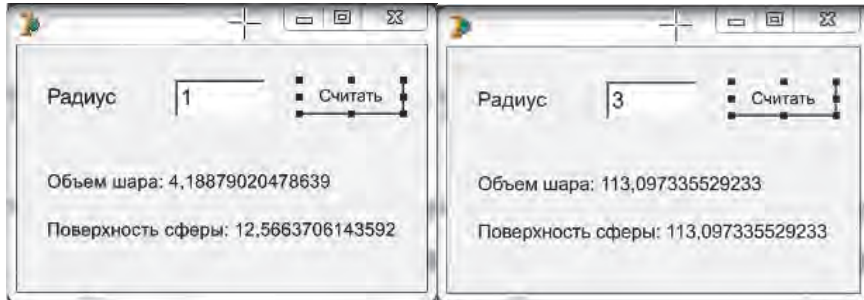
Создадим приложение для расчёта объёма шара и площади поверхности сферы. Для этого создадим новое приложение и вставим в него три объекта **Label**, один **Edit** и один **Button**. Изменим окно приложения и его объекты как показано на рисунке.

Дважды нажав на **Button1** в появившемся окне приложения, в окне кода программы введём две следующих строки. Каждая строка заканчивается точкой и запятой, их нужно ввести в отдельную строку.

```
Label2.Caption:='Объем  
шара:'+FloatToStr(4/3*pi*sqr(StrToFloat(Edit1.Text)  
) * StrToFloat(Edit1.Text));
```



```
Label3.Caption:='Поверхность
сферы:'+FloatToStr(4*pi*sqr(StrToFloat(Edit1.Text))
);
```



Дважды нажав клавишу **F9**, запускаем приложение. Сначала вводим 1 в поле ввода, нажимаем кнопку **Вычислить**, затем вычислим объём шара и сферы, радиусы которых равны 3. Результат работы приложения приведен на рисунках выше.

3. Использование переменных в приложениях. Последние две строки имеют довольно сложный вид и можно легко ошибиться при их вводе. Учитывая то, что объём шара и поверхность сферы вычисляются с помощью формул $V = \frac{4}{3}\pi r^3$; $S = 4\pi r^2$, вышеприведённые формулы действительно имеют громоздкий вид.

Можно ли устранить этот недостаток? Конечно, можно, для этого нужно использовать дополнительные переменные. Чтобы вспомнить способ ввода переменных в программу, поближе познакомимся с окном кода программы и текстом программы в нём.

4. Приложение суммы целых чисел. Даны целые числа m и n . Создадим приложение, вычисляющее сумму их и всех целых чисел между ними. Для этого создадим новое приложение и вставим в него три объекта **Label**, два **Edit** и один **Button**. Их расположение и свойства даны на рисунке внизу. При нажатии на кнопку **Button1** вычисление суммы чисел производится. Для ввода операторов в приложение нажмём данную кнопку дважды. На экране появится окно хода программы. Познакомимся с текстом:

```

Procedure TForm1.
Button1Click
Sender:TObject);
begin end;

```



Это простая процедура, написанная на языке Paskal, и пока она пуста. Значит, когда мы нажимаем на **Button1**, расположенный в окне приложения **Form1**, приложение переводит управление на процедуру **TForm1.Button1Click**. Вводим в данную процедуру нужный оператор. Процедура должна выглядеть так:

```

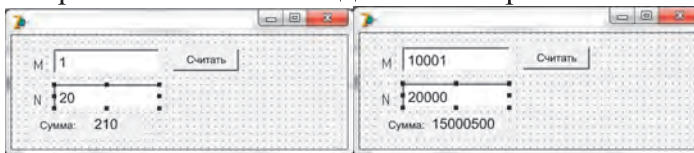
procedure TForm1.Button1Click(Sender: TObject);
var i,m,n,s:integer; begin
m:=Strtoint(Edit1.Text);
n:=Strtoint(Edit2.Text);
s:=0; for i:=m to n do s:=s+i;
Label3.Caption:='Сумма: '+Inttostr(s); end;

```

Во второй строке процедуры описаны 4 переменных типа целых чисел с помощью оператора **var**. С третьей строки начинается тело процедуры. В него размещены выполняемые операторы. В четвертой строке введённое в поле **Edit1** число превращается в целое и передаётся переменной *m*. В пятой строке введённое в **Edit2** число передаётся переменной *n*.

В шестой строке переменной, хранящей сумму, присваивается начальное значение: 0. Цикл в шестой строке находит нужную сумму. В седьмой строке значение найденной суммы через **Label3** выводится на окно приложения. Оператор **end** заканчивает процедуру.

Запустим приложение и наблюдаем за его работой:



ЗАПОМНИТЕ !

Если в операторе типы переменных и значений выражений различны, то в программе произойдёт ошибка.



ВОПРОСЫ И ЗАДАНИЯ

1. Какие типы данных вы знаете?
2. Из какого типа в какой переводит данные **StrToInt**?
3. Из какого типа в какой переводит данные **FloatToStr**?
4. Из какого типа в какой переводит данные **StrToFloat**?
5. Создайте приложение, вычисляющее периметр и площадь: а) квадрата; б) правильного четырёхугольника.
6. Создайте приложение, вычисляющее сумму: а) квадрат; б) квадратных корней всех чисел от m до n .



ДОМАШНЕЕ ЗАДАНИЕ

1. Выпишите в тетрадь функции, меняющие тип данных.



50 УРОК. КОНТРОЛЬНАЯ РАБОТА

Ниже приведены образцы вариантов для контрольной работы:

1 вариант

1. Основные свойства объекта **Label**.
2. Стандартная функция **FloatToStr**.
3. Создайте приложение, вычисляющее длину окружности.

2 вариант

1. Основные свойства объекта **Button**.
2. Стандартная функция **IntToStr**.
3. Создайте приложение, вычисляющее площадь круга.

3 вариант

1. Основные свойства объекта **Edit**.
2. Стандартная функция **StrToInt**.
3. Создайте приложение, вычисляющее площадь квадрата.

4-вариант

1. Процедура **OnClick** объекта **Button**.
2. Стандартная функция **StrToFloat**.
3. Создайте приложение, вычисляющее периметр квадрата.



51 УРОК. ПРИМЕНЕНИЕ ФЛАЖКОВ В ПРИЛОЖЕНИИ

1. Примеры пользования флажками.

Один из часто используемых объектов управления в приложении – это флажки.

Флажок может быть установлен или не установлен. Если флажок установлен, то какое-либо действие производится, если не установлен, то ничего не происходит.

Пример пользования флажками в одном из томов диалогового окна текстового редактора **Word** показан на рисунке выше.

Из имеющихся в нём шести флажков установлен только первый.

Можно ли пользоваться флажками в приложениях, созданных в **Delphi**? Конечно, можно. Это создаёт дополнительные удобства для пользователя, и ими широко пользуются в интерактивных приложениях.

2. Управляемый объект CheckBox и его свойства. В Delphi объект флажка называется **CheckBox** (Проверяемая ячейка).

Этот объект размещён девятым во вкладке стандарт панели объектов. На рисунке ниже он отмечен красной рамкой:



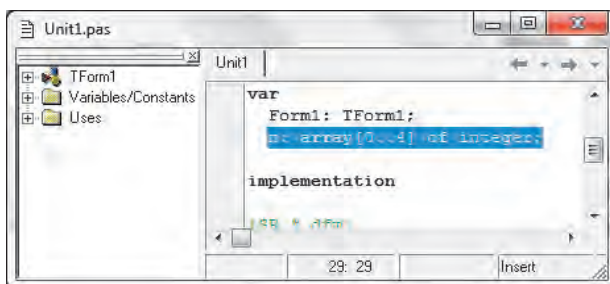
Основным свойством объекта **CheckBox** является **Checked** (установлен), оно принимает два значения: истина и ложь. Ещё одним его свойством является знакомый нам из предыдущих объектов **Caption**. Заголовок. С его помощью рядом с флажком можно расположить короткие записи о нём.

3. Создание глобальных переменных в приложении. Пусть в приложении имеется флажок и важно хранить количество установок флажка. Как можно сохранить эти данные?

Для сохранения этих данных можно делать следующее. Создав один лишний объект в приложении, оставляем его скрытым (**Visible:=False**). Первоначально его ширину делаем равной 0 (**Width:=0**), каждый раз, устанавливая флажок, увеличиваем его ширину на одну единицу (**Width:=Width+1**).

Однако это мера от безысходности. Этим способом, например, нельзя создать массив переменных. Оставим попытку заново создавать велосипед и вспомним последнее приложение: о сумме целых чисел.

Там с помощью оператора `var i,m,n, s:integer;` мы создали несколько переменных. Так как эти переменные созданы внутри процедуры, пользоваться ими за её пределами невозможно. Такие переменные называются **local** (локальными) переменными. В отличие от них **global** (глобальными) переменными можно пользоваться в любом месте программы. Глобальные переменные должны быть описаны в начале программы (приложения). Для нахождения этого места переходим в окно кода программы создаваемого приложения и находим оператора описания `var`.

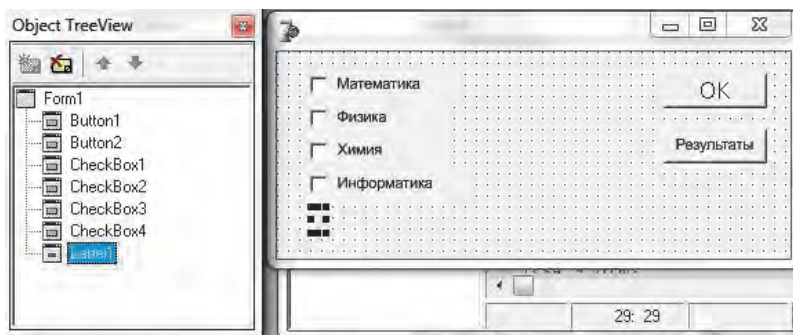


После этого введем нужный нам оператор описания `n: array[0..4] of integer`.

Теперь он находится рядом с переменной окна приложения `Form1` и на него, как и переменной `Form1`, можно обращаться ото всюду.

Описанием переменных дело не заканчивается. В начале работы приложения им нужно дать исходные значения: 0. Это можно сделать во время создания окна приложения.

4. Приложение интересных предметов. Чтобы научиться пользоваться флажками, разработаем приложение опроса интересных для учащихся предметов. Это приложение изучает интерес учащихся к тем или иным предметам на основе опроса. Для упрощения приложения ограничимся несколькими точными и естественными предметами. Создадим приложение в **Delphi**. Разместим в нём четыре объекта **CheckBox**, два объекта **Button** и один объект **Label**, как показано на рисунке. Заголовки окна приложений и объектов в них установим с помощью свойств **Caption**, как показано на рисунке:

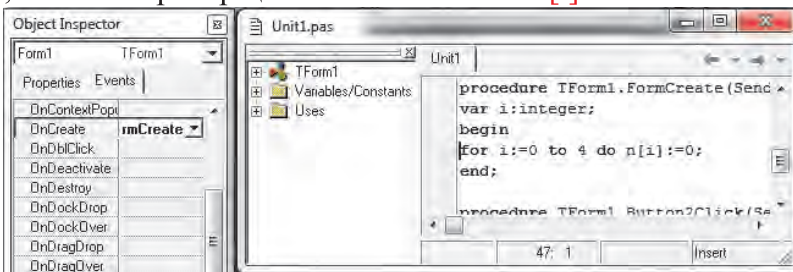


В код приложения введем оператор `n: array[0..4] of integer` в то место, которое было указано в предыдущем пункте (см.на предпоследний рисунок).

Для того, чтобы элементам этого массива дать первоначальные значения, выберем окно приложения **Form1** (нажимаем кнопку мыши на свободное от других объектов место окна приложения). Перейдя на вкладку **Events** (события) в **Object inspector**, находим в нём **OnCreate**. Справа от него в пустой клетке дважды нажимаем кнопку мыши.

В результате на экране появится окно приложения программы и в нём процедура **TForm1.FormCreate**.

В эту процедуру перед **begin** введем оператор описания `var i:integer`, затем оператор цикл `for i:=0 to 4 do n[i]:=0`.



Теперь в окно приложений **Button1** дважды нажимаем мышью, в место расположения курсора на экране введем следующий код программы.

```
n[0]:=n[0]+1;
if Checkbox1.Checked then n[1]:=n[1]+1; Checkbox1.Checked:=False;
if Checkbox2.Checked then n[2]:=n[2]+1; Checkbox2.Checked:=False;
if Checkbox3.Checked then n[3]:=n[3]+1; Checkbox3.Checked:=False;
```

```
if CheckBox4.Checked then n[4]:=n[4]+1; CheckBox4.Checked:=False;
```

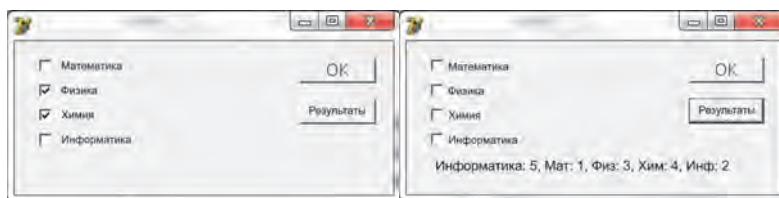
Каждый раз при нажатии кнопки **ОК** (при голосовании очередного учащегося) первый из этих операторов увеличивает число проголосовавших на один. В каждом из последующих операторов проверяется установка флажков и, если они установлены (соответственно голосовал за предмет), число голосов увеличивает значение переменной на один. Следующие за оператором **if** операторы после голосования сбрасывают флажки **CheckBox**, и приложение готово к следующему голосованию.

Для вывода результатов голосования на экран нажимаем кнопку **Результаты** два раза, введем в её процедуру следующие операторы:

```
Label1.Caption:='Qatnashchilar: '+Inttostr(n[0]);  
Label1.Caption:=Label1.Caption+',Mat: '+Inttostr(n[1]);  
Label1.Caption:=Label1.Caption+',Fiz: '+Inttostr(n[2]);  
Label1.Caption:=Label1.Caption+',Kim: '+Inttostr(n[3]);  
Label1.Caption:=Label1.Caption+',Inf: '+Inttostr(n[4]);
```

Первый из этих операторов выводит на экран число участников голосования, второй, третий, четвёртый, пятый при помощи свойств **Caption** объекта **Label1** выводят на экран число голосов, данных соответственно, по математике, физике, химии, информатике.

Запустим приложение и проверим правильность его работы. Ниже в рисунках изображены выбранные учащимся предметы и результаты опроса:



ЗАПОМНИТЕ!

Если флажок установлен, то выполняется соответствующее действие, в противном случае – нет.



ВОПРОСЫ И ЗАДАНИЯ

1. Какую функцию выполняют флажки?

2. При помощи какого объекта вводятся флажки в приложение?
3. При помощи какого свойства проверяется установка флажков?
4. Где описываются глобальные переменные?
5. Где можно присвоить исходные значения переменным?
6. Создайте приложение для проведения опроса Самые активных учащихся.
7. Создайте приложение для проведения опроса Самые лучшие учителя.



ДОМАШНЕЕ ЗАДАНИЕ

1. Чем отличаются друг от друга глобальные и локальные переменные?



52 УРОК. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

1. Создайте приложение для вычисления дискриминанта квадратного уравнения и его корней введённым коэффициентам уравнения.

2. Создайте приложение для вычисления среднего значения функции $y = \sin x$ в заданном интервале.

3. Создайте приложение по вычислению квадратного корня, корня третьей, четвёртой, шестой степени заданного числа, а зависимости от установленных флажков.

4. Выполните на компьютере упражнения предыдущего занятия.

5. Создайте окно приложения и поместите в него четыре флажка. запрограммируйте их таким образом, что при установке хотя бы одного флажка остальные должны сбрасываться.

5.1. Измените заголовки флажков соответственно на 1, 2, 3, 4.

5.2. Щёлкните два раза мышью на первый флажок, в его процедуру `TForm1.CheckBox1Click` введите операторы:

```
CheckBox2.Checked:=false;
```

```
CheckBox3.Checked:=false;
```

```
CheckBox4.Checked:=false;
```

5.3. В такую же процедуру второго флажка введите операторы смещения первого, третьего и четвёртого флажков.



5.4. Так же создайте подобные процедуры для третьего и четвертого флажков.

5.5. Начните работу в приложении и проверьте её правильность.



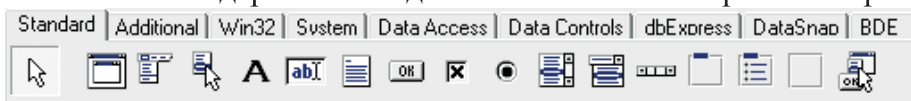
53 УРОК. ИСПОЛЬЗОВАНИЕ ГРУПП РАДИОКНОПОК В ПРИЛОЖЕНИИ

1. Примеры пользования радиокнопками. Часто приходится выбирать один вариант из нескольких. Например, при вступительных экзаменах в высшее учебное заведение. В каждом тестовом вопросе имеется четыре варианта ответов, из которых только один можно выбрать как правильный.

Из четырёх типов форматирования абзаца данных в **Word**: выравнивание слева, справа, посередине и с обеих сторон, только один можно применять к абзацу.

Выбор одного из них приводит к отмене всех остальных. Подобные примеры можно привести и из других сфер.

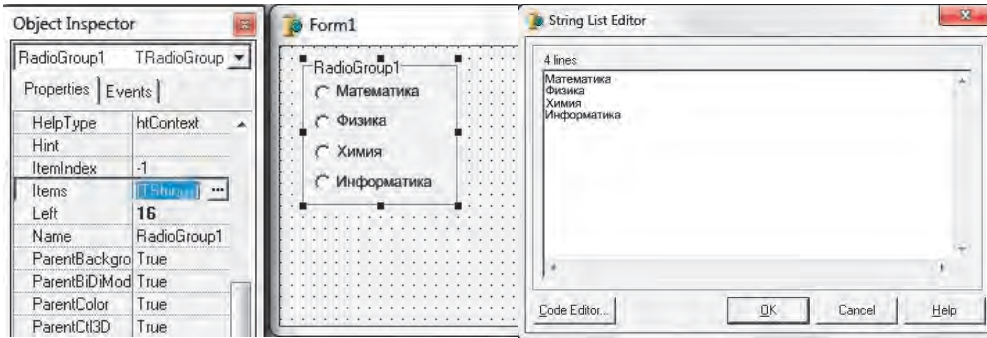
2. Объект управления RadioGroup и его свойства. Выбор одного варианта из нескольких выполнялся в **Pascal** при помощи оператора **Case**. В **Delphi** существует несколько таких способов, один из которых называется **RadioGroup** (группа радиокнопок). Этот объект управления расположен в стандартной вкладке панели объектов третьим справа:



Чтобы узнать, как работать с этим объектом, помещаем его в окно приложений. Одним из основных его свойств является не встречавшийся ранее **Items** (вариант). Выбираем двойным щелчком в **ObjectInspector** это свойство. При этом на экране появится его диалоговое окно. Туда можно ввести список названий вариантов кнопок радио. В него вводим четыре варианта названий, каждый с новой строки. Пусть это будет математика, физика, химия, информатика. Вид диалогового окна показан ниже.

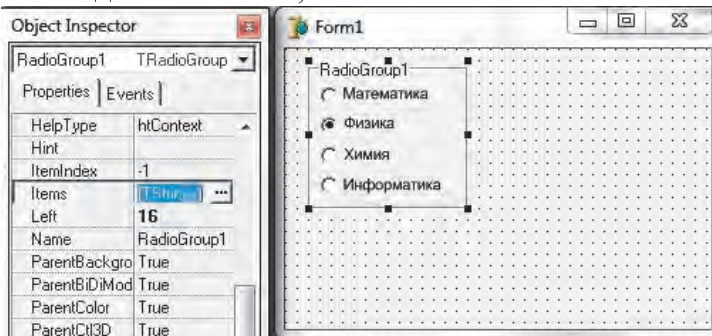
В диалоговом окне, нажимая кнопку **OK**, сохраняем введённые варианты или, нажимая кнопку **Cancel** (отменить), отказываемся от

введённых вариантов. Если принять введённые варианты, они тут же появятся в окне приложений:



Ещё одним важным свойством **RadioGroup** является **ItemIndex** (индекс варианта), в котором сохраняется номер выбранного варианта. Если ни один вариант не выбран, значение этого свойства будет равно 1. В инспекторе объектов значение этого свойства примем равным 1. Этот объект также обладает свойством **Caption**, в него в инспекторе объектов введем строку **Наука**. Эти изменения сразу же отразятся в окне приложения.

На первый взгляд, кажется странным, что выбрал науки второй в списке – физика, а не в списке – математика. Но здесь ничего необычного нет. Необходимо помнить, что счёт начинается в не с 1, а с 0:



Одно из свойств **RadioGroup** именуется как **Columns** (столбцы). Это свойство показывает количество столбцов для размещения вариантов группы кнопок радио.

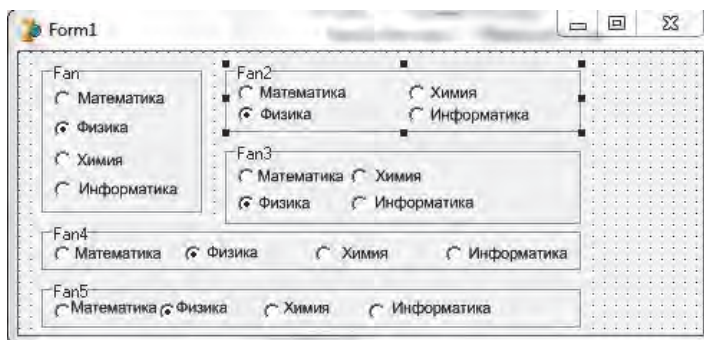
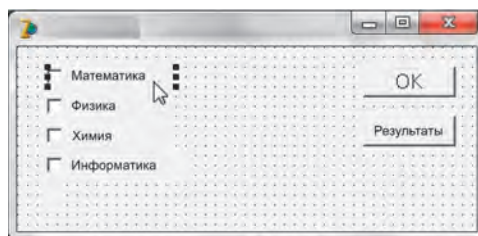
На рисунке ниже показано размещение в столбцах четырёх-вариантной группы кнопок радио, от одного до пяти. Из них вытекают следующие выводы:

1. Ширина выделенного места для группы кнопок радио равномерно распределяется между столбцами.

2. Определяются число кнопок в каждом столбце делением числа кнопок на число столбцов. Если количество кнопок в столбце будет дробным, оно округляется до большого целого числа.

3. Каждый столбец заполняется кнопками равномерно.

4. Один или несколько последних столбцов могут остаться пустыми. И для пустых столбцов выделяется место:



3. Приложение – Самый интересный предмет. Немного изменим приложение интересных предметов. Пусть каждый учащийся укажет при опросе только один предмет. В этом приложении воспользуемся объектом **Radiogroup**. Сначала удалим заголовок и четыре объекта **Checkbox**. Для этого, нажав поочередно левой кнопкой мыши над ними, нажимаем кнопку **Delete** на клавиатуре:

На его место описанным выше способом размещаем объект **Radiogroup** и вводим в него четыре радиокнопки, как показано выше.

Теперь, сделав двойной щелчок на кнопке **OK**, в его процедуру введем следующий код:

```

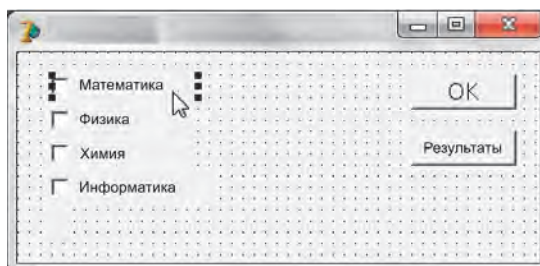
procedure TForm1.Button1Click(Sender: TObject);
var i:integer;begin
n[0]:=n[0]+1;
i:= Radiogroup1.Itemindex; if
i>=0 then n[i+1]:=n[i+1]+1;
Radiogroup1.Itemindex:=-1;
end;

```

Каждый раз, когда начинается работа этой процедуры (при нажатии кнопки **OK** в



приложении), помощи оператором из второй строки создаётся новая переменная **i**. В третьей строке число участников опроса увеличивается на один. В четвёртой строке выбранный в **Radiogroup1** вариант передаётся переменному **i**.



Если это значение больше 0 или равно ему, то есть, если какой-нибудь вариант выбран, то количество выбора этого варианта увеличивается на один.

В пятом ряду варианты переводятся в невыбранное состояние и процедура заканчивает свою работу. Приложение готово к приёму выбора следующего пользователя:



ЗАПОМНИТЕ !

Группа радиокнопок служит для выбора одного варианта из нескольких.



ВОПРОСЫ И ЗАДАНИЯ

1. С какой целью используется свойство объекта группы радиокнопок **Items**?
2. Какую функцию выполняет свойство **Columns** объекта группы радиокнопок?
3. В приложении, в котором размещены имена одноклассников, поместите группу радиокнопок. Выбрав одну из них, создайте приложение устного

приветствия по имени выбранного учащегося, когда нажимается кнопка управления.

4. В предыдущее приложение, в котором записаны названия наук и список оценок, введите ещё две группы радиокнопок. Создайте приложение вывода на экран отобранной оценки выбранного учащегося по выбранной науке.



ДОМАШНЕЕ ЗАДАНИЕ

1. Приведите пример ситуаций, в которых необходима группа радиокнопок.
2. Где расположен объект группы радиокнопок **RadioGroup** в панели объектов?



54 УРОК. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

1. Создайте новое приложение и сохраните его в папке **07_01**.
 - 1.1. Измените заголовок окна на **Тестовой системе**.
 - 1.2. Создайте группу радиокнопок и измените её заголовок на **Вопросы тестов**. Установите в нём 36 кнопок, их названия пронумеруйте от 1 до 36, сделайте необходимые исправления.
 - 1.3. В окно приложения добавьте новую группу радиокнопок, с названием **Ответы**. Добавьте к ней четыре кнопки, с названиями **A, B, C, D**.
 - 1.4. Запустите приложение, начните работу с ним, попробуйте выбрать в нём тестовые вопросы и ответы.
 - 1.5. Сохраните приложение и закройте его.
2. Создайте новое приложение и сохраните его в папке **07_02**.
 - 2.1. Приложения окно назовите **Электронная покупка**.
 - 2.2. Создайте группу радио кнопок в окне приложения и дайте ей заголовок **Тип платежа**.
 - 2.3. В список радиокнопок введите следующие варианты платежа: **PayNet; Click; U Pay; Uz card; Наличными; Денежным переводом**.
 - 2.4. Начните работу в приложении, попробуйте изменить тип платежа.
 - 2.5. Сохраните приложение и закройте его.
3. Создайте новое приложение и сохраните его в папке **07_03**.
 - 3.1. Назовите окно приложения **Выпускники**.

3.2. Разместите в окне приложений объект группы радиокнопок и назовите его **Образовательное учреждение**.

3.3. Введите следующие пункты в список радиокнопок:

- **Общеобразовательная школа;**
- **Академический лицей;**
- **Профессиональный колледж.**

3.4. Начните работу в приложении и выберите в нём тип образовательного учреждения.

3.5. Сохраните приложение и закройте его.

4. Выполните на компьютере задания из предыдущего занятия.



55 УРОК.

Объекты ListBox и ComboBox

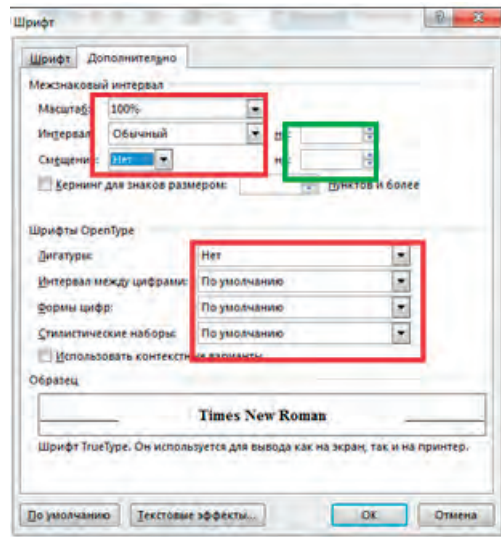
1. Объект управления ListBox и его свойства. У объекта **RadioGroup** есть один недостаток: если список вариантов длинный, то он занимает много места на экране. Есть объекты, лишённые этого



недостатка. Один из них – это объект управления **ListBox** (Ячейка списка), расположенный в стандартном вкладке панели объектов:

Объект **ListBox** выводит на экран список вариантов. Если этот список длинный и не помещается в отведённое для объекта место, то справа от него появляется полоса вертикальной прокрутки. Передвигая его, можно перейти в нужное место списка.

Многие свойства этого объекта одинаковы с соответствующими свойствами объекта **Radio Group**. С помощью свойств **Items**



вводятся элементы списка. В свойстве **ItemIndex** сохраняется порядковый номер вариантов выбранного элемента списка. И здесь порядковый номер начинается с 0.

Свойство Columns показывает количество столбцов вывода списка на экран. Если его значение равно 0, то список не подразделяется на столбцы. Свойством **Columns ListBox** следует пользоваться лишь при крайней необходимости. Потому что список, расположенный в нескольких столбцах, перестаёт быть похожим на список.

2. Объект управления ComboBox и его свойства. Объект **ComboBox** (скрытая ячейка списка), с значительно удобнее, чем **ListBox** и им чаще пользуются. Удобство его в том, что он, подобно полю ввода **Edit**, занимает мало места. Если нажать на треугольную кнопку справа от него, то можно вывести на экран скрытый список.



На верхнем рисунке скрытые списки диалогового окна шрифт текстового редактора **Word** изображены красным цветом. Основными свойствами этого объекта выступают **Items**, **Item Index** и **Column**. В **ComboBox** можно выбирать статью списка или вводить новую.

3. Объект управления SpinEdit и его свойства. В предыдущих приложениях для ввода численных величин мы пользовались объектом **Edit**. Пользуясь им, приходилось переводить введённые величины из строки текстового типа в тип целых чисел. Если при введении числа происходит ошибка, например, вместо 3 вводится три, в таком случае приложение прекращает свою работу и сообщает об ошибке.



Объект **SpinEdit** (поле введения чисел) предназначен для введения целых чисел. В нём есть возможность введения ранее выделенных в

промежутке целых чисел. Это очень удобно для приложений. В верхнем рисунке два из этих объектов выделены зелёной каёмкой.

Объект **SpinEdit** расположен в томе **Samples** (Образцы) панели объектов. Эта вкладка не видна на экране. Чтобы увидеть её, нужно несколько раз нажать на вторую кнопку треугольной формы в конце списка вкладок.

В **SpinEdit** можно непосредственно вводить число в поле ввода. Но можно увеличить или уменьшить находящееся в нём число при помощи двух кнопок в правой стороне поля ввода с заданными заранее шагом.

Введённое в **SpinEdit** числовое выражение сохраняется в его свойстве **Value** (значение). Наименьшее принимаемое числовое значение сохраняется в свойстве **MinValue**, наибольшее – в **MaxValue**. Если значения этих свойств будут равны 0, то для вводимых значений устанавливается граница как снизу, так и сверху. В свойстве **Increment** (шаг) указывается шаг увеличения числа. По умолчанию шаг равен 1.

4. Приложение лучший футболист. Чтобы научиться пользоваться рассмотренными объектами, создадим приложение – Лучший футболист. Создав новое приложение, разместим в его окно по одному объекту **Listbox**, **ComboBox**, **SpinEdit** и **Button**, как показано на рисунке. Нажимаем два раза свойство **Items** объекта **Listbox**, в появившемся диалоговом окне размещаем список из 8 футболистов, как показано на верхнем рисунке. Выбрав объект **SpinEdit**, в окне **Object Inspector** устанавливаем его свойство **MaxValue**, равным 5, свойство **MinValue**, равным 1, свойство **Value**, равным 5.

Нажав два раза на объект **Button1**, вызываем окно кода программы и в процедуру **Button1Click** введем следующий код:

```
procedure TForm1.Button1Click(Sender: TObject);
var i,j:integer;
begin
i:=ListBox1.ItemIndex; if i>=0 then begin
j:=StrToInt(ComboBox1.Items[i]);
ComboBox1.Items[i]:=IntToStr(j+SpinEdit1.Value);
end; end;
```

Не забудьте, что первая, третья и седьмая строка кода уже имеется в окне кода. Во второй строке создаются две новые переменные. В четвёртой строке первой из них: **i**, присваивается номер выбранного из списка футболиста. Здесь же проверяется неотрицательность номера (иначе футболист выбран).

Второй переменной **j** в шестой строке (присваивается в начале равно 0) значение **i** – строки списка **ComboBox1**. К этому числу в шестом ряду прибавляется число, имеющееся в **SpinEdit1** (балл, данный футболистом) и, преобразовав его в строку текста, записывается в **i** строку списка **ComboBox1**.

В результате набранные каждым футболистом баллы собираются в списке **ComboBox1**. При желании, нажав на треугольную кнопку в правой стороне этого объекта, мы можем их увидеть.

Нажав кнопку **F9**, начнём работу с приложением. Выбрав в нём футболиста из первого списка, во втором поле выбираем балл, добавляемый ему, нажимаем кнопку **OK** (рисунок слева, чтобы увидеть, результаты). Проголосовав несколько раз за футболистов, нажимаем треугольную кнопку в правой стороне третьего поля (рисунок справа), чтобы увидеть результаты.



ЗАПОМНИТЕ !

Объект **ComboBox** служит для создания скрытого списка.



ВОПРОСЫ И ЗАДАНИЯ

1. Какую функцию выполняет объект **Listbox**?
2. Назовите функции и основные свойства объекта **ComboBox**.
3. Назовите функции и основные свойства объекта **SpinEdit**
4. Разместите в приложении объект **ListBox**. Измените свойства этого объекта так, чтобы на нем можно было выбрать один из автомобилей, выпускаемых на заводе Асака.
5. Разместите в приложении объект **ComboBox**. Измените его свойства и введите в него список областей.



ДОМАШНЕЕ ЗАДАНИЕ

1. В каком месте панели объектов расположен объект **Listbox**?
2. Какую функцию выполняет объект **ComboBox**?



56 УРОК. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

1. Выполните с помощью объектов **ComboBox** упражнения с предыдущего урока **07_01**, **07_02** и **07_03**. Сохраните их соответственно в папках **08_01a**, **08_02a** и **08_03a**.

2. Выполните с помощью объектов **ListBox** упражнения с предыдущего урока **07_01**, **07_02** и **07_03**. Сохраните их соответственно в папках **08_01b**, **08_02b**, **08_03b**.

3. Создайте приложение, отображающее в объекте **ListBox** введённые в числа Edit.

4. Создайте приложение, отображающее в объекте **ComboBox** введённые в числа Edit.

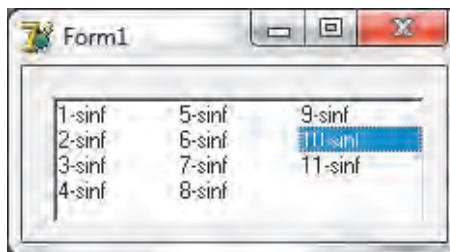
5. Создайте приложение, отображающее в объекте **ListBox** классы с первого по одиннадцатый.

5.1. Создайте приложение, и в его окно вставьте объект **ListBox**.

5.2. Дважды нажав на пустое место в окне приложения, заполните процедуру на экране следующим образом:

```
procedure TForm1.FormCreate(Sender: TObject);  
var  
i:integer;  
begin  
listbox1.Columns:=3;  
for i:=0 to 10 do  
listbox1.Items[i]:=intto  
str(i+1)+'-sinf';  
end;
```

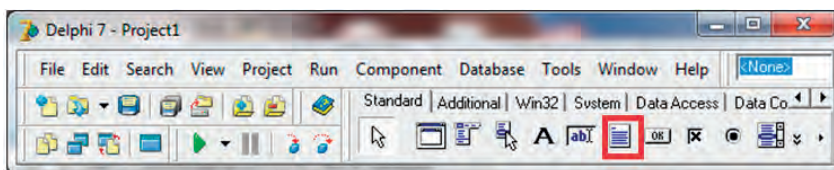
5.3. Запустите приложение и проверьте правильность его работы:



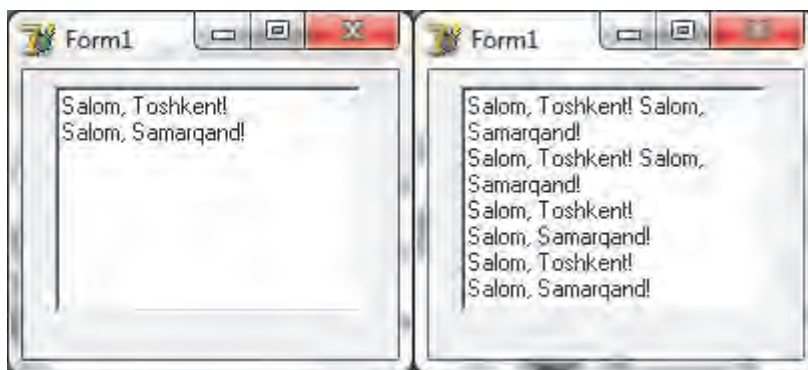
57 УРОК. ОБЪЕКТ УПРАВЛЕНИЯ Memo

1. Объект Memo и его возможности. В некоторых приложениях возникает необходимость работы с большим объёмом данных. В объектах **Delphi**, которые мы изучаем, такой возможности нет. Теперь мы познакомимся с объектом управления, предоставляющими такие

возможности. Это объект **Мемо**, сокращённый от **Memorandum** которое означает **служебные записи**. Этот объект расположен в стандартной вкладке панели объектов.



Создав в **Delphi** новое приложение, вставим в него данный объект. Открыв приложение, нажимаем мышью на **Мемо**. В его рабочем поле появится курсор. Значит, в него можно вводить текст:



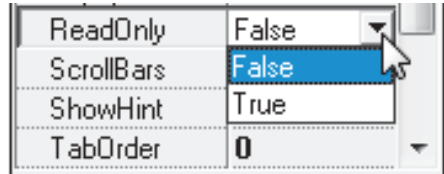
Если при вводе текста строка заполнится, то ввод текста продолжится со следующей строки. А если окно заполнится текстом, текст передвигается вверх и появляется новая строка. Можно продолжать ввод текста.

Передвигая мышью или с помощью кнопок управления курсором можно выделить часть текста, которую можно копировать, вырезать и вставлять в текст с помощью **Ctrl+X**, **Ctrl+C**, **Ctrl+V**. Можно сказать, что данный объект может выполнять функцию.

2. Основные свойства объекта Мемо. Сколько текста можно ввести в объект **Мемо**? Есть ли ограничение? Конечно, есть. В тексте, введённом в данный объект, максимально может быть $2^{16}=65536$ букв и других знаков. Это означает более 1000 строк текста.

Если мы хотим, чтобы пользователь не смог вводить или изменить данные в **Меню**, то должны будем перевести свойство **ReadOnly** (только для чтения) с **False** (Ложь) на **True** (Истина).

Введённый в **Memo** текст хранит в его свойстве **Text**. В нём может быть больше одной строки текста. Для отделения строк друг от друга, в конце и в начале строк необходимо будет вставлять знаки переноса на новую строку и перехода на начало строки. Это можно осуществить следующим образом:



```
Memо1.Text := '1-строка' + #13 + #10 + '2-строка' ;
```

Для того, чтобы научиться работать на нём, сначала создадим приложение для вычисления квадратов первых n количества натуральных чисел. Вставив в него по одному **Memo**, **Edit**, **Button**, изменим их свойства, как показано на рисунках:



Дважды нажав на кнопку **Button**, в процедуру на экране введём следующее:

```
procedure TForm1.Button1Click(Sender: TObject);
var i, i2, n: integer;
begin
n := StrToInt(Edit1.Text);
for i := 1 to n do begin i2 := i * i;
Memо1.Text := Memо1.Text + IntToStr(i) + 'квадрат' : ' + In
tToStr(i2) + #13 + #10;
end;
end;
```

Запускаем приложение и для начала находим квадраты первых 100, 1000 натуральных чисел.

Работа со свойством `Text` объекта `Memo` несколько неудобна, потому что его длина может быть до 65 тысяч. Намного удобнее использовать свойство `Lines` (строки). `Lines` – массив текстовых строк, на нужный элемент которого можно обратиться через его индекс. Например, следующий оператор вставляет 123 на начало первой строки текста `Memo1.Lines[0]:='123'+ Memo1.Lines[0]` (нумерация строк начинается с 0).

На самом деле `Lines` не свойство, а объект, который обладает собственными свойствами и методами. Например, `Memo1.Lines.Clear` удаляет все строки. Посредством свойства `Memo1.Lines.Count` можно узнать количество строк в тексте. `Memo1.Lines.Add('Последняя строка')` добавляет в конец текста новую строку и вводит в него предложение `Последняя строка`.

Новую строку можно ввести в любую часть текста. Для этого есть метод `Memo1.Lines.Insert`. К примеру, оператор `Memo1.Lines.Insert(4, 'Новая 5-строка')`; вводит перед пятой строкой еще одну строку новая `5-строка`. Следующие строки спускаются на строку ниже.

Еще двумя полезными способами `Lines` являются `SaveTo` и `LoadFrom`. С их помощью можно сохранить в формате текста введённый в объект текст и загрузить в объект сохранённый текст. Например, оператор `Memo1.Lines.SaveTo('d:\a1.txt')` введённый в `Memo` текст сохраняет в диске `d:` в файле `a1.txt`.

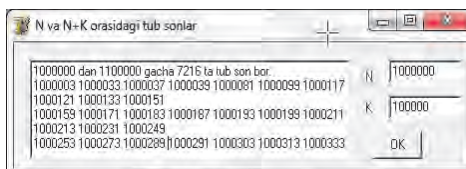
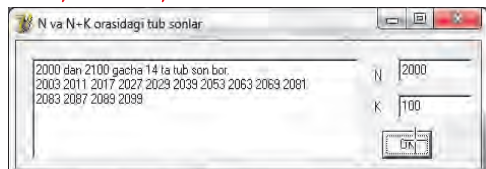
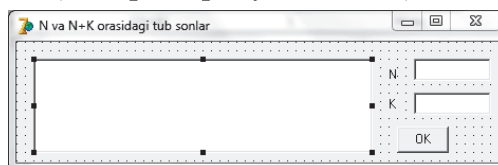
Оператор `Memo1.Lines.LoadFrom('d:\a1.txt')` загружает в объект сохранённый файл.

3. Приложение простых чисел. Является ли число 2017 простым? Если оно не простое, найдите ближайшее ему простое число. Напомним, что простым называют натуральное число, которое делится только на себя и единицу без остатка. Если число сложное, то хотя бы один из его делителей будет меньше его квадратного корня. В обратном случае произведение всех делителей будет больше данного числа. Значит, для того, чтобы определить, простое ли данное n число, достаточно проверить его неделимость на числа до \sqrt{n} .

Для того, чтобы решить эту задачу, создадим приложение для поиска всех простых чисел от n и всех последующих k натуральных чисел. Для ввода данных параметров разместим в окно создаваемого приложения два объекта **Label**, два **Edit**, один **Button** и один **Memo** и установим его свойства в нужном виде (смотрите рисунок выше).

Нажав дважды на кнопку **OK**, введём в процедуру следующий код:

```
procedure
 TForm1.Button1Click(Sender: TObject);
 var i, j, n, k, a: integer;
     b: boolean;
 begin
  n:=StrToInt(Edit1.Text);
  k:=StrToInt(Edit2.Text);
  for i:=n to n+k do begin
   b:=true; a:=round(sqrt(i)); j:=2;
   while b and (j<=a) do
    if i mod j = 0 then
     b:=false else j:=j+1;
   if b then Memo1.Lines.Add(IntToStr(i));
  end; end;
```



ЗАПОМНИТЕ !

В объекте **Memo** может быть размещено несколько тысяч строк текста.



ВОПРОСЫ И ЗАДАНИЯ

1. Где в панели объектов расположен объект **Memo**?

2. Строку текста, введённую в поле **Edit**, выведите в **Memo** – каждую букву на отдельной строке.

3. Введите в **Memo** текст из 50 строк. Уберите из текста строки с чётным числом.



ДОМАШНЕЕ ЗАДАНИЕ

1. Какую функцию выполняет свойство **Text** объекта **Memo**?

2. Как работает с текстом элемент **Lines** объекта **Memo**?



58 УРОК. КОНТРОЛЬНАЯ РАБОТА

Ниже приведены образцы вариантов для контрольной работы:

1 вариант

1. Объект **CheckBox** и его основные свойства.

2. Ввод строк текста в объект **Memo**.

3. Создайте приложение, выводящее на экран первые 10 нечётных чисел.

2 вариант

1. Объект **RadioGroup** и его основные свойства.

2. Свойство **ReadOnly** объекта **Memo**.

3. Создайте приложение, выводящее на экран первые 15 чётных чисел.

3-вариант

1. Объект **ListBox** и его основные свойства.

2. Свойство **Text** объекта **Memo**.

3. Создайте приложение, выводящее на экран первые 8 чисел, кратных 7.

4 вариант

1. Объект **ComboBox** и его основные свойства.

2. Свойство **Lines** объекта **Memo**.

3. Создайте приложение, выводящее на экран первые 12 чисел, кратных 5.



59 УРОК. РАБОТА С ГРАФИКОЙ В DELPHI

1. Графические возможности Delphi. В **Delphi** имеют широкие возможности для работы с графикой. У ряда объектов есть свойство

Canvas (полотно, поверхность), которое само считается объектом. Используя его методы и свойства, можно рисовать различные фигуры.

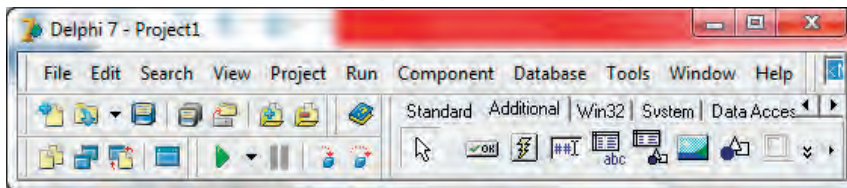
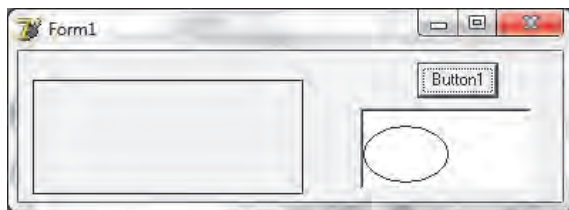
Например, для того, чтобы нарисовать прямоугольник в окне приложения, используем оператор `Form1.Canvas.Rectangle(10, 20, 200, 100);`. А `List box1.Canvas.Ellips(0, 10, 60, 50);` в **ListBox1** рисует эллипс, расположенный внутри прямоугольника, полуоси которого равны 30 и 20, а концы расположены в точках (0, 10) и (60, 50).

Для просмотра данных форм на экране создадим ещё одно приложение. Вставим в него объекты **Button** и **ListBox**. Дважды нажав на кнопку **Button1**, на место курсора введём два оператора:

```
Form1.Canvas.Rectangle(10, 20, 200, 100);  
ListBox1.Canvas.Ellips(0, 10, 60, 50);
```

Если открыть приложение и нажать на кнопку, то на экране появится прямоугольник и эллипс. Как видно из рисунка, начало отсчета находится в левом верхнем углу объекта. Указываются сначала координата горизонтальная, затем вертикальная:

2. Объекты Image и Shape. В **Delphi** для работы с графикой имеются специальные объекты управления. Это **Image** (рисунок, снимок) и **Shape** (форма), которые расположены в том **Additional** (дополнительной) панели объектов:



В объектах **Canvas** (полотно) создаётся графиком с помощью свойств, способов и объектов предшественников. Основными рабочими инструментами **Canvas** являются **Pen** (карандаш) и **Brush** (кисть). С помощью **Pen** рисуются линии. А **Brush** используется для покраски полей.

Основными свойствами **Pen** (карандаш) являются **Width** (ширина), **Style** (стиль) и **Color** (цвет). Оператор `Image1.Canvas.Pen.Width:=5;` устанавливает величину ширины линии, равную 5 пикселям.

Свойство **Style** изменяет вид рисуемых линий. Он может быть равным заранее известным значениям следующих постоянных:

1. **psSolid** – непрерывный (заданный).
2. **psDash** – штрих.
3. **psDot** – точки.
4. **psDashDot** – тире и точка.
5. **psDashDotDot** – тире, точка и точка.
6. **psClear** – скрытый.

Оператор, меняющий стиль карандаша, может быть вида

```
Image1.Canvas.Pen.Style:=psDash;
```

Свойство **Color** (цвет) объекта **Pen** (карандаш) устанавливает цвет рисуемой линии. И в этом свойстве тоже можно пользоваться заданными постоянными. К примеру, **clBlack** (чёрный, задано), **clWhite** (белый), **clBlue** (синий), **clGreen** (зелёный), **clRed** (красный), **clYellow** (жёлтый), **clGrey** (серый), **clSilver** (серебрянный), **clAqua** (голубой) и еще более десятка постоянных.

Оператор `Image1.Canvas.Pen.color:=clRed;` меняет цвет рисуемой линии на красный. В **Delphi** можно использовать более 16 миллионов цветов. Все эти цвета получаются путём смешивания трёх основных цветов: **красного**, **зеленого** и **синего**.

Для каждого из основных цветов выделяется один байт. В этих трёх байтах можно хранить $2^4=16777216$ цветов. Порядковый номер цветов удобно писать в 16-ной системе счисления. В данной системе счисления трём байтам соответствует шестизначное число. Здесь последние две ячейки соответствуют красному цвету, две ячейки посередине – зелёному и левые две ячейки – синему цвету: `$BBGRR` (здесь **BB** числа, соответствующие синему цвету, **GG** –зелёному, **RR** – красному).

Например, синий цвет – `$FF0000`, зелёный `$00FF00`, красный – `$0000FF`, жёлтый – `$00FFFF`, розовый цвет – `$FF00FF`. Если работать с

шестнадцатиричными числами неудобно, то можно использовать стандартную функцию `RGB(RR, GG, BB)`.

У этой функции три аргумента, первый из них соответствует **красному**, второй – **зелёному**, третий – **синему**. Значения этих аргументов могут быть от 0 до 255. Например, `RGB(0, 0, 0)` выдаёт черный цвет, `RGB(255, 255, 255)` – белый, `RGB(0, 255, 255)` – **голубой**, `RGB(127, 0, 0)` – **красный**, `RGB(0, 127, 0)` – **зелёный**, `RGB(200, 200, 0)` – **жёлтый**.

Цвет объекта **Brush** (кисть) также определяется таким образом. Его стили можно можно изменить следующими постоянными:

1. **bsSolid** – гладкий (сплошной).
2. **bsClear** – скрытый.
3. **bsBDDiagonal** – по боковой диагонали.
4. **bsFDDiagonal** – по главной диагонали.
5. **bsCross** – решётчатый.
6. **bsDiagCross** – решётка по диагонали.
7. **bsHorizontal** – горизонтальный.
8. **bsVertical** – вертикальный.

3. Свойство Pixels (пикселей). С помощью свойства **Pixels Canvas** можно управлять цветом любой точки в области рисунка. В данном свойстве пиксели образуют двухмерный массив. Для обращения к пикселю, расположенному в 100 столбце на 50 строке, достаточно показать его индекс `Pixels[100,50]`. Если мы хотим поменять цвет данного пикселя на красный, используем оператор `Image1.Canvas.Pixels[100, 50]:=RGB(255, 0, 0);`.

А оператор `PC:=Image1.Canvas.Pixels[100, 50];` даёт возможность сохранить цвет пикселя в переменной **PC**.

4. Методы MoveTo va LineTo. В процессе рисования чертежа часто используется отрезок прямой. Какими возможностями обладает **Delphi**? Методы **MoveTo** (перейти к...) и **LineTo** (рисовать до...) объекта **Canvas** служат для рисования линий. Они оба имеют по два аргумента с целыми значениями. С помощью этих аргументов указываются координаты точек на рисунке.

Для того, чтобы начертить отрезок, концы которого расположены в точках (x_1, y_1) и (x_2, y_2) , используется метод **MoveTo(x1, y1)**, который совершает переход к первому концу отрезка, затем методом **Lineto(x2,y2)** прочерчивается отрезок до второго конца. Графический курсор переходит к точке (x_2, y_2) . Если ещё раз применить метод **Lineto(x3,y3)**, прочерчивается линия с концами в точках (x_2, y_2) и (x_3, y_3) и графический курсор переходит к точке (x_3, y_3) .

Практическое использование этих методов мы рассмотрим на примере приложения под названием **Диагонали правильного N угольника**. Создадим в **Delphi** новое приложение и вставим в него по одному объекту **Image**, **SpinEdit** и **Button**.

Заголовок **Button** меняем на **Рисовать**, а заголовок **SpinEdit** на **N**. Наибольшим значением **SpinEdit** (**MaxValue**) берём число 40, наименьшим значением (**MinValue**) – число 3, текущим значением (**Value**) число 7. Высотой **Image** (**Height**) и шириной (**Width**) берем число 500.

Дважды нажав на кнопку **Button**, введём в него команду **diagonal**. Такой команды пока нет, именно поэтому данную процедуру введём прежде процедуры **TForm1.Button1Click**:

```
procedure diagonal;
var i,j,n,r:integer;
x,y: array [0..40] of integer; begin
r:=Form1.Imagel.Height div 2;
Form1.Imagel.Canvas.Pen.Color:=clRed;
Form1.Imagel.Canvas.Rectangle(-1,-1,2*r+2,2*r+2);
n:=form1.SpinEdit1.Value;
for i:=1 to n do begin
y[i]:=round(r-r*cos(2*pi/n*i));
x[i]:=round(r+r*sin(2*pi/n*i)); end;
for i:=1 to n-1 do for j:=i+1 to n do begin
Form1.Imagel.Canvas.MoveTo(x[i],y[i]);
Form1.Imagel.Canvas.LineTo(x[j],y[j]); end; end;
```

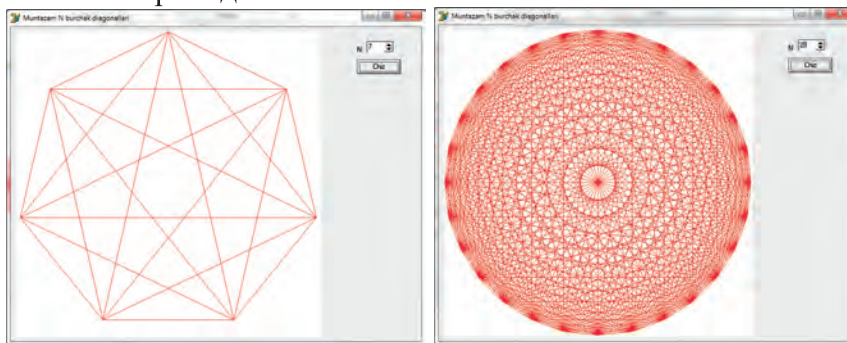
В эту процедуру ввели шесть переменных. Переменные i и j используются для управления циклом. В переменной n хранится

количество углов многоугольника, в переменной r хранится радиус круга, нарисованного вокруг правильного многоугольника. В массивах x и y хранятся координаты вершин многоугольника.

Операторов в теле процедуры можно разделить на три части. В первой части четыре оператора, первый из которых получает радиус круга, равный половине высоты рисунка. Второй оператор обеспечивает рисование линий красным цветом. Третий оператор до рисования очищает рисунок. Четвертый оператор определяет количество углов многоугольника.

Во второй части есть только один оператор цикла. В нём определяются координаты вершин многоугольника: $x[i]$ и $y[i]$.

В третьей части тоже имеется один оператор цикла, который рисует стороны и диагонали многоугольника. Загрузим приложение. Нарисуем сначала правильный семиугольник, затем правильный 28 угольник. Они приведены ниже:



ЗАПОМНИТЕ !

Элемент **Canvas** имеется во многих объектах и в нем можно рисовать.



ВОПРОСЫ И ЗАДАНИЯ

1. Как работают с графикой **Delphi**?
2. В каком месте панели объектов расположен объект **Image**?
3. Какую функцию выполняет **RGB**?
4. Создайте приложение, чтобы нарисовать на экране три прямоугольника красного, зеленого и чёрного цветов.

5. Создайте приложение, которое нарисует на экране три эллипса синего, серебряного и жёлтого цветов.



ДОМАШНЕЕ ЗАДАНИЕ

1. Перечислите основные свойства объекта Карандаш.
2. Какими свойствами обладает объект Щётка?



60 УРОК. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

1. Создайте приложения, выполняющие следующие графические задачи.

1.1. Создайте приложение, выводящее на экран 10 правильных четырёхугольников, расположенных один внутри другого.

1.2. Создайте приложение, выводящее на экран 12 квадратов, расположенных один внутри другого, с неизменным нижним левым углом.

1.3. Создайте приложение, выводящее на экран 10 правильных треугольников, расположенных один внутри другого.

1.4. Создайте приложение, выводящее на экран 15 кругов, расположенных один внутри другого с центрами в одной точке.

1.5. Создайте приложение, выводящее на экран 8 эллипсов, расположенных один внутри другого с центрами в одной точке.

1.6. Создайте приложение, выводящее на экран 5 прямоугольников, расположенных один внутри другого и выполненных разными цветами.

1.7. Создайте приложение, выводящее на экран 5 кругов, расположенных один внутри другого, заштрихованных линиями разной толщины.

1.8. Создайте приложение, выводящее на экран 5 эллипсов, нарисованных линиями различных стилей.

1.9. Создайте приложение, выводящее на экран 11 квадратов, расположенных один внутри другого, с постоянным верхним правым углом.

1.10. Создайте приложение, выводящее на экран 9 кругов, расположенных один внутри другого с центрами в верхней точке.

1.11. Создайте приложение, выводящее на экран 8 эллипсов, расположенных один внутри другого с общим центром в левой точке.

1.12. Создайте приложение, выводящее на экран 13 квадратов, расположенных один внутри другого, с постоянным верхним левым углом.

1.13. Создайте приложение, выводящее на экран 7 кругов, расположенных один внутри другого с общим центром в нижней точке.

1.14. Создайте приложение, выводящее на экран 14 эллипсов, расположенных один внутри другого с общими центрами в правой точке.

1.15. Создайте приложение, выводящее на экран 16 квадратов, расположенных один внутри другого, с постоянным нижним правым углом.

2. Создайте новое приложение и дайте ему имя «графические формы».

2.1. Вставьте в него одну кнопку управления и три группы радиокнопок.

2.2. Пусть первая группа радиокнопок служит для определения рисуемой формы. Возможные варианты: квадрат, правильный прямоугольник, круг и эллипс.

2.3. Вторая группа радиокнопок будет определять сторону выравнивания форм. Возможные варианты: посередине, слева, справа, сверху, снизу, сверху слева, сверху справа, слева снизу, справа снизу.

2.4. Третья группа радиокнопок будет определять цвет линий. Возможные варианты: красный, зелёный, синий, жёлтый, розовый, голубой, чёрный.

2.5. При нажатии на кнопку управления выбранная геометрическая форма в выбранном цвете, выбранном методе выравнивания один внутри другого выводится на экран.



61 УРОК.

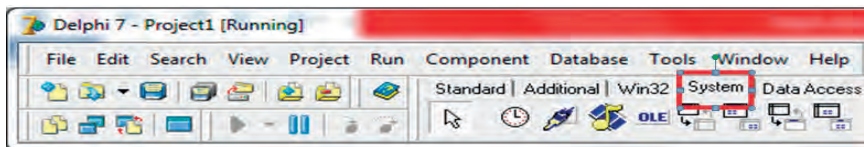
ОБЪЕКТ Timer И ЕГО ПРИМЕНЕНИЕ

1. Свойства объекта Timer. Иногда в объекте приходится выполнять действия в реальном времени. Например, если в

приложении имеются часы, показывающие текущее время, приходится каждую секунду обновлять вид часов. В противном случае часы отстанут. В программе тестирования знаний для выполнения задания отводится определённое время. С окончанием этого времени, прекращается сдача теста. В презентации каждые десять (двадцать) секунд на экран выводится новый слайд. В анимационных приложениях состояние движущейся формы в секунду изменяется несколько раз.

Во всех этих примерах используется объект **Timer**. Особенностью приложения **Timer** является его скрытость на экране. Если **Timer** находится в установленном состоянии, то ожидает назначенное время. Когда назначенное время наступает, приложение временно останавливает свою работу и выполняется специальная процедура **Timer**. После выполнения данной процедуры, приложение вновь продолжает свою работу.

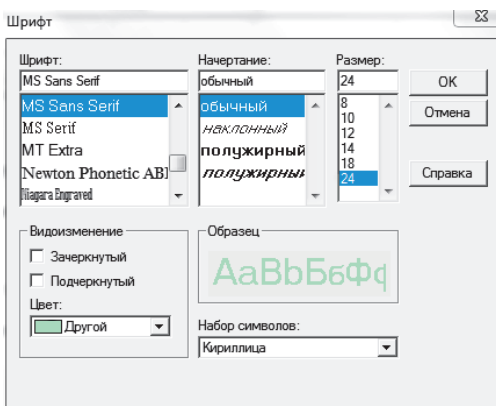
Объект **Timer** расположен во вкладке **System** (Система) панели объектов:



У объекта **Timer** есть всего два свойства. Первое –это знакомое нам свойство **Enabled** (доступный). Это свойство принимает истинные и ложные значения. Если его значение **True** (истина), **Timer** начинает работать. Если значение **False**, то **Timer** временно останавливает свою работу. Первоначально значение данного свойства равно **True**.

Вторым свойством **Timer** является **Interval**, в нём указывается время, когда **Timer** включится в следующий раз. Это время измеряется в миллисекундах. Если **Timer** должен включаться каждые две секунды, свойство интервала устанавливается на величину, равную 2000. Первоначальное его значение равняется 1000.

2. Электронные часы и приложение Timer. Создадим приложение электронных часов. Создав новое приложение, изменим его заголовок на электронные часы. В окно приложения вставим два объекта: **Label** и **Timer**. Нажав дважды на свойство **Font** (шрифт) **Label**, в диалоговом окне шрифта установим размер шрифта на значении 24:



Нажав дважды на объект **Timer**, введём в его процедуру оператор **Label1.Caption:=TimeToStr(Now)**; В данном операторе используется две стандартные функции. Первой из них является **Now** (сейчас), её значение равно текущему состоянию системных часов.



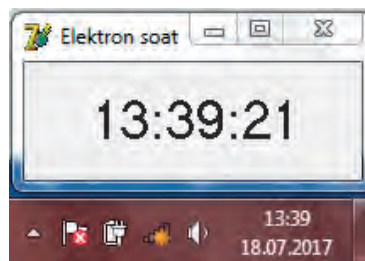
Второй функцией является **TimeToStr** (**Time to String** от вида времени к виду строки текста), значение которой состоит в виде строки текста. Это значение можно дать свойству **Caption** объекта **Label1**.

3. Приложение Timer. Создав новое приложение, вставим в его окно объекты **Label**, **Button**, **SpinEdit** и **Timer**. Заголовки **Label** и **Button** заменим на **ОСТАВШЕЕСЯ ВРЕМЯ:** и **START**.

Свойства **SpinEditning** установим в следующем виде:

```
SpinEdit1.MaxValue:=100;
SpinEdit1.Value:=30;
SpinEdit1.Enabled:=False;
```

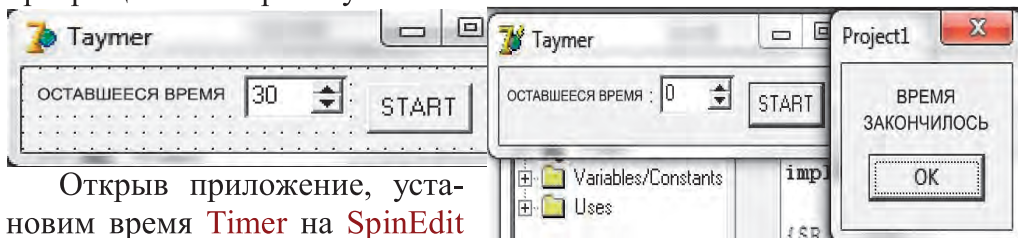
Нажав дважды на кнопку **Button**, введём в его процедуру оператор **Timer1.Enabled:=True**; Нажав дважды на



Timer, в процедуру `TForm1.Timer1Timer` введем следующий код программы:

```
SpinEdit1.Value:=SpinEdit1.Value-1;  
if SpinEdit1.Value=0 then  
begin  
Timer1.Enabled:=False;  
ShowMessage('Время закончилось');  
Form1.Close;  
end;
```

Каждый раз, когда включается данная процедура, оператор на её первой строке уменьшает значение **SpinEdit** на единицу. Второй оператор проверяет, равно ли значение **SpinEdit** нулю. Если 0, оператор четвёртой строки останавливает работу **Timer**, на пятой строке на экран выводит диалоговое окно. С закрытием этого окна оператор с шестой строки закрывает окно приложения и приложение прекращает свою работу.



Открыв приложение, установим время **Timer** на **SpinEdit** и нажав кнопку Старт, включаем **Timer**. Время в **Timer** будет уменьшаться и с его окончанием на экран выводится диалоговое окно. С его закрытием приложение прекращает свою работу.

4. Автоматизация приложения диагоналей многоугольника.

Автоматизируем работу приложения под названием диагонали правильного N угольника, созданного на предыдущем уроке. При нажатии его кнопки приложение переходит в автоматический режим и ежесекундно увеличивается количество углов многоугольника, рисуются все его диагонали. При втором нажатии кнопки выходит из автоматического режима, переходит в режим ожидания. При еще одном нажатии кнопки приложение продолжает свою работу.

Перезагрузим приложение в **Delphi**, вставим в него объект **Timer**. Значение его свойства **Enabled** переведем на Ложь. Заменяем заголовок кнопки **Button1** на **Start**. Дважды нажав на эту кнопку, в полученную процедуру введём следующие два оператора:

```
Form1.Timer1.Enabled:=not Form1.Timer1.Enabled;  
if Form1.Timer1.Enabled=True then  
Form1.Button1.Caption:='Stop' else  
Form1.Button1.Caption:='Start'
```

Оператор первой строки изменяет активность **Timer1** с помощью свойства **Enabled** на противоположное. А второй оператор, исходя из значения **Enabled**, меняет надпись на кнопке на **Stop** или **Start**.

Дважды нажав на **Timer1**, в его процедуру введём следующие три оператора:

```
Form1.SpinEdit1.Value:=form1.SpinEdit1.Value+1;  
if Form1.SpinEdit1.Value>40  
then Form1.SpinEdit1.Value:=3; диагонали;
```

Первый из этих операторов увеличивает значение **SpinEdit1** на единицу. Если значение превысит 40, следующий оператор заново определяет его значение равным 3. Третий оператор рисует стороны и диагонали данного многоугольника.

Запустив приложение, нажимаем на его кнопку. Наблюдаем за процессом рисования диагоналей многоугольника на экране. В нужный момент, ещё раз нажав на кнопку, останавливаем работу приложения и внимательно рассматриваем диагонали многоугольника на экране. Затем ещё раз нажав на кнопку, восстановим процесс рисования следующих многоугольников.



ЗАПОМНИТЕ !

Для создания анимации используется объект **Timer**.



ВОПРОСЫ И ЗАДАНИЯ

1. Приведите примеры использования **Timer**.
2. В какой вкладке панели объектов расположен объект **Timer**?

3. Какую функцию выполняет интервальное свойство **Interval** объекта **Timer**?
4. Какие значения принимает свойство **Enabled** объекта **Timer** и что происходит при изменении данного значения?
5. В приложении **Timer**, используя объект **Label** вместо процедуры **ShowMessage**, создайте его заново.
6. Измените приложение Электронные часы так, чтобы часы каждую секунду издавали короткий сигнал.



ДОМАШНЕЕ ЗАДАНИЕ

1. Сколько свойств у объекта **Timer**?



62 УРОК. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

1. Вставьте объект **Timer** в приложения, созданные на предыдущих уроках и выводите геометрические фигуры на экран каждые 2 секунды.

2. Создайте программу часов с использованием объектов **Button** и **Label**.

3. Создайте приложение, у которого ширина окна меняется с шагом 1 в диапазоне от 100 до 800 с частотой 20 раз в секунду.

4. Разработайте приложение светофор. Пусть красный и зелёный сигналы горят 5 секунд, а жёлтый – 1 секунду. Погаснув, они должны принимать серый цвет.

4.1. Создайте окно приложения и вставьте в него объект **Timer**.

4.2. Перейдя в окно кода приложения, найдите оператор **var Form1: TForm1**; и введите оператор **var i:integer** после него.

4.3. Щелкну дважды мышью на объекте **Timer**, введите в его процедуру следующие операторы:

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  i:=i+1;
  if i mod 12 =1 then
  begin light_off;
  canvas.brush.Color:=clRed;
  canvas.Ellipse(20,20,70,70);
  end;
```

```

if i mod 6 =0 then
begin light_off;
canvas.brush.Color:=clYellow;
canvas.Ellipse(80,20,130,70);end;
if i mod 12 =7 then
begin light_off;
canvas.brush.Color:=clGreen;
canvas.Ellipse(140,20,190,70);
end; end;

```

4.4. Перед этой процедурой введите процедуру: `light_off`:

```

procedure light_off;
begin
form1.canvas.brush.Color:=clBlue;
form1.canvas.Rectangle(10,10,200,80);
form1.canvas.brush.Color:=clGray;
form1.canvas.Ellipse(20,20,70,70);
form1.canvas.Ellipse(80,20,130,70);
form1.canvas.Ellipse(140,20,190,70);
end;

```

4.5. Загрузите приложение и наблюдайте за его работой:



63 УРОК.

РАЗМЕЩЕНИЕ ДРУГИХ ОБЪЕКТОВ В РИСУНОК

1. Вставка текста в рисунок. При работе с рисунками часто приходится вставлять текст в рисунок. Для выполнения этого действия используется метод `TextOut` (вывод текста) из `Canvas`. Он имеет три аргумента. Первые два из них – координаты левого верхнего угла области вывода текста, а третий – строка текста, выводимого на экран.

Например, оператор `Image1.Canvas.TextOut(10,20, 'Salom!')`; начиная с точки (10, 20) области рисунка, выводит надпись 'Здравствуйте!'.

Шрифт выводимого на рисунок текста определяется свойством `Font` (шрифт) объекта `Canvas`. Например, чтобы изменить размер

шрифта на 14, используется оператор `Image1.Canvas.Font.Size:=14`, для изменения цвета шрифта в красный используется оператор `Image1.Canvas.Font.Color:=clRed`;

Для ознакомления с применением данного оператора создадим следующее приложение. Вставим в него кнопки `Image` и `Button`, дважды щелкнем мышью на кнопке и введём следующие операторы:

```
Image1.Canvas.Font.Size:=8;
Image1.Canvas.TextOut(10,20, '8');
Image1.Canvas.Font.Size:=14;
Image1.Canvas.Font.Color:=clRed;
Image1.Canvas.TextOut(20,20, '14');
Image1.Canvas.Font.Size:=20;
Image1.Canvas.Font.Color:=clBlue;
Image1.Canvas.TextOut(40,20, '20');
Image1.Canvas.Font.Name:='Times New Roman';
Image1.Canvas.Font.Size:=30;
Image1.Canvas.Font.Color:=clGreen;
Image1.Canvas.TextOut(70,20, '30');
Image1.Canvas.Font.Size:=40;
Image1.Canvas.Font.Color:=clBlack;
Image1.Canvas.TextOut(110,20, '40');
Image1.Canvas.Font.Size:=50;
Image1.Canvas.Font.Color:=RGB(255,0,255);
Image1.Canvas.TextOut(180,20, '50');
```

Запустив приложение, нажимаем его кнопку.

Обратите внимание на то, что после изменения нами шрифта на знакомый нам `Times New Roman`, качество текста на экране заметно улучшилось.

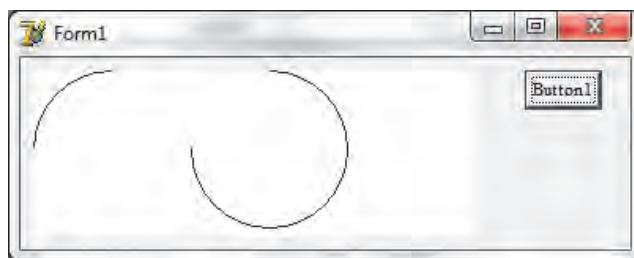
2. **Вставка фигур в рисунок.** В рисунок можно вставить ряд геометрических фигур. На прошлых уроках мы рассматривали вставку в рисунок прямоугольника, эллипса, отрезка. Теперь научимся вставлять другие фигуры.

Для того, чтобы вставить дугу эллипса в рисунок, используется метод `arc(x1,y1,x2,y2, x3,y3,x4,y4)`. При этом берется часть эллипса,

расположенного между радиусами, соединяющими центр эллипса с такими (x_3, y_3) и (x_4, y_4) . Сам эллипс встроено в прямоугольник с вершинами (x_1, y_1) и (x_2, y_2) . Здесь движение осуществляется от третьей точки к четвертой против часовой стрелки.

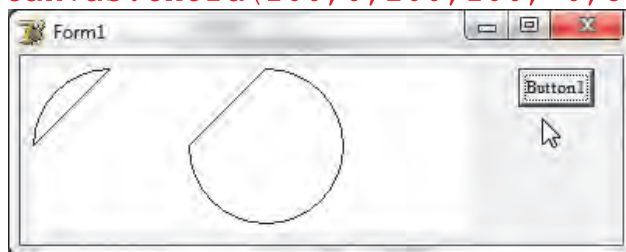
Операторы, рисующие дугу эллипса и нарисованные ими дуги, показаны ниже:

```
Image1.Canvas.Arc(0,0,100,100, 50,0,0,50);  
Image1.Canvas.Arc(100,0,200,100, 0,50,150,0);
```

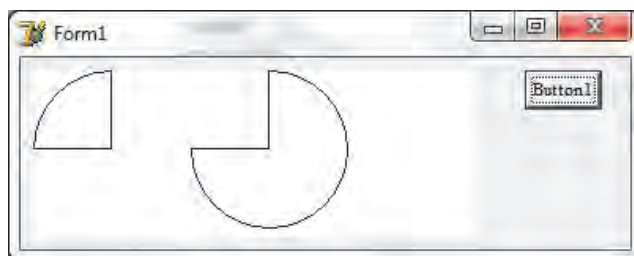


Чтобы вставить сегмент эллипса в рисунок, используют метод `Chord(x1,y1,x2,y2, x3,y3, x4,y4)`. Вот пример:

```
Image1.Canvas.Chord(0,0,100,100, 50,0,0,50);  
Image1.Canvas.Chord(100,0,200,100, 0,50,150,0);
```



Чтобы вставить сектор эллипса в рисунок, используют метод `Pie(x1,y1,x2,y2, x3,y3,x4,y4)`.





```
Image1.Canvas.pie(0,0,100,100, 50,0,0,50);  
Image1.Canvas.pie(100,0,200,100, 0,50,150,0);
```



ЗАПОМНИТЕ !

В рисунок можно вставить текст и геометрические фигуры.



ВОПРОСЫ И ЗАДАНИЯ

1. Как в рисунок вставляется сегмент эллипса?
2. Как в рисунок вставляется сектор эллипса?
3. Создайте приложение, вставив в рисунок несколько строк текста.
4. Создайте приложение, вставив в рисунок несколько геометрических

фигур.



ДОМАШНЕЕ ЗАДАНИЕ

1. Как вставляется текст в рисунок?
2. Как вставляется дуга эллипса в рисунок?



64 УРОК. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

1. Создайте новое приложение. Нарисуйте в нём систему координат. Введите в рисунок название, стрелки и числа в координатных осях.
2. Создайте новое приложение. Загрузите в него рисунок из файла компьютера. Название файла рисунка поместите справа рисунка.
3. Создайте новое приложение. Нарисуйте на нём несколько дуг эллипса разных цветов.

4. Создайте новое приложение. Нарисуйте на нём несколько сегментов круга.

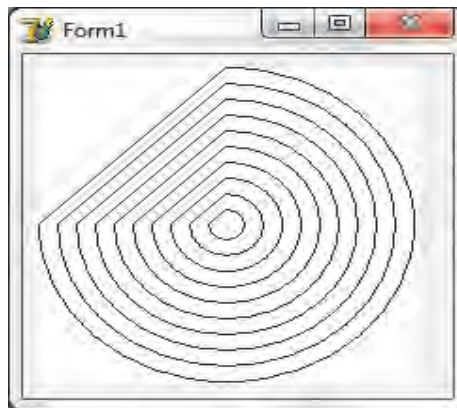
5. Создайте новое приложение. Вставьте в него 10 сегментов эллипса, вложенных друг на друга.

5.1. В окно приложения вставьте объект **Image** .

5.2. Щелкнув дважды мышью на пустое место в окне приложения, введите следующую процедуру:

```
procedure TForm1.FormCreate(Sender: TObject);  
var i:integer;  
begin  
for i:=10 downto 1 do  
image1.Canvas.Chord(100-10*i,100-  
10*i,100+10*i,100+10*i, 0,100,100,0);  
end;
```

5.3. Откройте приложение. Проверьте результат работы программы.



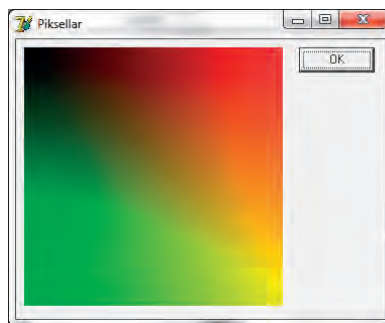
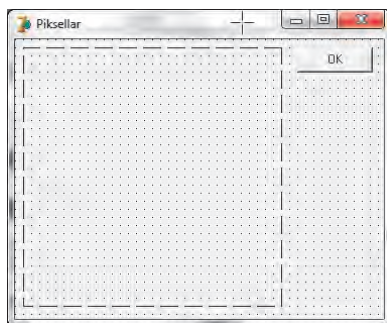


65 УРОК. ПРИМЕРЫ ПРИЛОЖЕНИЙ С ГРАФИКОЙ И АНИМАЦИЕЙ

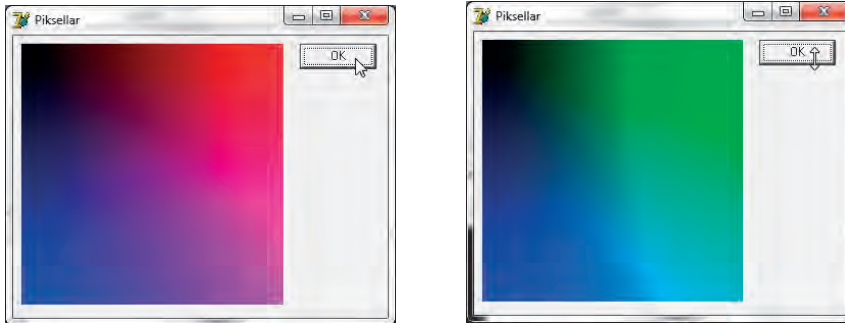
1. Палитра цветов. С помощью свойства `Pixels` создадим приложение, которое путём смешивания красного и зелёного цветов в разных пропорциях выводит на экран 65536 пикселей с разными цветами. В приложение вставим объекты `Image` (256×256) и `Button` как показано на рисунке ниже. Дважды нажав на кнопку `Button`, введём в код программы следующую процедуру:

```
procedure TForm1.Button1Click(Sender: TObject);  
var i,j,цвет:integer;  
begin  
  for i:=0 to 255 do  
    for j:=0 to 255 do begin  
      цвет:=RGB(i,j,0);  
      Image1.Canvas.Pixels[i,j]:= цвет;  
    end;  
  end;
```

Открыв приложение, смешав **красный** и **зелёные** цвета, получим **жёлтый** цвет:



Поменяв в приложении оператор `color:=RGB(i,j,0)` на оператор `color:=RGB(i,0,j)`, смешав красный и синий цвета, получаем розовый цвет, поменяв на `color:=RGB(0,i,j)`, смешав зелёный и синий цвета, получаем голубой цвет:



2. Приложение Мячик. Объект `Timer`, в первую очередь, применяется для создания анимационного движения. Чтобы увидеть, как это делается, создадим новое приложение. Вставим в него достаточно большой объект `Image`, а также по одному `Timer`, `CheckBox` и `SpinEdit`.

Пусть мяч движется по области окна приложения `Image`. При ударе о стенку мяч возвращается назад. Скорость его движения определяет значение в `SpinEdit` и с его изменением пусть изменится и скорость мяча. Если в `CheckBox` установлен флажок, пусть выводятся и следы траектории движения, в противном случае пусть выводится только текущее положение мяча. Состояние мяча обновляется 20 раз в секунду (с интервалом в 50 миллисекунд).

Изменим заголовок `CheckBox` на название `с траекторией`, свойство `Checked` установим как истину. Максимальное значение `SpinEdit` возьмём равным 10. Значение свойства `Interval` у `Timer` приравниваем к 50.

Приложению понадобятся две переменные `x`, `y` для хранения текущего состояния мяча, для хранения текущего направления ещё две переменные: `dx`, `dy`. Описывающий их оператор `var x,y,dx,dy:integer` разместим в окно кода программы после `var Form1: TForm1`. Чтобы

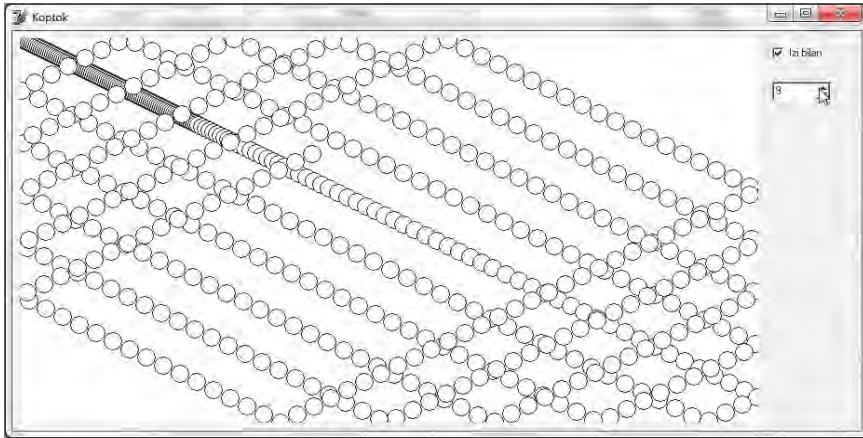
дать этим переменным исходные значения, дважды нажмём на пустое от других объектов место **Form1**. На появившейся в окне программы процедуре **TForm1.FormCreate** введём операторы $x:=0$; $y:=0$; $dx:=2$; $dy:=1$.

Дважды щелкнув на объекте **Timer**, заполним его процедуру следующим образом:

```
procedure TForm1.Timer1Timer(Sender: TObject);
var n:integer;
begin
n:=SpinEdit1.Value;
if not CheckBox1.Checked then
  Image1.Canvas.Rectangle(-3,-
3,Image1.Width+3,Image1.Height+3); x:=x+dx*n;
if x>Image1.Width then
begin x:=2*Image1.Width-x; dx:=-dx;end;
if x<0 then begin x:=-x; dx:=-dx;end;
y:=y+dy*n;
if y>Image1.Height then
begin y:=2*Image1.Height-y; dy:=-dy;end;
if y<0 then begin y:=-y; dy:=-dy;end;
Image1.Canvas.Ellipse(x-10,y-10,x+10,y+10);
end;
```

Во второй строке описывается новая переменная **n**, которая используется в процедуре. В четвёртой строке этой переменной присваивается значение скорости, введенное в **SpinEdit1**.

В пятой строке проверяется, установлен ли флажок на **CheckBox1**. Если этот флажок не установлен, в шестой строке, стирая весь рисунок, можно избавиться от прежнего состояния мяча:



В седьмой строке находится координата x нового состояния мяча. В восьмой строке проверяется, не пересекается ли новое состояние с правой границей рисунка. Если пересекается, то находится новое значение x и направление горизонтального движения dx меняется на противоположное. В девятой строке проверяется, пересекается ли новое состояние с левой границей рисунка. Если пересекается, то находится новое значение x , и направление горизонтального движения меняется на противоположное.

С десятой по двенадцатую строки таким же образом вычисляется координата y мяча. И наконец, в тринадцатой строке новое состояние мяча рисуется на экране.

Запустив приложение, проверим правильность его работы.



ЗАПОМНИТЕ !

В **Delphi** имеется более 16 миллионов цветов. Все эти цвета получаются путём смешивания трёх основных цветов: красного, зелёного и синего.



ВОПРОСЫ И ЗАДАНИЯ

1. В приложении Pixels сколько пикселей меняют свой цвет?

2. При помощи каких переменных определяется состояние мяча в приложении Мячик?
3. Каким образом изменяется скорость мяча в приложении Мячик?
4. В какой переменной хранится направление движения мяча в приложении Мячик?
5. Обеспечьте участие третьего из основных трёх цветов в формировании цветов в приложении Pixels.
6. Заменяв первоначальные значения переменных dx и dy , измените направление движения мяча в приложении Мячик:
 - a) $dx:=1; dy:=1;$
 - b) $dx:=1; dy:=2;$
 - c) $dx:=3; dy:=2;$
 - d) $dx:=3; dy:=1.$



ДОМАШНЕЕ ЗАДАНИЕ

1. Как получить соответствующий цвет вершин квадрата в приложении Pixels и смещением каких цветов это можно сделать?



66 УРОК. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ

1. В приложении Pixels при смешивании двух цветов добавьте третий основной цвет.

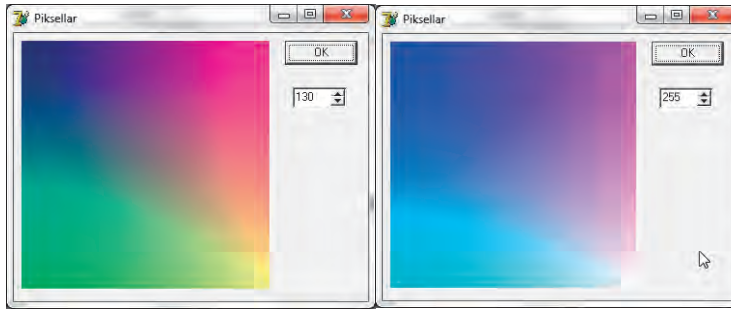
Для этого загрузите вариант приложения пикселей по смешиванию красного и зелёного цветов.

1.1. Поместите в него один объект **SpinEdit**. Из его свойств поменяйте следующие.

MaxValue:=255; Value:=130; Increment:=5;

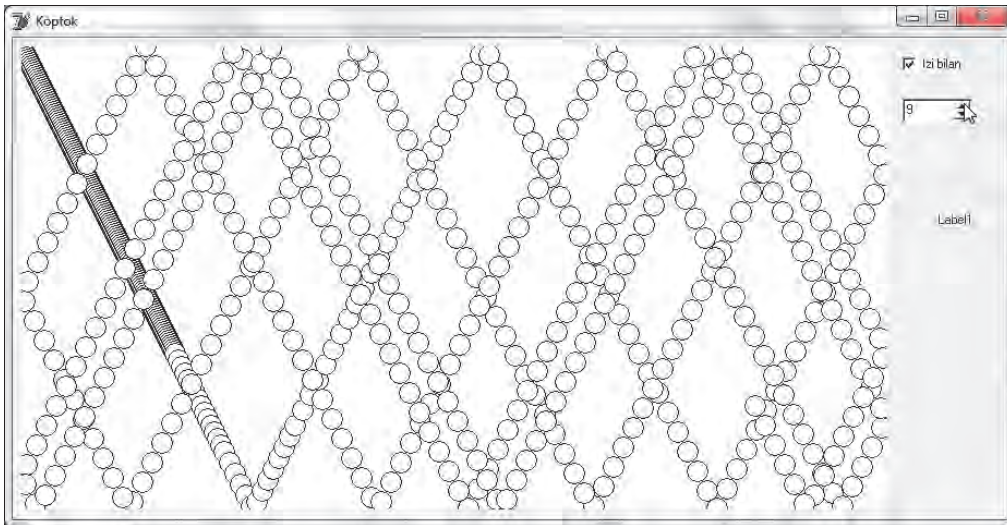
1.2. В программном коде приложения **rang:=RGB(i,j,0)** заменяем на **rang:=RGB(i,j,SpinEdit1.Value);**

1.3. Начав работу приложений, изучите влияние **синего** цвета на смесь **красного** и **зелёного** цветов:



2. Добавив третий из основных цветов и в другие два цвета создайте варианты приложения Pixels. Понаблюдайте за изменениями. На рисунке выше показано влияние зелёного цвета на смесь **красного** и **синего** цветов.

3. Измените приложение мячик так, чтобы движение мяча было таким, как показано на рисунке.





67 УРОК. КОНТРОЛЬНАЯ РАБОТА

Ниже приведены образцы вариантов контрольных работ:

1 вариант

1. Объект **Pen** объекта **Canvas**.
2. Вставка эллипса в рисунок.
3. Два способа перевода карандаша в чёрный цвет.

2 вариант

1. Объект **Brush** объекта **Canvas**.
2. Вставка прямоугольника в рисунок.
3. Два способа перевода карандаша в жёлтый цвет.

3 вариант

1. Объект **Pixels** объекта **Canvas**.
2. Рисование отрезка на рисунке.
3. Два способа перевода кисти в белый цвет.

4 вариант

1. **Canvas**.
2. Ввод текста в рисунок.
3. Два способа перевода кисти в зелёный цвет.



68 УРОК. ВОПРОСЫ ДЛЯ ПОВТОРЕНИЯ

1. Что вы знаете о методах программирования?
2. Перечислите особенности интерфейса программной среды **Delphi**.
3. Назовите свойства объект окна приложения **Form** и их функции.
4. Что вы знаете об основных свойствах кнопки управления **Button**?
5. Что произойдёт, если щёлкнуть мышью над кнопки **Button**?
6. Что вы знаете об окне **ShowMessage**?

7. Назовите предназначение объекта **Label**.
8. Предназначение и свойства объекта **Edit**.
9. Посредством каких свойств устанавливается доступность и видимость объектов?
10. Расскажите о функциях перевода значений переменных из одного типа в другой.
11. Каким объектом вводятся в приложение флажки?
12. Перечислите основные свойства группы Радиокнопок **RadioGroup**?
13. Что вы знаете об объекте **ListBox**?
14. Чем удобен объект **ComboBox**?
15. Назовите предназначение и свойства объекта **Memo**, его объекты.
16. Как осуществляется работа с графикой в **Delphi**?
17. Что вы знаете о работе с карандашом?
18. Назовите основные свойства кисти.
19. Как вводится текст в рисунок?
21. Как нарисовать прямоугольник в приложении?
22. Как нарисовать эллипс в приложении?
23. Расскажите о работе с цветами в рисунке.
24. Что вы знаете о работе с пикселями?
25. Назовите предназначение и свойства объекта **Timer**.

ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

1. Постановление Кабинета Министров Республики Узбекистан от 6 апреля 2017 года №187 “Об утверждении государственных образовательных стандартов среднего и среднего специального, профессионального образования”
2. Ahmedov A., Taylaqov N. Informatika. AL va KHK uchun darslik. –Т.: O‘zbekiston, 2002. 2-nashri. -272 b.
3. Nazirov Sh. A., Musayev M.M., Ne‘matov A., Qobulov R.V. Delphi tilida dasturlash asoslari. - 2007. -324 b.
4. Крыгин С.В. Microsoft Excel 2010. Учебное-практическое пособие: Н. -Новгород, 2011. -72 с.
5. Карчевский Е.М., Филиппов И.Е., Филиппова И.А. Access 2010 в примерах: Учебное пособие. -Казань, 2012. -140 с.
6. Taylaqov N.I EHM ishlari va hisob usuli. KHK uchun darslik. –Т.: «Ijod duyosi» nashryot uyi, 2003. - 224 b.
7. Холматов Т.Х, Тайлақов Н.И. Амалий математика, дастурлаш ва компьютернинг дастурий таъминоти. Олий ўқув юртлари учун ўқув кўланма.-Т.: Мехнат, 2000. -304 б.
8. Холматов Т.Х., Тайлақов Н.И., Назаров У.А. Информатика. Олий ўқув юртлари учун дарслик. -Т.: Ўзбекистон миллий энциклопедияси, 2003.-254 б.
9. Taylaqov N.I. Informatika va hisoblash texnikasi asoslari. 9-sinf uchun darslik. –Т.: «Ijod duyosi» nashryot uyi, 2002. -152 b.

ОГЛАВЛЕНИЕ

Предисловие	3
ГЛАВА I. ДЕЙСТВИЯ И ФОРМУЛЫ В MS EXCEL 2010	
1 УРОК. Вычисление простых выражений	4
2 УРОК. Ссылка на клетку: относительная, абсолютная и смешанная ссылка	8
3 УРОК. Преимущества копирования при выполнении действий, используя ссылки	10
4 УРОК. Графики простых и сложных функций	12
5 УРОК. Ссылка на другой лист или книгу	14
6 УРОК. Библиотека функций MS Excel	18
7 УРОК. Окно аргумента функции	20
8 УРОК. Использование строку формул	22
9 УРОК. Текстовые функции	24
10 УРОК. Логические функции.....	26
11 УРОК. Практические занятия для закрепления	27
12 УРОК. Контрольная работа	29
13 УРОК. Математические функции	31
14 УРОК. Функции для вычисления произведения	34
15 УРОК. Статистические функции	35
16 УРОК. Практическое занятие	37
17 УРОК. Решение некоторых задач в MS Excel	38
18 УРОК. Практическое занятие.....	40
19 УРОК. Контрольная работа	41
ГЛАВА II. БАЗА ДАННЫХ	
20 УРОК. Понятие о базе данных	43
21 УРОК. Системы управления базой данных	44
22 УРОК. Практическое занятие	46
23 УРОК. Основные элементы MS Access 2010 и свойства полей	47
24 УРОК. Практическое занятие.....	50
25 УРОК. Создание базы данных в MS Access 2010	51
26 УРОК. Практическое занятие.....	54
27 УРОК. Соединение таблиц в MS Access 2010	55
28 УРОК. Практическое занятие.....	60
29 УРОК. Поиск и обработка данных по указанному шаблону в MS Access 2010	61
30 УРОК. Создание базы данных в MS Access 2010 и использование формул при редактировании	65
31 УРОК. Практическое занятие.....	69
32 УРОК. Выполнение математических операций в MS acces 2010	70

33 УРОК. Задания для повторения.....	74
34 УРОК. Контрольная работа	75
ГЛАВА III. СОЗДАНИЕ ПРИЛОЖЕНИЙ В СРЕДЕ DELPHI	
35 УРОК. Современные методы создания приложений	79
36 УРОК. Среда программирования Delphi	83
37 УРОК. Приложение и его окно	87
38 УРОК. Практическое занятие.....	89
39 УРОК. Кнопка управления	90
40 УРОК. Практическое занятие.....	92
41 УРОК. Окно Showmessage	93
42 УРОК. Контрольная работа	95
43 УРОК. Размещение данных в окне приложений	96
44 УРОК. Практическое занятие.....	98
45 УРОК. Доступность и видимость объектов управления	99
46 УРОК. Практическое занятие.....	101
47 УРОК. Ввод данных в приложение	102
48 УРОК. Практическое занятие.....	106
49 УРОК. Изменение типа данных	107
50 УРОК. Контрольная работа	110
51 УРОК. Применение флажков в приложении	111
52 УРОК. Практическое занятие.....	115
53 УРОК. Использование группы радиокнопок в приложении	116
54 УРОК. Практическое занятие.....	120
55 УРОК. Объекты Listbox и Combobox	121
56 УРОК. Практическое занятие.....	125
57 УРОК. Объект управления Memo	126
58 УРОК. Контрольная работа	130
59 УРОК. Работа с графикой в Delphi	130
60 УРОК. Практическое занятие.....	136
61 УРОК. Объект Timer и его применение	137
62 УРОК. Практическое занятие.....	142
63 УРОК. Размещение других объектов в рисунке	143
64 УРОК. Практическое занятие.....	146
65 УРОК. Примеры приложений с графикой и анимацией	148
66 УРОК. Практическое занятие.....	153
67 УРОК. Контрольная работа	154
68 УРОК. Вопросы для повторения	154
Использованная литература	156

Учебник для учащихся 10 классов средних и средних специальных,
профессиональных учебных заведений

ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Тайлаков Норбек Исакулович
Ахмедов Акром Бурхонович
Пардаева Мехринисо Данияровна
Абдуганиев Абдували Абдулхаевич
Мирсанов Уралбой Мухаммадиевич

Редактор: Покачалова Н.С.
Художественный редактор: Убайдуллаев З.
Технический редактор: Мадияров К.
Корректор: Маликова И.
Набор на компьютере: Ходжабаев Ф.
Компьютерный дизайн: Ахмедов А.
Перевод: Абдукадыров А., Закиров Ш.

Номер лицензии АИ № 296 от 22.05.2017. Разрешено к печати
01.10.2017. Формат 70x90 1/16. Гарнитура Times New Roman. Усл.-
печ.л. 9,0. Печ.л. 8,56. Тираж 49607 экз.

Заказ № 405.

Договор № 11-2017.

Отпечатано в издательстве-типографии творческого дома имени
Гафура Гуляма агентства печати и информации Узбекистана
100128, Ташкент, улица Лабзак, дом 86.

Таблица состояния учебников, данных в аренду

№	Фамилия, имя учащегося	Учебный год	Состояние учебника перед арендой	Подпись классного руководителя	Состояние учебника при сдаче	Подпись классного руководителя
1						
2						
3						
4						
5						
6						

Данная таблица об аренде учебников и их возвращении в конце учебного года. Заполняется классным руководителем на основе нижеследующего критерия оценок:

Новый	Состояние учебника, даваемого пользоваться в первый раз
Хороший	Обложка целая, не отделена от основной части учебника. Все страницы сохранены, не порваны, не передвинуты, на страницах записей и чёрточек нет
Удовлетворительный	Обложка помятая, изрядно исписанная, края потёрты, отделяется от основной части, частично отреставрирована пользователем. Отделившиеся листы подклеены, некоторые страницы исчерчены
Неудовлетворительный	Обложка исчерчена, порвана, отделена от основной части или вовсе отсутствует, небрежно отреставрирована. Листы порваны, некоторые отсутствуют, исписаны и исчерчены. Учебник не подлежит восстановлению