

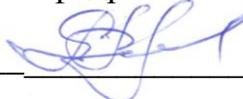
МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Тульский государственный университет»

Институт горного дела и строительства
Кафедра «Санитарно-технические системы»

Утверждено на заседании кафедры
«Санитарно-технические системы»
«12» января 2021 г., протокол № 6

Заведующий кафедрой



Р.А. Ковалев

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению лабораторных работ
по дисциплине (модулю)
«Информатика»

основной профессиональной образовательной программы
высшего образования – программы бакалавриата

по направлению подготовки
08.03.01 – "Строительство"

с профилем
"Теплогазоснабжение и вентиляция"

Форма(ы) обучения: очная, заочная

Идентификационный номер образовательной программы: 080301-06-21

Тула 2021 год

Разработчик(и) методических указаний

Белоусов Р.О., доцент, к.т.н., доцент
(*ФИО, должность, ученая степень, ученое звание*)



(подпись)

Обработка данных средствами электронных таблиц Microsoft Excel

Номер текущей ячейки Столбец текущей ячейки Содержимое текущей ячейки Панель стандартная Панель форматирования

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Наименование продукции	Цена за ед., руб.	Продано, шт.	Выручка от продажи, руб.										
2														
3	Модем	1460	10	14600										
4	Принтер	2500	15	37500										
5	Монитор, 17"	5750	20	115000										
6	Компьютер	19899	25	497475										
7	Цифровая фотокамера Переносной	10900	4	43600										
8	накопитель, 128 Mb	1535	6	9210										
9	Сканер	3050	7	21350										
10	Жесткий диск, 80 Gb	2840	13	36920										
11	Ноутбук	51470	1	51470										
12	CD-ROM	745	4	2980										
13	CD-ReWriter	1550	6	9300										
14	Итого, сумма выручки, руб.			839405										
15														
16														
17														
18														
19														
20														
21														
22														
23														
24														
25														
26														
27														
28														
29														
30														

Номер текущей строки Текущая ячейка Маркер заполнения

Рис. 1.1. Рабочий лист электронной таблицы Excel

Документ Excel называется *Рабочей книгой*. Рабочая книга представляет собой набор *рабочих листов*, каждый из которых имеет табличную структуру и может содержать одну или несколько таблиц. В окне документа Excel отображается только *текущий* рабочий лист, с которым и ведется работа (см. рис. 3.1). Рабочий лист состоит из *строк* и *столбцов*. Столбцы озаглавлены прописными латинскими буквами и, далее, двухбуквенными комбинациями. Всего рабочий лист может содержать до 256 столбцов, пронумерованных от А до IV. Строки последовательно нумеруются цифрами от 1 до 65536.

Одна из ячеек всегда является активной и после щелчка на нее мышью выделяется *рамкой активной ячейки*. Эта рамка играет роль курсора. Операции ввода и редактирования всегда производятся в активной ячейке.

Редактирование рабочей книги

Цель работы: создание и сохранение электронной таблицы (рабочей книги).

Изучение способов работы с данными в ячейке (форматирование содержимого ячеек, выбор диапазона ячеек и работа с ними, редактирование содержимого ячеек). Изучение возможностей автозаполнения.

Методика выполнения работы

Создайте новую рабочую книгу (кнопка *Создать* на стандартной панели инструментов или меню *Файл* команда *Создать как*).

Переименуйте текущий рабочий лист (дважды щелкните на ярлыке текущего рабочего листа и переименуйте его).

Добавьте еще один рабочий лист в рабочую книгу (щелкните правой кнопкой мыши на ярлыке листа и в *контекстном меню* выберите команду *Добавить*).

Сохраните созданный Вами файл под именем *book.xls* в своем каталоге (меню *Файл*, команда *Сохранить*).

Создайте таблицу по предложенному образцу (см. табл. 1.1). Для этого нужно выполнить следующие действия:

- в ячейку *A1* ввести заголовок таблицы “Экзаменационная ведомость”;
- в ячейку *A3* ввести “№ п/п”;
- в ячейку *B3* ввести “Фамилия, имя, отчество”;
- в ячейку *C3* ввести “№ зачетной книжки”;
- в ячейку *D3* ввести “Оценка”;
- в ячейку *E3* ввести “Фамилия экзаменатора”.

Пример выполнения пятого пункта задания.

Таблица 3.1

№п/п	Фамилия, имя, отчество	№ зачетной книжки	Оценка	Фамилия экзаменатора
1	Иванов И. И.	3-2330/12		Шевелев Г.Е.
2	Петров В. В.	3-2331/21		Огородников А.С.
3	Сидоров С. С.	3-6230/09		Шевелев Г.Е.
4	Федоров Ф. Ф.	3-11230/23		Огородников А.С.
5	Фролов Е.Е.	3-6230/05		Шевелев Г.Е.
6	Демидов Д. Д.	3-3530/05		Огородников А.С.

■ Отформатируйте ячейки шапки таблицы:

выделите блок ячеек (A3:E3);

выполните из меню *Формат* команду *Ячейки* и откройте вкладку *Выравнивание*;

в диалоговом окне *Выравнивание* выберите опции:

Горизонтальное – по центру, *Вертикальное* – по верхнему краю;

установите флажок *Переносить по словам*;

откройте вкладку *Шрифт* и установите шрифт Times New Roman, начертание полужирное, размер 12 пт. Аналогичные операции проделайте для ячейки A1.

7. Измените ширину столбцов, в которые не поместились введенные данные. Для этого можно перетащить границы между строками и столбцами или навести указатель мыши на границу между заголовками столбцов, дважды щелкнуть основной кнопкой мыши. Для более точной настройки надо выбрать команду *Строка (Столбец)* из меню *Формат* и активизировать подходящую команду из открывающегося меню.

8. Присвойте каждому студенту свой порядковый номер (не менее 10 студентов), используя маркер заполнения. Для этого:

сделайте текущей первую ячейку столбца “№ п/п” и введите в нее цифру 1;

затем заполните цифрой 2 следующую ячейку этого столбца;

выделите блок, состоящий из двух заполненных ячеек;

установите указатель мыши на правый нижний угол выделенного блока. Указатель мыши станет черным крестиком – это *маркер заполнения*. Перетащите маркер заполнения при нажатой правой кнопке мыши вниз или выберите команду *Правка—> Заполнить—>Прогрессия*.

9. Заполните столбец “Фамилия экзаменатора”. Воспользуйтесь методом автозавершения, который состоит в том, что Excel “угадывает” слово, которое собирается вводить пользователь, или заполните ячейки с помощью *маркера заполнения*. Для включения *Автозавершения* надо в меню сервис выполнить команду *Параметры*, открыть вкладку *Правка* и установить флажок *Автозавершение значений ячеек*.

10. Заполните 2-ой и 3-ий столбцы таблицы данными для своей группы.

- Обрамите таблицу: *Панель инструментов*—> кнопка *Обрамление (Граница)*.
- Скопируйте таблицу на другой рабочий лист при помощи буфера обмена. Для этого следует:
 - выделить таблицу или диапазон ячеек;
 - правой клавишей мыши вызвать *контекстное меню*;
 - выполнить команду *Копировать*; затем перейти на другой лист;
 - установить курсор в первую ячейку предполагаемой таблицы;
 - выполнить команду *Вставить* из контекстного меню.
- Добавьте в новую таблицу одну строку и один столбец. Для этого нужно:
 - выделить диапазон ячеек по столбцу;
 - щелкнуть правой кнопкой мыши и в открывшемся контекстном меню выбрать команду *Добавить ячейки*;
 - то же самое повторить для строки.
- Внесите в таблицу ряд изменений:
 - очистите колонку с фамилией экзаменатора;
 - озаглавьте эту колонку *“Подпись экзаменатора”*.
- Отсортируйте в новой таблице столбцы 2 и 3 по возрастанию – *Данные* —> *Сортировка* или на Стандартной панели инструментов - кнопка *Сортировать по возрастанию*.

Построение диаграмм

Задание. На основе данных, приведенных в табл. 1.2, постройте несколько типов диаграмм, наглядно показывающих итоги сессии.

Таблица 1.2

Средний балл по группе				
Группа	Информатика	Математический анализ	История	Экономика
3- 8530	4,2	3,8	4,5	4,3
3- 2330	4,0	4,4	4,4	4,2
3- 3530	3,9	4,0	4,0	3,9
3- 6530	4,3	4,4	4,4	4,1
3-2231	3,8	4,0	4,0	3,9
3-2232	3,3	3,9	3,9	3,6
3-11230	4,5	4,8	4,8	3,9

Методика выполнения работы

- На листе 1 создайте таблицу “Сведения о результатах сдачи сессии в представительстве ТПУ”, внесите в нее данные.
 - Постройте диаграмму для всех групп и всех предметов на отдельном листе типа *График* (см. табл. 1.2). Для этого следует:
 - выделить всю таблицу;
 - выполнить команду меню *Вставка* → *Диаграмма* или воспользоваться кнопкой *Мастер диаграмм* на стандартной панели инструментов.
 - На третьем шаге построения диаграммы внесите название диаграммы, обозначения осей, добавьте легенду (рис. 1.2).
 - Постройте диаграммы и сравните результаты сдачи по отдельным предметам (например, по предмету “История” см. рис. 1.3). Для этого следует:
 - выделить столбцы “Группа”, “Информатика” и построить диаграмму по этому предмету;
 - выделить столбец “Группа” и, удерживая клавишу *Ctrl*, выделить столбец “Математический анализ” и построить диаграмму по этому предмету.
- Аналогично строятся диаграммы и для остальных предметов, столбцы которых не граничат со столбцом “Группа”.
- Измените результаты сдачи сессии и проверьте, как это отразилось на построенных диаграммах.

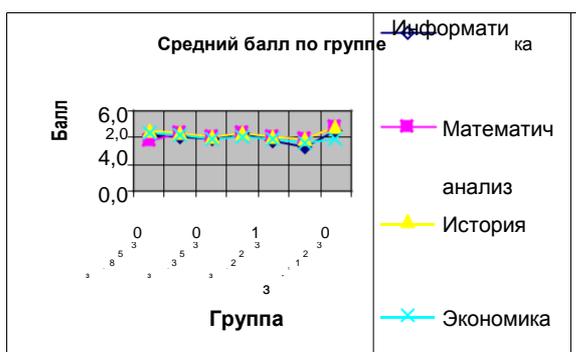


Рис. 1.2. Средний балл по группам для всех предметов

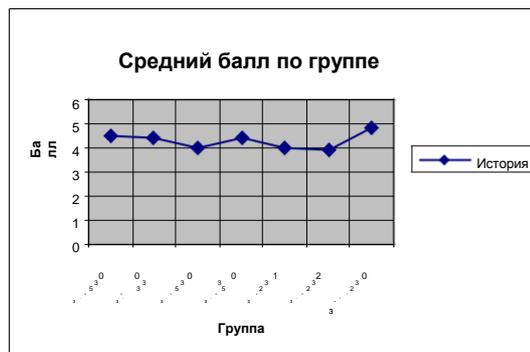


Рис. 1.3. Средний балл по группам по предмету “История”

- Отчет о работе представьте в виде диаграмм на отдельных листах рабочей книги.

Формулы в Excel

Цель работы: создание и использование простых формул в Excel.

Задание № 1

Компьютерная фирма имеет следующие результаты своей торговой деятельности за отчетный период (табл.1.3).

Таблица 1.3

A	B	C	D
Наименование продукции	Цена за ед., р	Продано, шт.	Выручка от продажи, р.
1.			
2.			
3. Модем	1460	10	
4. Принтер	2500	15	
5. Монитор, 17"	5750	20	
6. Компьютер	19899	25	
7. Цифровая фотокамера	10900	4	
8. Переносной накопитель, 128 Mb	1535	6	
9. Сканер	3050	7	
10. Жесткий диск, 80 Gb	2840	13	
11. Ноутбук	51470	1	
12. CD-ROM	745	4	
13. CD-ReWriter	1550	6	
14. Итого, сумма выручки , р.			

Используя возможности Excel, найти сумму выручки от продаж по каждому виду продукции и общую суммы выручки.

Методика выполнения работы

■ Для того чтобы иметь возможность вводить в ячейки информацию в несколько строк, необходимо выполнить *Формат* → *Ячейка*, затем выбрать закладку *Выравнивание* и в пункте *Отображение* выбрать *Переносить по словам*.

■ В столбец **A**, начиная с ячейки **A1** и до **A13** ввести наименование продукции, затем ввести информацию в столбцы **B** и **C** (табл.3.3).

■ В ячейку **D3** ввести формулу расчета выручки в следующей последовательности:

сделать эту ячейку активной;

набрать знак '=';

щелкнуть на ячейку

B3; набрать знак '*';

щелкнуть на ячейку **C3**.

результате в ячейку **D3** будет записана формула: **=B3*C3**. После нажатия <Enter> в этой ячейке появится результат расчета по этой формуле.

■ Теперь методом *Автозаполнения* получите аналогичные формулы для остальных видов продукции. Для этого установите указатель мыши на правый нижний угол ячейки **D3**. Указатель мыши станет черным крестиком – это *маркер заполнения*. Перетащите маркер заполнения при нажатой правой кнопке мыши вниз до ячейки **D13**. После отпускания клавиши мыши выручка будет подсчитана для всех видов продукции фирмы.

■ Подсчитайте сумму выручки от продажи всех видов товаров. Щелкните мышкой на ячейку **D14** и нажмите кнопку *Автосумма* на стандартной панели инструментов (знак Σ) или воспользуйтесь кнопкой *Вставка функции*, расположенной также на стандартной панели. В окне *Мастер функций* следует выбрать *СУММ* из категории *Математические*.

■ результате Вы должны получить следующие результаты расчета:

D3→ 14600 **D4**→37500 ... **D13**→ 9300 **D14**→ 839405

Задание № 2

■ Изучите создание и использование простых формул, используя тематику финансового и банковского менеджмента.

■ Сопоставьте доходность акции по уровню дивидендов за 2019 г. по отдельным эмитентам. Исходные данные задачи представлены в табл. 1.4.

Таблица 1.4

Эмитент	Номинал акции р.	Цена продажи р.	Дивиденды, объявленные в расчете на год		Доходность акций по дивидендам	
	<i>NA</i>	<i>CP</i>	<i>% Div</i>	<i>DivR</i>	К номиналу <i>DN</i>	Фактическая <i>DF</i>
Сибирьгазбанк	10000	17780	400			
Инкомбанк	10000	22900	400			
Сургутнефтегаз банк	5000	5600	320			
Нефтехимбанк	1000	2015	653			
Сбербанк	1000	2482	736			
КБ Аккобанк	1000	1000	325			
СКВ банк	50000	27050	360			
Промстройбанк	1000	1200	1535			

NA – номинал акции; *CP*– цена продажи; *Div* – дивиденды в расчете на год.

■ Визуально проанализируйте полученные результаты.

Методика выполнения работы

Создайте в Excel табл. 1.4., введите в нее исходные данные.

- В соответствующие столбцы введите формулы для расчета выходных показателей:

$$DivR(i) = NA(i) * Div(i);$$

$$DN(i) = DivR(i) / NA(i);$$

$$DF(i) = DivR(i) / CP(i),$$

где $i = [1, n]$, n – число рассматриваемых эмитентов.

Создайте табл. 1.5.

Таблица 1.5

Расчетная величина	Значение
Средняя цена продажи акций	
Максимальная цена продажи акций	
Минимальная цена продажи акций	
Максимальная фактическая доходность акций	
Минимальная фактическая доходность акций	
Средняя фактическая доходность акций	

- На основании исходного документа “Доходность акций по отдельным дивидендам” рассчитайте следующие значения:

а) **среднюю цену продажи акций по всем эмитентам** – щелкнуть на соответствующую ячейку в столбце “Значение” табл.3.5, затем выполнить

Вставка → *Функция* → категория *Статистические* → функция = СРЗНАЧ. Убрать появившееся диалоговое окно *Аргументы функции* с табл. 3.4 (если оно ее закрывает). Выделить ячейки столбца “Цена продажи” со значениями исходных данных в табл. 3.4. В строке *Число 1* окна *Аргументы функции* появятся адреса начала и конца данных этого столбца, разделенные двоеточием. Щелкните на кнопке <ОК>;

б) **максимальную цену продажи акций по всем эмитентам** – щелкнуть на соответствующую ячейку в столбце “Значение” табл.3.5, затем выполнить

Вставка → *Функция* → категория *Статистические* → функция = МАКС. Выделить ячейки столбца “Цена продажи” со значениями исходных данных в табл. 3.4. <ОК>;

с) **минимальную цену продажи акций** - щелкнуть на соответствующую ячейку в столбце “Значение” табл.3.5, затем выполнить

Вставка → *Функция* → категория *Статистические* → функция = МИН. Выделить ячейки столбца “Цена продажи” со значениями исходных данных в табл. 3.4. <ОК>;

- **максимальную фактическую доходность акций по уровню дивидендов** - щелкнуть на соответствующую ячейку в столбце “Значение” табл.3.5, затем выполнить

Вставка → *Функция* → категория *Статистические* → функция = МАКС. Выделить ячейки столбца “Фактическая *DF*” со значениями исходных данных в табл. 3.4. <ОК>.

минимальную фактическую доходность акций по уровню дивидендов – щелкнуть на соответствующую ячейку в столбце “Значение” табл.3.5, затем выполнить

Вставка → *Функция* → категория *Статистические* → функция = МИН. Выделить ячейки столбца “Фактическая *DF*” со значениями исходных данных в табл. 3.4. <ОК>.

среднюю фактическую доходность акций по уровню дивидендов – щелкнуть на соответствующую ячейку в столбце “Значение” табл.3.5, затем выполнить *Вставка* → *Функция* → категория *Статистические* → функция ■ СРЗНАЧ. Выделить ячейки столбца “Фактическая *DF*” со значениями исходных данных в табл. 3.4. <ОК>.

В исходной таблице отсортируйте записи в порядке возрастания фактической доходности по дивидендам (выделите столбец “Фактическая *DF*”, выполните команду *Сортировка* меню *Данные*).

Выполните фильтрацию таблицы, выбрав из нее только тех эмитентов, фактическая доходность которых больше средней по таблице. Алгоритм фильтрации следующий:

выделить данные таблицы с прилегающей одной строкой заголовка “Фактическая *DF*”;

выполнить команду из меню *Данные* : *Фильтр* → *Автофильтр*;

в заголовке столбца “Фактическая доходность” нажать кнопку раскрывающегося списка и выбрать *Условие*;

в окне пользовательского автофильтра задать условие > “среднее значение” (“среднее значение” взять из последней строки табл.3.5).

7. Результаты фильтрации поместите на новый рабочий лист, включив в него следующие графы:

эмитент;

номинал акции;

цена продажи;

доходность по дивидендам фактическая.

■ Постройте на отдельном рабочем листе Excel круговую диаграмму, отражающую фактическую доходность по дивидендам каждого эмитента в виде соответствующего сектора (выделите столбцы “Эмитент” и “Фактическая доходность”, выполнить команду меню *Вставка* → *Диаграмма*). На графике показать значения доходности, вывести легенду и название графика “Анализ фактической доходности акций по уровню дивидендов”.

■ Постройте на новом рабочем листе Excel смешанную диаграмму, в которой представьте в виде гистограмм значения номиналов и цены продажи акций каждого эмитента, а их фактическую доходность покажите в виде линейного графика на той же диаграмме. Выведите легенду и название графика “Анализ доходности акций различных эмитентов”. Алгоритм построения

смешанного графика следующий:

выделить столбцы “Эмитент”, “Номинал акции” и “Цена продажи”;

выполнить команду меню *Вставка* —> *Диаграмма* —> тип диаграммы *Гистограмма*;

для добавления линейного графика “*Фактическая доходность по дивидендам*” правой клавишей мыши активизировать меню *Диаграмма* —> *Исходные данные* —> во вкладке *Ряд*, выбрать кнопку <Добавить>, в поле *Имя* ввести название ряда “Доходность”, в поле *Значения* ввести числовой интервал, соответствующий фактической доходности по дивидендам;

на полученной диаграмме курсор мыши установить на столбец, соответствующий значению “Доходность”, правой клавишей мыши активизировать контекстное меню, выбрать команду *Тип диаграммы*, где выбрать тип диаграммы — *График*.

Сортировка данных в списке

Задание

- Выполнить сортировку данных табл. 3.6 по возрастанию кода предмета, даты проведения занятия, номера группы.
- Выполнить сортировку данных табл. 3.6 по возрастанию, используя сочетания признаков: код предмета и дату проведения занятия; код предмета и номер группы; номер группы и дату проведения занятия, а также сочетание всех трех признаков.

Методика выполнения работы

- Создайте новую рабочую книгу (меню *Файл* команда *Создать*) и сохраните ее под именем SORT.XLS в рабочем каталоге (меню *Файл* команда *Сохранить как*).
- Сформируйте таблицу результатов занятий.

Таблица 3.6

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>
1	№ группы	№ зачетной книжки	Код предмета	Табл. № препод.	Вид занятия	Дата	Оценка
2	3-3230	3-3230/03	П1	A1	Практика	26.05.99	3
3	3-3230	3-3230/12	П2	A2	Лекция	26.05.99	4
4	3-3230	3-3230/06	П1	A1	Лекция	11.06.99	4
5	3-3330	3-3230/08	П1	A2	Лекция	11.06.99	5
6	3-3330	3-3230/18	П2	A1	Практика	16.05.99	2
7	3-6230	3-6230/03	П2	A3	Лекция	20.05.99	3
8	3-3230	3-3230/09	П1	A1	Лекция	16.05.99	3
9	3-3230	3-3230/18	П1	A3	Лекция	16.05.99	4
10	3-3330	3-3330/03	П1	A2	Лекция	26.05.99	4
11	3-3531	3-3531/02	П2	A1	Лекция	11.06.99	2
12	3-3532	3-3532/03	П1	A2	Практика	20.05.99	5
13	3-3230	3-3230/20	П2	A1	Лекция	26.05.99	5

Отформатируйте шапку таблицы следующим образом:

шрифт Times New Roman;

размер шрифта 12 пт., курсив;

выравнивание по горизонтали — *По значению*;

выравнивание по вертикали — *По верхнему краю*;

установите ключ *“Переносить по словам”* (выделить соответствующие ячейки и выполнить команду *Формат —> Ячейки*).

Выполните сортировку по столбцу *“Код предмета”*, расположив коды предметов по возрастанию. Для этого нужно:

выделить таблицу с одной строкой заголовка;

выполнить команду меню *Данные —> Сортировка*;

в окне *Сортировка диапазона* в строке *Сортировать по “коду предмета”*.

5. Результат сортировки скопируйте на Лист 2:

выделите всю таблицу, выполнить команду *Правка —> Копировать*;

затем на Листе 2 установите курсор в ячейку *A1* и выполните команду *Правка —> Вставить*.

6. Переименуйте Лист 2, дав ему имя – *Сортировка*:

указатель мыши установите на ярлычке Лист 2;

правой клавишей мыши вызовите контекстное меню;

выполните команду *Переименовать*.

■ Выполните сортировку по столбцу *“Дата”*, расположив данные по возрастанию. Для этого следует установить курсор в любую ячейку поля *“Дата”* и ввести команду *Сортировка* из меню *Данные*, при этом должна выделиться вся область списка, а в окне *Сортировка Диапазона* в строке *Сортировать по* – столбец *G*. Если этого не произошло, то предварительно выделите весь список, а затем выполните указанную команду.

■ Выполните сортировку по сочетанию признаков *“Дата”*, *“№ группы”*, *“Код предмета”*. Для этого следует выделить всю таблицу и в диалоговом окне *Сортировка* установить:

в строке *Сортировать по* — поле *“Дата”* по возрастанию;

в строке *Затем* — поле *“№ группы”*, по возрастанию;

в следующей строке *Затем* — поле *“Код предмета”* по возрастанию; установите флажок *Строка меток столбцов*.

Результат сортировки скопировать на Лист 3 и переименовать его в *Сортировка 2*.

Фильтрация записей

Цель работы: ознакомиться со способом фильтрации записей списка, автофильтрации, работой с формой данных.

Методика выполнения работы

- Создайте новую рабочую книгу с названием “Фильтрация”.
- Скопируйте в новую рабочую книгу таблицу, созданную в работе № 4 (см. табл. 3.6).
- Переименуйте Лист1, присвоив ему имя “Автофильтр №1”.
- Чтобы применить *Автофильтрацию*, установите курсор в область шапки таблицы и выполните команду *Данные*—>*Фильтр*—>*Автофильтр*.
- Сформируйте условия отбора: для преподавателя А1 выбрать сведения о сдаче экзамена на положительную оценку, вид занятий – Лекция. Для этого выполните следующие действия:
 - в столбце *Таб № препод.* нажмите кнопку *Фильтр*, из списка условий отбора выберите А1;
 - в столбце *Оценка* нажмите кнопку *Фильтр*, из списка условий отбора выберите *Условие* и в диалоговом окне сформируйте условие отбора >2;
 - в столбце *Вид занятий* нажмите кнопку *Фильтр*, из списка условий отбора выберите *Лекция*.
- Результат фильтрации скопируйте на новый лист, присвоив ему имя “Автофильтр №2”.
- На листе “Автофильтр №1” результат автофильтрации отмените, установив указатель мыши в область списка и выполнив команду *Данные* —> *Фильтр* —> *Автофильтр*.
- Сформулируйте выборку: для группы 3-3230 получите сведения о сдаче экзамена по предмету П1 на оценки 3 и 4.
- Результат сохраните на новом листе, присвоив ему имя “Автофильтр №3”.
- Скопируйте исходную таблицу на новый рабочий лист, переименовав его в *Форма данных*.
- Установите курсор в область шапки таблицы и выполните команду *Данные* —>*Форма*.
- В окне *Форма данных* просмотрите записи списка и внесите не-обходимые изменения по своему усмотрению с помощью кнопок <*Предыдущая*> и <*Следующая*>.
- С помощью кнопки <*Создать*> добавьте новые записи.
- В окне *Форма данных* сформируйте условия отбора записей. Для этого нажмите кнопку <*Критерии*>, название которой поменяется на <*Правка*>. В пустых строках имен полей списка введите критерии:
 - в строку *Табл. № препод.* введите А1;

в строку *вид занятия* введите *Лекция*;

в строку *оценка* введите условие > 2 .

- Просмотрите отобранные записи нажатием на кнопку *<Предыдущая>* или *<Следующая>*.
- По аналогии сформулируйте условия отбора записей, указанные в п. 8.

Использование логических функций

Задание № 1

- Подсчитайте количество отличных, хороших и т. д. оценок на основании зачетной ведомости, представленной в табл. 3.7.
- Произведите расчет, используя операцию “Присвоение имени блоку ячеек”.

Методика выполнения работы

- На новом листе рабочей книги создайте таблицу по образцу табл.3.7.
- Заполните данными столбцы *A, B, C, D*.

Таблица 3.7

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>
1	№ п/п	Фам., имя, отчество	№ зач. книжки	Оценка	Кол-во 5	Кол-во 4	Кол-во 3	Кол-во 2	Неявка
2	1	Демидов М.И.	3-3230/04	5					
3	2	Иванов И. П.	3-3230/05	4					
4	3	Кукушкин В. Л.	3-3230/07	3					
5	4	Орлов А. П.	3-3230/11	4					
6	5	Петров К.Н.	3-3230/13	5					
7	6	Сидоров В.О.	3-3230/15	2					
8	7	Фролов В А.	3-3230/18	0					

- В столбцы *E, F, G, H, I* введите формулы, для этого:

установите курсор в первую ячейку столбца количества отличных оценок (*E2*) и выполните *Вставка* → *Функция* → категория *Логические* → функция = ЕСЛИ, убрать появившееся диалоговое окно *Аргументы функции* с табл. 3.7 (если оно ее закрывает);

в диалоговом окне *Аргументы функции* установите курсор в поле *Лог. выражение* и щелкните мышью в рабочей области Excel на ячейке *D2*;

появится адрес ячейки *D2* и с клавиатуры введите $\leq 5 >$, т.е. сформируется логическое выражение $D2=5$;

в поле *Значение_если_истина* введите $\langle 1 \rangle$;

в поле *Значение_если_ложь* введите $\langle 0 \rangle$;

щелкните на кнопке $\langle OK \rangle$.

Методом протягивания скопируйте формулу по столбцу *E* - “Кол-во 5”.

■ С помощью *Мастера функций* аналогичным способом введите формулы в столбцы “Кол-во 4“, “Кол-во 3” и т. д., изменяя соответственно значение поля *Логическое выражение* на $D2 = 4$, $D2 = 3$, $D2 = 2$. Для подсчета количества неявившихся на экзамен необходимо задавать логическое выражение $D2=0$;

■ Чтобы подсчитать сумму всех пятерок, четверок и т. д. и результаты представить в виде отдельной таблицы, нужно по каждому столбцу “Кол-во оценок” задать имена блокам соответствующих ячеек. Для этого выполните следующие действия:

выделите блок ячеек **E2:E8** столбца “Кол-во 5”;

выполните команду меню *Вставка* → *Имя* → *Присвоить*;

в диалоговом окне *Присвоение имени* в строке *Имя* введите слово *Отлично* и щелкните на кнопке *Добавить* и затем *<ОК>*;

далее выделите ячейки **F2:F8** столбца “Кол-во 4” и выполните команду *Вставка* → *Имя* → *Присвоить*;

в диалоговом окне *Присвоение имени* в строке *Имя* введите слово *Хорошо* и щелкните на кнопке *<Добавить>* и затем *<ОК>*;

аналогичные действия выполните с остальными столбцами табл. 3.7, создав имена блоков ячеек *Удовлетворительно*, *Неудовлетворительно*, *Неявка*.

■ Создайте таблицу *Итоги сессии* (табл.3.8).

Таблица 3.8

ИТОГИ СЕССИИ	
Количество отличных оценок	
Количество хороших оценок	
Количество удовлетворительных оценок	
Количество неудовлетворительных оценок	
Неявки	
ИТОГО	

Введите формулу подсчета количества полученных оценок определенного вида:

установите курсор в ячейку подсчета количества отличных оценок и выполните *Вставка* → *Функция* → категория *Математические* → функция

■ СУММ;

щелкните на кнопке *<ОК>*;

в диалоговом окне *Аргументы функции* установите курсор в строку *Число1* и выполните *Вставка* → *Имя* → *Вставить*;

в диалоговом окне *Вставка имени* выберите имя блока ячеек *Отлично*

■ щелкните на кнопке *<ОК>*; повторите аналогичные действия для подсчета количества других оценок.

- Подсчитайте ИТОГО – количество всех полученных оценок, используя кнопку *Автосумма* на стандартной панели инструментов.

Кнопка *Автосумма* – это символ Σ на стандартной панели инструментов. Если его нет, необходимо выполнить *Вид* → *Панели инструментов* → поставить флажок на кнопку *Стандартная*.

Для подсчета ИТОГО:

- щелкнуть на ячейку, где должен быть результат ИТОГО; щелкнуть на кнопку *Автосумма*; выделить группу ячеек с количеством всех оценок;

<Enter>.

Задание № 2. Определить, в какой из заданных интервалов попадает зарплата каждого сотрудника НИИ, представленная в табл. 3.9.

Методика выполнения работы

Создайте новую рабочую книгу.

Создайте таблицу из восьми столбцов, в которой содержатся сведения о пяти сотрудниках ОАО: № п/п, Ф.И.О., ежемесячная зарплата (табл. 3.9).

Создайте таблицу, содержащую четыре интервала числовых значений зарплат: 3000 — 4000, 4000 — 5000, 5000 — 6000, 6000—8000 (см. табл. 3.10).

Чтобы определить, попадает ли значение зарплаты из столбца *С* в заданный интервал, нужно использовать логическую функцию **ЕСЛИ** с заданием сложного условия **И**.

Для этого необходимо выполнить следующее (для интервала 3000 – 4000): установите курсор в ячейку **D2**;

выполните *Вставка* → *Функция* → категория *Логические* → функция

- **ЕСЛИ**;

щелкните на кнопке <ОК>;

в открывшемся окне *Аргументы функции* в поле *Лог. выражение* введите следующее логическое выражение: **И(C2>\$A\$10;C2<=\$B\$10)**;

в поле *Значение_если_истина* введите <1>;

в поле *Значение_если_ложь* введите <0>;

щелкните на кнопке <ОК>.

Для остальных интервалов аналогично, только будут другие номера ячеек **A** и **B** – **A11, B11** и т. д. (см. табл.3.10).

Пример выполнения практической работы.

Таблица 3.9

	A	B	C	D	E	F	G	H
1	№п/п	Ф.И.О.	Зарплата	1 ин.	2 ин.	3 ин.	4 ин.	Проверка
2	1	Кузнецов	7896	0	0	0	1	1
3	2	Свиридов	5990	0	0	1	0	1
4	3	Молотов	4098	0	1	0	0	1
5	4	Иванов	3980	1	0	0	0	1
6	5	Петров	4346	0	1	0	0	1
7	ИТОГО			1	2	1	1	5

Таблица 3.10

		<i>A</i>	<i>B</i>
		Интервалы	
10	1 ин.	3000	4000
11	2 ин.	4000	5000
12	3 ин.	5000	6000
13	4 ин.	6000	8000

В ячейке **D2** находится формула **ЕСЛИ(И(C2>\$A\$10;C2<=\$B\$10);1;0)**.
Эту формулу операцией *автозаполнения* скопировать по столбцу **D** от **D2** до **D6** для остальных сотрудников НИИ.

Подобные формулы ввести в столбцы **E, F, G**.

Для подсчета числа попаданий в каждый интервал выполните следующие действия:

выделите блок **D2:D6**;

нажмите кнопку *Автосумма* на *Стандартной панели* инструментов;

повторите это действие для каждого столбца.

■ Значения столбца *Проверка* получите, используя операцию *Автосумма* для значений блоков строк **D2:G2, D3:G3** и т. д.

■ Значение ячейки *Итого* столбца *Проверка* должно совпадать с количеством сотрудников.

Задания для самостоятельной работы

Задание № 1

Продукцией городского молочного завода являются молоко, кефир и сметана. На производство 1 т молока, кефира и сметаны требуется соответственно 1010, 1020 и 9450 кг молока.

Прибыль от реализации 1 т молока, кефира и сметаны соответственно равна 300, 220 и 1360 р. Было изготовлено молока 123 т, кефира 342 т, сметаны

■ т.

Требуется:

при помощи электронной таблицы рассчитать:

прибыль от реализации каждого вида продукции, общую прибыль;

долю (в %) прибыльности каждого вида продукции от общей суммы;

расход молока (сырья);

построить диаграмму по расходу сырья для каждого вида продукции.

Задание № 2.

На книжную базу поступили 3 наименования книг: словари, книги по кулинарии и пособия по вязанию. Они были распределены по трем магазинам: “Книжный мир”, “Дом книги” и “Глобус”.

■ “Книжный мир” поступило словарей – 10400 экземпляров, кулинарных книг – 23650 экземпляров, пособий по вязанию – 1500 экземпляров;

■ “Дом книги” – 10300 словарей, 22950 кулинарных книг и 1990 пособий по вязанию;

“Глобус” – соответственно 9100, 23320 и 2500 экземпляров.

■ первом магазине было продано словарей – 8945 экземпляров, кулинарных книг – 19865 экземпляров, пособий по вязанию – 873 экземпляра.

Во втором магазине было продано словарей – 9300 экземпляров, кулинарных книг – 21900 экземпляров, пособий по вязанию – 1020 экземпляра.

третьем магазине соответственно было продано 8530, 18100 и 2010 экземпляров.

Требуется:

■ при помощи электронной таблицы рассчитать:

общее количество книг каждого наименования, поступивших на книжную базу;

процент продажи каждого наименования книг в каждом магазине;

количество книг, оставшихся после реализации;

■ построить диаграмму по распределению книг в магазинах.

Задание № 3

Производственная единица изготавливает изделия трех видов: П1, П2 и П3. Затраты на изготовление единицы продукции П1, П2 и П3 составляют 7, 15 и 10 (\$) соответственно.

Прибыль от реализации одного изделия данного вида соответственно равна 20, 16 и 25 (\$). План производства изделий П1—200482 шт., П2—43292 шт., П3—1463012 шт. В январе было изготовлено П1— 135672 шт., П2— 60712 шт., П3— 1456732 шт.

Требуется:

а) при помощи электронной таблицы рассчитать в долларах (курс доллара – величина изменяющаяся):

плановые затраты на производство;

прибыль от реализации каждого вида изделий;

прибыль, полученную предприятием в январе;

процент выполнения плана в январе по каждому виду

изделия. б) построить диаграмму по прибыли каждого вида изделия.

Задание № 4

Часовой завод изготовил в январе часы вида А – 150 шт., вида В – 230 шт., вида С – 180 шт. В феврале производство продукции выросло: вида А – на 5 %, вида В – на 3 %, С – на 2 %. В марте рост составил соответственно 1,5; 1,6 и 2 %. Затраты на изготовление каждого вида часов составляют А – 85 р., В – 73 р., С – 84 р. Продажная стоимость каждого вида изделий составляет соответственно 120 р., 100 р. и 110 р.

Требуется:

- а) при помощи электронной таблицы рассчитать:
 - какое количество часов изготовлено в каждый месяц;
 - прибыль от реализации каждого вида изделий;
 - ежемесячные затраты на производство каждого вида изделий;
- б) построить диаграмму по прибыли каждого вида изделия.

Задание № 5

На предприятии работники имеют следующие оклады: начальник отдела – 9000 р., инженер 1 кат. – 7000 р., инженер – 5000 р., техник – 3000 р., лаборант – 2000 р.

Все работники получают надбавку 10 % от оклада за вредный характер работы. Все работники получают 50 % премии в том месяце, когда выполняется план.

При невыполнении плана из зарплаты вычитают 10 % от начислений. Со всех работников удерживают 12 % подоходный налог, 1 % – профсоюзный взнос. Все удержания производятся от начислений.

Требуется:

- а) при помощи электронной таблицы рассчитать суммы к получению каждой категории работников по месяцам;
- б) построить две диаграммы, отражающие отношение зарплаты всех работников в различные месяцы.

Контрольные вопросы и задания

- Каково назначение электронной таблицы?
- Как называется документ в программе Excel? Из чего он состоит?
- Каковы особенности типового интерфейса табличных процессоров?
- Какие типы данных могут содержать электронные таблицы?
- Какие данные называют зависимыми, а какие независимыми?
- По какому признаку программа определяет, что введенные данные являются не значением, а формулой?
- Что в Excel используется в формулах в качестве операндов?
- Что такое формула в электронной таблице и ее типы? Приведите

примеры.

Что такое функция в электронной таблице и ее типы? Приведите примеры.

Поясните, для чего используются абсолютные и относительные адреса ячеек.

Что такое *автозаполнение*?

Каков приоритет выполнения операций в арифметических формулах Excel?

Как можно “размножить” содержимое ячейки?

Как посмотреть и отредактировать формулу, содержащуюся в ячейке?

Какой тип адресации используется в Excel по умолчанию?

В чем состоит удобство применения относительной и абсолютной адресации при заполнении формул?

Что такое диапазон, как его выделить?

Как защитить содержимое ячеек электронной таблицы от несанкционированного доступа и внести изменения?

Укажите, какие Вы знаете типы диаграмм, используемых для интерпретации данных электронной таблицы. Поясните, когда следует или не следует использовать каждый из них.

Какие способы объединения нескольких исходных электронных таблиц ■ одну Вам известны?

Как использовать электронную таблицу для моделирования по типу решения задачи “Что будет, если...”?

СОЗДАНИЕ МАКРОСОВ В СРЕДЕ *Excel*

Цель работы: получить начальное представление о работе с макросами *Excel* и изучить основные возможности меню.

Термином макрос обычно обозначают файл, хранящий последовательность действий, заданных пользователем системы. Каждый макрос должен иметь собственное имя. С помощью макроса можно автоматизировать типовые технологические этапы при работе с системой. После вызова макроса записанная в нем последовательность действий (команд) будет автоматически исполнена. По своей сути макрос представляет собой программу и может быть создан автоматически в [специальном режиме](#) работы программной системы (в том числе и *Excel*) или как результат программирования в терминах команд системы. Очевидно, что созданный любым способом макрос может быть подвергнут редактированию с целью изменения его возможностей или устранения ошибок.

Первая строка представляет собой заголовок созданной процедуры Проверка. Параметры процедуры отсутствуют. Символом ' обозначен комментарий. Первая рабочая строка программы вызывает объект Application (приложение) с методом Run (выполнить), вызывающим на выполнение объект Range (диапазон) с параметрами начальной установки рабочего листа. Далее производится выделение ячейки A1 текущего рабочего листа и запись в активную ячейку в формате R1C1 текста “Проверка”. Отметим, что в *Excel* существует два формата (A1 и R1C1) и два способа адресации (относительная и абсолютная) ячеек памяти. В формате A1 признаком абсолютной адресации является символ “\$”. В формате R1C1 абсолютный адрес задается индексами рабочей строки и столбца, а при необходимости использования относительной адресации в квадратных скобках указывается относительное смещение со знаком по отношению к текущей рабочей ячейке. Последующие действия программы соответствуют выполненной последовательности при программировании макроса. Как следует из текста, константы и формулы заносятся с использованием метода FormulaR1C1 объекта ActiveCell.

Модифицируйте созданную программу. Введите в ее текст описание двух дополнительных переменных:

```
Dim b, c As Integer
```

Задайте в тексте программы оператором присваивания им некоторые начальные значения и модифицируйте строчки:

```
Range("A2").Select
```

```
ActiveCell.Formula = b
```

```
Range("B2").Select
```

```
ActiveCell.FormulaR1C1 = c
```

Запустите макрос на выполнение и убедитесь, что он выполняется правильно.

Введите в текст программы стандартное сообщение:

```
MsgBox "Первый сомножитель", vbOKOnly, "Проверка"
```

Убедитесь, что программа выполняется правильно.

Предварительно описав переменную d как строку символов, введите в текст программы оператор [ввода данных](#):

```
Dim d As String
```

```
d = InputBox("Первый сомножитель", "Проверка")
```

Преобразуйте значение d из символьного вида к числовому с помощью функции

```
b = Val(d)
```

и убедитесь, что содержимое ячейки может быть задано с клавиатуры в [диалоговом режиме](#).

Ниже представлен пример созданной в процессе модернизации макроса программы на языке VBA:

```
Sub Проверка()
```

```
,
```

```
' Проверка Макрос
```

```
' Макрос записан 16.07.2001 (stepanov)
```

```
,
```

```
' Сочетание клавиш: Ctrl+z
```

```
,
```

```
Dim b, c As Integer
```

```
Dim d As String
```

```
*****n Range("AUTOSAVE. XLA! mcs02.OnTime")
```

```
Worksheets(1).Activate
```

```
b = 5
```

```
c = 6
```

```
MsgBox "Первый сомножитель", vbOKOnly, "Проверка"
```

```
d = InputBox("Первый сомножитель", "Проверка")
```

```
b = Val(d)
```

```
Range("A1").Select
```

```
ActiveCell. FormulaR1C1 = "Проверка"
```

```
Range("A2").Select
```

```
ActiveCell. Formula = b
```

```
Range("B2").Select
```

```
ActiveCell. FormulaR1C1 = c
```

```
Range("C2").Select
```

```
ActiveCell. FormulaR1C1 = "=RC[-2]*RC[-1]"
```

```
End Sub
```

Цель работы: получить начальное представление о возможностях отладчика VBA.

1. Методические указания

Одной из важнейших проблем, связанных с программированием, является проблема своевременного обнаружения и устранения ошибок, возникающих при создании программ. Человечество подходит к разрешению этой проблемы с разных сторон. Во-первых, создаются методы и средства автоматизации программирования, позволяющие уменьшить вероятность возникновения ошибки как таковой. К их числу относятся технологии структурного программирования и само объектно-ориентированное программирование. Во вторых, совершенствуются сами программные средства и языки программирования, создаются специализированные программы, которые позволяют относительно легко обнаруживать ошибки. Наконец, в третьих, ведутся статистические исследования и выявляются типовые ошибки, которые делают программисты. На основании полученной статистики выдаются рекомендации разработчикам нового программного обеспечения. Настоящая лабораторная работа посвящена изучению методов обнаружения ошибок, реализованному в VBA.

Обычно выделяют три основных вида ошибок, которые приводят к неправильному выполнению программы или делают ее выполнение просто невозможным. Первый вид ошибок – это ошибки, возникающие на этапе компиляции. Основной смысл определяемых на этапе компиляции ошибок – это некорректная запись операторов программы с точки зрения правил языка программирования. Как следствие, компилятор не может создать код и требует внести изменения в программу. Компилятор VBA высвечивает строку программы, которая содержит ошибку, красным цветом и выдает дополнительное диагностическое сообщение.

Ошибки этапа компиляции устраняются программистом с относительно небольшими затратами труда, поскольку их поиск автоматизирован, а для уточнения правил языка программист может легко воспользоваться [справочной литературой](#) или встроенной в компилятор VBA системой помощи. Ее вызов осуществляется при нажатии клавиши F1. С целью минимизации вероятности возникновения [орфографической](#) ошибки при записи класса или метода объекта может быть вызвано специальное контекстное меню вводом команды *Редактирование/Список свойств и методов*. Аналогичное меню может быть вызвано и для списка констант. Быстрый вызов меню можно осуществить и правой клавишей мыши при наборе текста. Наконец, распознанные операторы

языка выделяются цветом, что позволяет уменьшить вероятность ошибки, связанной с неправильным именем переменной. В любом случае ошибки компиляции сопровождаются диагностическим сообщением, из которого, воспользовавшись при необходимости системой помощи, можно установить их причину.

Более сложный класс ошибок – это ошибки, возникающие на этапе выполнения программы. Эти ошибки в том или ином виде связаны с обрабатываемыми данными и, как следствие, не могут быть определены на этапе компиляции, поскольку конкретные значения данных в этот момент неизвестны. При возникновении подобных ошибок на экран выдается диагностическое сообщение с указанием кода ошибки и его кратким описанием. Составляя алгоритм, программист обязан предусмотреть возможность их появления и принять дополнительные меры по их локализации и, если это требуется, перехвату. Список некоторых ошибок этапа выполнения приведен в табл. 2.1.

Наиболее сложным видом ошибок при программировании являются алгоритмические ошибки. Причина таких ошибок двояка – с одной стороны они возникают из-за неправильного составления алгоритма, с другой из-за неправильного кодирования (записи операторов программы не в соответствии с составленным алгоритмом). К сожалению, единственным способом обнаружения алгоритмических ошибок является тестирование. Под тестированием обычно понимают испытание программы при условии подачи на нее заведомо известных данных (теста) и проверки результатов ее работы (они должны быть определены совместно с подготовкой теста). Особенностью тестирования является то обстоятельство, что если тест обнаруживает факт существования алгоритмической ошибки (программа выполняется неверно), то ошибка существует и должна быть устранена. В тоже время, если тест не находит ошибки, то это обстоятельство не является доказательством того, что ошибка отсутствует. Как следствие, созданная программа должна быть подвергнута максимально возможному тестированию. Однако исчерпывающее тестирование программы, как правило, является невозможным из-за чрезвычайно большого числа возможных вариантов данных, в связи с чем приходится использовать методы программирования, уменьшающие вероятность возникновения ошибки, и рассчитывать на искусство программиста. Если в процессе тестирования была обнаружена ошибка, программист должен начать процесс определения конкретных операторов программы, вызвавших появление ошибки, обычно называемый отладкой. Для автоматизации процесса тестирования и отладки созданы специальные программы, которые получили название программ-отладчиков. Подобная программа есть и в составе редактора VBA.

Таблица 2.1.

Ошибки этапа выполнения

Код ошибки	Диагностическое сообщение
5	Приложение не запущено
6	Переполнение
7	Не хватает памяти
9	Выход индекса за границы диапазона
11	Деление на нуль
13	Несоответствие типов данных
18	Произошло прерывание, вызванное пользователем
52	Неправильное имя файла или идентификатора
53	Файл не найден
54	Неправильный режим работы с файлом
55	Файл уже открыт
56	Ошибка ввода-вывода
61	Переполнение диска
68	Устройство не доступно

71	Диск не готов
72	Повреждена поверхность диска
335	Невозможен доступ к системным ресурсам
368	Истек срок действия данного файла
482	Ошибка принтера



Рис. 4. Меню отладчика.

Режим отладки включается из главного меню при активном окне редактора при выполнении команды *Отладка/Шаг с заходом*. Выключение режима обеспечивает выполнение команды *Выполнить/Сброс*. Внешний вид меню показан на рис. 4. Обратите внимание на то, что основные команды отладчика вызываются комбинациями клавиш F8 и F9. Команда *Шаг с заходом* позволяет оператору за оператором выполнить тестируемую программу, включая вызываемые программой функции и процедуры. Команда *Шаг с обходом* исключает пошаговое выполнение вызываемых модулей. Команда *Шаг с выходом* завершает пошаговое выполнение вызванного модуля. Команда *Выполнить до текущей позиции* выполняет программу до оператора, на

котором установлен курсор. Команда *Точка останова* задает и снимает точку останова в тексте программы, причем конкретный оператор предварительно выбирается курсором. Команда *Снять все точки останова* удаляет все установленные в программе точки останова. Выполнение команд *Задать следующую инструкцию* и *Показать следующую инструкцию* позволяет найти следующий выполняемый оператор в окне редактирования.

Текущие значения переменных можно наблюдать задавая их имена в окне контрольных значений (*Отладка/Добавить контрольные значения*, *Отладка/Контрольные значения*, *Вид/Окно локальных переменных*). При выполнении команды *Вид/Проверка* можно задать дополнительный оператор VBA, или изменить значение любой переменной оператором присваивания.

2. Порядок выполнения работы

1. Включите компьютер. Загрузите *Windows*. Загрузите *Excel* и выполните команду *Файл/Открыть*. При стандартной настройке на экране появится окно *Открытие документа*, а в окне *Папка* появится название основной рабочей папки *Мои документы*. Если по каким-то причинам установилась другая папка, нажмите кнопку меню *Папка* и выберете папку *Мои документы*. Ниже будет показано содержимое папки *Мои документы*. Найдите папку с номером вашей группы. Откройте эту папку и найдите файл, соответствующий вашей фамилии, установите на него курсор. Нажмите кнопку *Открыть* и убедитесь, что в верхней левой области экрана после текста *Microsoft Excel* появилось название рабочей папки с вашей фамилией.

2. Перейдите на следующий лист рабочего поля. Если свободные листы отсутствуют, выполните команду *Вставить/Лист*.

3. Последовательностью команд *Сервис/Макрос/Редактор Visual Basic* запустите редактор VBA.

4. Вернитесь к макросу, записанному вами при выполнении предыдущей лабораторной работы.

5. Запустите отладчик и выполните программу по шагам.

6. Введите точки останова и выполните программу с точками останова. Удалите точки останова.

7. Включите окно контрольных значений, введите в него несколько переменных и проконтролируйте в нем изменения значений переменных при пошаговом выполнении программы.

8. Повторите пошаговое выполнение программы, контролируя значения в окне локальных переменных.

9. Воспользуйтесь окном проверки для оперативного изменения текста выполняемой программы, а также для изменения значений переменных.

3. Порядок оформления отчета

Отчетом о лабораторной работе является файл с именем, совпадающим с фамилией студента с результатами работы в папке *Мои документы/номер группы*.

4. Контрольные вопросы

1. Перечислите виды ошибок, возникающих в процессе создания и эксплуатации программного обеспечения.

2. Как диагностируются ошибки, выявляемые компилятором VBA? В чем причина этих ошибок?

3. Какие ошибки могут возникнуть на этапе выполнения программы? В чем причина возникновения этих ошибок?

4. Что такое тест и как выполняется тестирование?

5. Каковы виды пошаговых режимов работы отладчика?

6. Какие существуют способы контроля над значениями переменных?

7. Как можно изменить значения переменных в процессе отладки программы?

СОЗДАНИЕ ФУНКЦИЙ ПОЛЬЗОВАТЕЛЯ

Цель работы: изучить основные приемы создания функций пользователя пакета *Excel*.

1. Методические указания

Библиотека встроенных функций *Excel* содержит большое количество разнообразных функций, ориентированных на самые разнообразные применения. К их числу относятся финансовые, математические, статистические, текстовые, логические функции и ряд других. Для облегчения поиска необходимой функции они объединены по разнообразным категориям, в число которых входит и категория функций *Определенные пользователем*. Функции для этой категории создаются пользователем самостоятельно в частности средствами VBA. Создание подобных функций позволяет использовать в *Excel* все возможности языка программирования высокого уровня. В их числе упомянем введение идентификаторов, возможность задания типов данных, в том числе и определенных пользователем, описание переменных, в том числе и обычных и динамических массивов, операций, встроенных диалоговых окон, использование стандартных операторов ветвления и циклов и введения в текст программы комментариев. Кроме этого имеется возможность создания процедур и использования собственных встроенных функций.

Типы данных. Тип данных определяет множество значений, которое может принимать переменная. Язык VBA поддерживает следующие основные типы данных, приведенные в табл. 3.1. Для описания переменных в тексте программы используется следующий синтаксис:

```
Dim [ WithEvents ] ИмяПеременной [(Индексы)] [ As [ New ] Тип ]
```

Одним оператором Dim можно описывать несколько переменных. В качестве обозначений используются:

WithEvents – ключевое слово, указывающее, что ИмяПеременной является именем объектной переменной, используемым при отклике на события, генерируемые другими приложениями (внешняя ссылка).

Индексы – размерности массивов, задаваемые в формате

[Нижний To] Верхний

По умолчанию нижний индекс считается равным 0.

New – ключевое слово, указывающее возможность неявного создания объекта. Новый экземпляр объекта создается при первой ссылке на него.

Тип – тип переменной в соответствии с табл. 3.1.

Имена переменных в VBA не должны превышать 255 символов и должны начинаться с буквы. Символы верхнего и нижнего регистров инвариантны и системой не различаются. Не допускается использование в имени символа пробела, точки, а также символов %, &, !, #, @, \$.

Для указания типа данных аргументов функций и возвращаемых значений используется инструкция DefТип, где Тип – тип данных в соответствии с табл. 3.1.

Операции. Язык VBA позволяет выполнять стандартный набор операций над данными (табл.3.2.).

Если в операторе используется несколько операций, то порядок их выполнения определяется их приоритетом. Высший приоритет 1 имеют вызов функции и круглые скобки.

Операторы языка. Язык VBA содержит весь базовый набор операторов классического языка Basic с добавлениями, учитывающими потребности объектно–ориентированного программирования. Операторы записываются на отдельных строках и могут не нумероваться. Для размещения нескольких операторов на одной строке между ними необходимо поставить символ двоеточие (:). Этот же символ используется для обозначения меток. Для переноса продолжения оператора на следующую строку используется комбинация символов пробел знак подчеркивания (_). Нельзя разбивать переносом выражения и строки. Допускается не более семи переносов строк одного оператора. В текст программы могут включаться комментарии. Символом комментария является апостроф (') или ключевое слово Rem.

Таблица 3.1.

Типы данных VBA

Тип данных	Размер (байт)	Диапазон значений
Byte (байт)	1	От 0 до 255
Boolean (логический)	2	True или False
Integer (целые)	2	От -32768 до 32767
Long (длинное целое)	4	От - до
Single (плавающее обычной точности)	4	От -3,402823E38 до -1,401298E-45 и от 1,401298E-45 до 3,402823E38
Double (плавающее двойной точности)	8	От -1,E308 до -4,E-324 и от 4,E-324 до 1,E308
Currency (денежный)	8	От -,5808 до ,5808
Decimal (масштабируемое целое)	14	+/- и 28 знаков после запятой. Минимальное отличное от нуля значение имеет вид +/-0,
Date (время и дата)	8	От 1 января 100 г. до 31 декабря 9999 г.
Object (объект)	4	Любой указатель объекта
String (строка переменной длины)	10+ длина строки	От 0 до приблизительно двух миллиардов
String (строка постоянной длины)	Длина строки	От 1 до 65400
Variant (числовые подтипы)	16	От -1,E308 до -4,E-324 и от 4,E-324 до 1,E308
Variant (строковые подтипы)	22+ длина строки	От 0 до приблизительно двух миллиардов

Тип (определяемый пользователем)

Определяется элементами Диапазон каждого элемента определяется его типом данных типа

Таблица 3.2

Операции языка VBA

Операции	Приоритет	Название	Пример	Результат
Арифметические операции			A=11 B=5	
-	3	Смена знака	-A	-11
+	7	Сложение	A+B	16
-	7	Вычитание	A-B	6
*	4	Умножение	A*B	55
/	4	Деление	A/B	2.2
\	5	Целочисленное деление	A\B	2
Mod	6	Остаток от деления по модулю	A Mod B	1
^	2	Возведение в степень	A^B	161015
Операции сравнения			A=11 B=5	
<	8	Меньше	A<B	False
>	8	Больше	A>B	True
<=	8	Меньше и равно	A<=B	False
>=	8	Больше и равно	A>=B	True
<>	8	Не равно	A<>B	True

=	8	Равно	A=B	False
Is		Сравнение со ссылкой на объекты	Dim A, B, C, D, E Set A=D Set B=D Set C=E F=A Is B F=A Is C	True False
Логические операции			A=True B=True C=False D=False	
And	10	Логическое умножение (и)	A And B A And C C And B C And D	True False False False
Or	11	Логическое сложение (или)	A Or B A Or C C Or B C Or D	True True True False
Xor	12	Исключающее или	A Xor B A Xor C C Xor B C Xor D	False True True False
Not	9	Отрицание	E=Not B E=Not D	False True
Imp	14	Импликация	A Imp B A Imp C C Imp B C Imp D	True False True True
Eqv	13	Эквивалентность	A Eqv B A Eqv C C Eqv B C Eqv D	True False False True
Операции со строками			A="abc" B="123"	
&		Сцепление строк	A&B	"abc123"

Оператор присваивания обеспечивает занесение информации в ячейки памяти, связанные с идентификатором и имеет символ равенства (=). Синтаксис оператора имеет вид

[Let] Идентификатор = Выражение

Объектные переменные рассматриваются как указатели на объект. Для записи значения в указатель (ссылки на объектную переменную) используется ключевое слово Set. Синтаксис оператора в этом случае:

Set ОбъектнаяПеременная = [New] ОбъектноеВыражение

или

Set ОбъектнаяПеременная = Nothing

Ключевое слово New используется при создании нового экземпляра класса, а ключевое слово Nothing позволяет освободить системные ресурсы от объекта, который в дальнейшем использоваться не будет.

Условный оператор имеет синтаксис:

If Условие Then [Операторы] [Else Операторы_Else] End If

Если Условие принимает значение True, выполняются операторы, указанные после ключевого слова Then, иначе выполняются операторы после ключевого слова Else. Допускаются вложенные операторы If.

Оператор ветвления имеет вид:

Select Case Выражение

[Case Значение1 [Операторы1]]

[Case ЗначениеN [ОператорыN]]

[Case Else [ОператорыElse]]

End Select

Значение выражения вычисляется каждый раз при выполнении оператора Select. Если вычисленное значение совпадает одной из возможных констант Значение, выполняются соответствующие операторы. Если совпадение не было найдено, выполняются ОператорыElse.

Оператор обработки ошибок позволяет перехватывать прерывания, возникающие в процессе выполнения программы из-за ошибок этапа выполнения. Синтаксис оператора:

On Error GoTo Строка

Начиная с этого момента, при возникновении ошибки управление передается оператору, помеченному как Строка. Модуль, содержащий включенный оператор обработки ошибок, должен содержать оператор Exit (например, Exit Sub, Exit Function, Exit Property), не позволяющий выполниться фрагменту программы обработки ошибок естественным путем. Программа обработки прерывания включается в текст модуля, начинается с первого оператора, помеченного как Строка: и заканчивается оператором Resume Next. Завершение всего модуля оформляется обычным способом (например, End Sub). Оператор On Error Resume Next указывает, что при возникновении ошибки управление передается на следующий оператор. Наконец, оператор On Error GoTo 0 отключает активизированный обработчик прерываний от ошибок в текущем модуле. В заключение отметим, что в VBA сохранился оператор безусловной передачи управления на метку или на номер строки GoTo Строка.

Операторы цикла. В языке VBA имеется стандартный набор операторов цикла. Далее приведено их формальное описание:

For Счетчик=Начало To Конец [Step Шаг]

[Операторы]

[Exit For]

[Операторы]

Next [Счетчик]

Если шаг не задан, то он предполагается равным 1. Строка Exit For предполагает прекращение выполнения оператора For.

For Each Элемент In Группа

[Операторы]

[Exit For]

[Операторы]

Next [Элемент]

Повторяет выполнение операторов для каждого элемента группы или семейства.

Do [While Условие]

[Операторы]

[Exit Do]

[Операторы]

Loop

или

Do

[Операторы]

[Exit Do]

[Операторы]

Loop [While Условие]

Операторы выполняются до тех пор, пока Условие = True.

Do [Until Условие]

[Операторы]

[Exit Do]

[Операторы]

Loop

или

Do

[Операторы]

[Exit Do]

[Операторы]

Loop [Until Условие]

Приведенные выше операторы выполняются до тех пор, пока Условие = False.

While Условие

[Операторы]

Wend

Оператор выполняется, пока Условие имеет значение True.

Процедуры и функции. Процедура представляет собой самостоятельную программу, которая может быть вызвана из другой программы. Отличие процедуры от функции заключается в том, что функция вызывается через оператор присваивания и заносит некоторое значение в ячейку, имя которой указано перед оператором присваивания. Процедура вызывается просто по имени (без использования оператора присваивания). Формально процедура может быть описана как:

[Private или Public] [Static] Sub Имя [(СписокАргументов)]

[Операторы]

[Exit Sub]

[Операторы]

End Sub

Ключевое слово Private означает, что процедура может быть вызвана только из того модуля, в котором она описана. Если указано ключевое слово Public, процедура может быть вызвана из других процедур любых модулей. Установленный ключ Static означает, что локальные переменные процедуры сохраняют свои значения между вызовами и могут быть использованы в последующих вычислениях при следующем вызове процедуры. Имя представляет собой обычный идентификатор языка VBA, а СписокАргументов представляет собой обычный список формальных параметров процедуры, каждый элемент которого имеет особый формат:

[Optional] [ByVal или ByRef] [ParamArray] ИмяПеременной[()] _

[As Тип] [=поУмолчанию]

Ключевое слово Optional означает, что элемент является необязательным и должен иметь тип Variant. Все последующие элементы списка должны иметь такой же ключ и тип. Ключ ByVal означает, что параметр передается по значению в отличие от ByRef, где параметр передается по ссылке (используется по умолчанию). Ключевое слово ParamArray может быть использовано только с последним элементом списка формальных параметров и позволяет передавать динамически объявляемый массив. Ключ Тип представляет собой тип передаваемого параметра (табл. 3.1), а значение поУмолчанию может использоваться только с ключом Optional и задает значение переменной.

Функция формально может быть описана как:

[Private или Public] [Static] Function Имя [(СписокАргументов)] [As Тип]

[Операторы]

[Имя=Выражение]

[Exit Sub]

[Операторы]

[Имя=Выражение]

End Sub

Ключевые слова и формат формальных параметров аналогичны описанным выше.

Как было указано ранее, функция вызывается по имени с фактическими параметрами, причем в строке вызова записывается оператор присвоения, который заносит в ячейки вызываемой программы вычисленное функцией значение. Подпрограмма может вызываться двумя способами: просто по имени, тогда список формальных параметров не заключается в круглые скобки, и с использованием ключевого слова Call (формальные параметры помещаются в круглые скобки).

2. Порядок выполнения работы

1. Включите компьютер. Загрузите *Windows*. Загрузите *Excel* и выполните команду *Файл/Открыть*. При стандартной настройке на экране появится окно *Открытие документа*, а в окне *Папка* появится название основной рабочей папки *Мои документы*. Если по каким-то причинам установилась другая папка, нажмите кнопку меню *Папка* и выберете папку *Мои документы*. Ниже будет показано содержимое папки *Мои документы*. Найдите папку с номером вашей группы. Откройте эту папку и найдите файл, соответствующий вашей фамилии, установите на него курсор. Нажмите кнопку *Открыть* и убедитесь, что в верхней левой области экрана после текста *Microsoft Excel* появилось название рабочей папки с вашей фамилией.

2. Перейдите на следующий лист рабочего поля. Если свободные листы отсутствуют, выполните команду *Вставить/Лист*.

3. Последовательностью команд *Сервис/Макрос/Редактор Visual Basic* запустите редактор VBA.

4. Выполните команду *Вставка/Модуль* и создайте новый модуль. Выберете его значок в окне проекта, чтобы активизировать окно редактора для создания

собственной функции. В открывшемся окне напишите на VBA программу функции в соответствии со своим индивидуальным заданием.

5. Убедитесь, что созданная вами функция появилась в списке функций в категории *Определенные пользователем*. Проверьте работоспособность созданной вами функции из рабочего листа *Excel*.

6. Сохраните результаты выполнения работы в файле на диске.

3. Порядок оформления отчета

Отчетом о [лабораторной работе](#) является файл с именем, совпадающим с фамилией студента с результатами работы в папке *Мои документы/номер группы*.

4. Варианты заданий к выполнению лабораторной работы

Напишите функцию или набор функций, автоматизирующих расчеты в следующих примерах таблиц.

1. Ведомость на начисление [заработной платы](#). (Фамилия, начисления, ставка, к выдаче).

2. Расписание движения самолетов. (Направление, время отправления, время прибытия, время в пути, стоимость билета, часовая стоимость полета).

3. Ведомость сдачи экзаменов. (Фамилия, оценки по предметам, даты, общее количество оценок разного вида, средний балл).

4. Ведомость складских остатков. (Наименование, дата поступления, количество, цена за штуку, стоимость, время хранения).

5. Каталог библиотеки. (Наименование, количество, цена, стоимость, количество книг определенного года издания).

6. Список товаров в магазине. (Наименование, количество, цена, стоимость, срок хранения).

7. График дежурств. (Фамилия, дни недели, общее время, среднее время).

8. Расписание занятий. (Дисциплина, дни недели, количество часов, среднее количество часов за неделю).

9. Расчет [коммунальных услуг](#). (Вид платежа, тариф, стоимость по тарифу, скидка по льготе, начислено).

10. Расчет оплаты за электроэнергию. (Фамилия, показания счетчика от, показания счетчика до, дата, расход, тариф, начислено, льготы, сумма).

5. Контрольные вопросы

1. Перечислите основные типы данных VBA.

2. Перечислите операторы изменения последовательности выполнения программы.

3. Каковы особенности использования функций в VBA.

4. Перечислите операторы цикла VBA.

5. Каково правило образования имен идентификаторов в VBA.

ВСТРОЕННЫЕ ДИАЛОГОВЫЕ ОКНА

Цель работы: изучить основные приемы работы с встроенными диалоговыми окнами.

1. Методические указания

Возможности ввода вывода информации в VBA чрезвычайно велики. Объектно–ориентированная структура языка позволяет программисту использовать в своих пользовательских приложениях все возможности, созданные для организации многооконной работы, развитых диалогов ввода-вывода, работы с мышью, а также файловой работы. Имеющиеся возможности организации ввода-вывода можно условно разбить на следующие основные группы в соответствии с имеющимися в составе VBA объектами и методами:

- § встроенные диалоговые окна;
- § средства создания пользовательского интерфейса;
- § средства создания панели инструментов;
- § средства работы с графическими объектами;
- § средства построения диаграмм и графиков;
- § средства работы с файлами.

Встроенные диалоговые окна являются простейшей формой организации ввода-вывода и могут рассматриваться как операторы ввода-вывода языка VBA. Функция `InputBox` выводит на экран диалоговое окно, содержащее сообщение и поле ввода, устанавливает режим ожидания ввода текста пользователем или нажатия кнопки, а затем возвращает значение типа `String`, содержащее текст, введенный в поле. Формат оператора:

```
InputBox(prompt[,title][,default][,Xpos][,Ypos][,helpfile, context])
```

Здесь `prompt` – строковое выражение, которое будет отображаться как сообщение в диалоговом окне, `title` - сообщение, отображаемое в заголовке окна (если оно опущено, то отображается имя приложения), `default` – сообщение, которое будет выводиться в строке при запуске, `Xpos` и `Ypos` задают положение окна на экране, `helpfile` и `context` – соответственно имя файла и номер раздела справочной системы.

Функция MsgBox выводит на экран диалоговое окно с сообщением и ожидает нажатия кнопки пользователем возвращая значение нажатой кнопки как число типа Integer. Формальная запись функции выглядит как:

MsgBox(prompt[,buttons][,title][, helpfile, context])

Одноименные с функцией InputBox аргументы имеют тот же самый смысл, а значение buttons определяется как сумма констант, задающих число и тип отображаемых кнопок, тип используемого значка и основную кнопку (см. табл. 4.1).

Таблица 4.1.

Константы аргумента buttons и возвращаемые значения функции MsgBox

Идентификатор константы	Значение	Пояснение
Количество и вид кнопок		
VbOKOnly	0	Только кнопка ОК
VbOKCancel	1	Кнопки ОК и Отмена
VbAbortRetryIgnore	2	Кнопки Стоп, Повтор, Пропустить
VbYesNoCancel	3	Кнопки Да, Нет, Отмена
VbYesNo	4	Кнопки Да и Нет
VbRetryCancel	5	Кнопки Повтор и Отмена
Информационные значки		
VbCritical	16	Ошибка
VbQuestion	32	Вопрос
VbExclamation	48	Утверждение

VbInformation	64	Информация
Основная кнопка		
VbDefaultButton1	0	Кнопка 1
VbDefaultButton2	256	Кнопка 2
VbDefaultButton3	512	Кнопка 3
VbDefaultButton4	768	Кнопка 4
Возвращаемое значение		
VbOK	1	Нажато ОК
VbCancel	2	Нажата Отмена
VbAbort	3	Нажата Прервать
VbRetry	4	Нажата повторит
VbIgnore	5	Нажато Пропустить
VbYes	6	Нажато Да
VbNo	7	Нажато Нет

2. Порядок выполнения работы.

1. Включите компьютер. Загрузите *Windows*. Загрузите *Excel* и выполните команду *Файл/Открыть*. При стандартной настройке на экране появится окно *Открытие документа*, а в окне *Папка* появится название основной рабочей папки *Мои документы*. Если по каким-то причинам установилась другая папка, нажмите кнопку меню *Папка* и выберите папку *Мои документы*. Ниже будет показано содержимое папки *Мои документы*. Найдите папку с номером вашей группы. Откройте эту папку и найдите файл, соответствующий вашей фамилии, установите на него курсор. Нажмите кнопку *Открыть* и убедитесь, что в

верхней левой области экрана после текста *Microsoft Excel* появилось название рабочей папки с вашей фамилией.

2. Перейдите на следующий лист рабочего поля. Если свободные листы отсутствуют, выполните команду *Вставить/Лист*.

3. Последовательностью команд *Сервис/Макрос/Редактор Visual Basic* запустите редактор VBA.

4. Модернизируйте написанную вами в процессе выполнения предыдущей лабораторной работы программу таким образом, чтобы включить в ее состав функции MsgBox и InputBox в различных вариантах их использования.

5. Проверьте работоспособность созданной вам функции из рабочего листа *Excel*.

6. Сохраните результаты выполнения работы в файле на диске.

3. Порядок оформления отчета

Отчетом о лабораторной работе является файл с именем, совпадающим с фамилией студента с результатами работы в папке *Мои документы/номер группы*.

4. Контрольные вопросы.

1. В чем отличие функций InputBox и MsgBox?

2. Каков смысл аргумента title?

3. Каков смысл аргумента default?

4. Каков смысл аргумента buttons?

5. Каков смысл аргумента prompt?

6. Каков смысл аргумента context?

7. Каков смысл аргументов helpfile, context?

СРЕДСТВА СОЗДАНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

Цель работы: изучить основные приемы работы с конструктором форм.

1. Методические указания

Средства создания пользовательского интерфейса позволяют создавать собственные пользовательские формы, для чего в состав объектов введен специальный объект UserForm. Для создания пользовательской формы откройте лист *Excel* и командой *Вид/Панели инструментов* включите панель *Элементы управления*. Пример панели представлен на рис. 5. На панели в данном случае присутствуют кнопки, соответствующие следующим элементам управления: *Флажок*, *Поле*, *Кнопка*, *Переключатель*, *Список*, *Поле со списком*, *Выключатель*, *Счетчик*, *Полоса прокрутки*, *Надпись*. Конкретный набор кнопок меню может задаваться в частности с помощью кнопки *Другие элементы*, позволяющей добавлять в меню большой набор кнопок, соответствующих другим отображаемым классам. Кроме этого имеются кнопки управления режимом конструктора, *Свойства* и *исходный текст*.



Рис. 5. Пример панели *Элементы управления*.

Для вставки необходимой формы включите режим конструктора и нажмите требуемую кнопку в меню. После этого выберете маркером место на рабочем листе, куда вы хотите внедрить форму и нажмите левую клавишу мыши. Нажмите кнопку *Свойства* и ознакомьтесь со списком свойств внедряемого объекта. Список может быть упорядочен по [алфавиту](#) или по категориям. Отметим, в частности, что свойство *Caption* позволяет устанавливать текст,

возвращаемый в заголовке формы, а свойство *Value* – текущее ее значение. При необходимости исходные значения свойств могут быть изменены.

При нажатии кнопки *Исходный текст* запускается редактор VBA, причем в окне текста появляется заготовка программы обработки прерывания пользователя при работе с формой, выполняемых при возникновении определенных событий. Полный перечень возможных событий можно посмотреть в открывающемся окне в правой верхней части рабочего окна редактора (рис.6.). При выборе из меню интересующего события можно получить заготовку программы, выполняемой при возникновении этого события. Отметим, что при обращении к имени свойства необходимо указывать и имя формы. Так например, при обращении к свойству *Value* формы *TextBox1* необходимо записать *TextBox1.Value*.

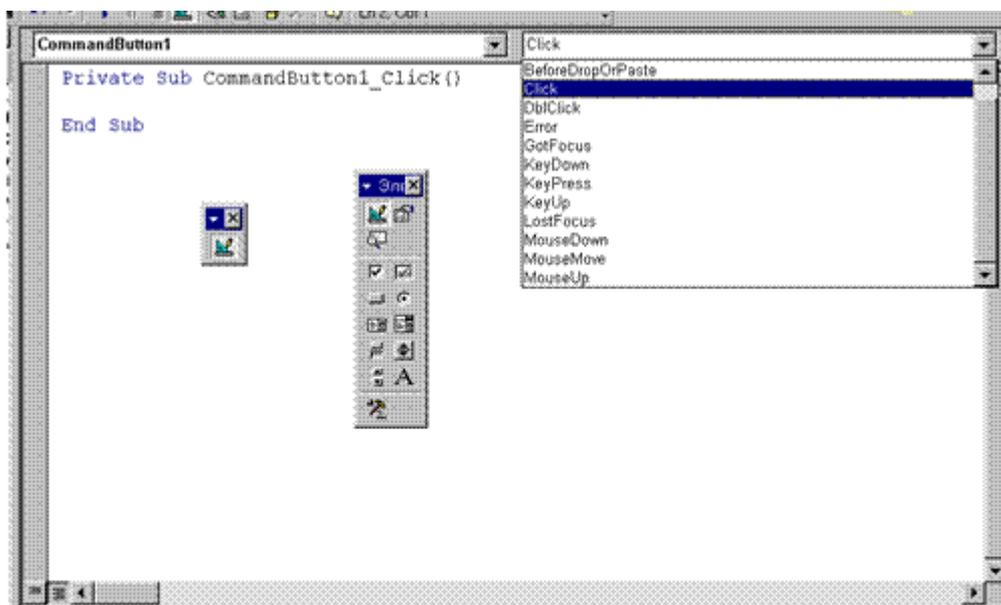


Рис. 6. Пример открытого меню событий при обработке экранной формы.

Программа обработки событий может использовать и изменять значения свойств формы. При ее создании могут быть использованы все возможности языка. Любая из созданных таким образом программ может быть вызвана из другой программы. Программа обработки начинает выполняться после выключения режима конструктора соответствующей кнопкой меню. В качестве примера ниже приведена программа обработки нажатия флажка и программа выдачи текстового сообщения в рабочее окно.

‘Обработка нажатия мышкой на флажок

```
Private Sub CheckBox1_Click()
```

‘Проверка состояния флажка

```
If CheckBox1.Value Then
```

‘Выдача сообщения в рабочую ячейку

```
Range("B3").Select
```

```
ActiveCell.FormulaR1C1 = "Флажок включен"
```

```
Else
```

```
Range("B3").Select
```

```
ActiveCell.FormulaR1C1 = "Флажок выключен"
```

```
End If
```

‘Вызов функции выдачи сообщения в окно

```
TextBox1_Change
```

```
End Sub
```

‘Функция выдачи сообщения в окно

```
Private Sub TextBox1_Change()
```

‘Проверка состояния свойства в другом объекте

```
If CheckBox1.Value Then
```

‘Изменение значения свойства

```
TextBox1.Value = "Флажок включен"
```

Else

TextBox1.Value = "Флажок выключен"

End If

End Sub

2. Порядок выполнения работы.

1. Включите компьютер. Загрузите *Windows*. Загрузите *Excel* и выполните команду *Файл/Открыть*. При стандартной настройке на экране появится окно *Открытие документа*, а в окне *Папка* появится название основной рабочей папки *Мои документы*. Если по каким-то причинам установилась другая папка, нажмите кнопку меню *Папка* и выберете папку *Мои документы*. Ниже будет показано содержимое папки *Мои документы*. Найдите папку с номером вашей группы. Откройте эту папку и найдите файл, соответствующий вашей фамилии, установите на него курсор. Нажмите кнопку *Открыть* и убедитесь, что в верхней левой области экрана после текста *Microsoft Excel* появилось название рабочей папки с вашей фамилией.

2. Перейдите на следующий лист рабочего поля. Если свободные листы отсутствуют, выполните команду *Вставить/Лист*.

3. Последовательностью команд *Сервис/Макрос/Редактор Visual Basic* запустите редактор VBA.

4. Модернизируйте написанную вами в процессе выполнения предыдущей лабораторной работы программу таким образом, чтобы включить в ее состав элементы пользовательских форм.

5. Проверьте работоспособность созданной вам функции из рабочего листа *Excel*.

6. Сохраните результаты выполнения работы в файле на диске.

3. Порядок оформления отчета

Отчетом о лабораторной работе является файл с именем, совпадающим с фамилией студента с результатами работы в папке *Мои документы/номер группы*.

4. Контрольные вопросы.

1. Как можно связать пользовательскую форму с переменными программы?
2. Как определить возможные события объекта?
3. Как создать функцию, обрабатывающую различные действия пользователя?
4. Как можно изменить свойства формы?
5. Каково назначение кнопки включения – выключения режима работы конструктора форм
6. Как можно расширить набор используемых пользовательских форм?
7. Перечислите основные типы пользовательских форм VBA.

РАБОТА С ВНЕШНИМИ ФАЙЛАМИ

Цель работы: изучить основные приемы работы с внешними файлами.

1. Методические указания

Средства работы с файлами, реализованные в VBA, позволяют осуществлять чтение и запись текстовых файлов в процессе выполнения программы.

Используются файлы последовательного доступа, в которых записи разделяются специальными символами, и файлы произвольного доступа с записями фиксированной длины, обращение к которым возможно по номеру записи. Набор операторов языка, предназначенных для работы с файлами, стандартен.

Инструкция Open позволяет установить связь между файлом и программой. Формат инструкции:

Open Путь For Режим [Access Доступ] [Блокировка] As [#]Номер файла
[Len=Длина]

Выражение Путь представляет собой строку, указывающую имя файла. Параметр Режим может принимать значения Append (добавления), Binary (двоичный), Input, Output или Random (произвольного доступа), а параметр Доступ – значения Read, Write или ReadWrite. Значение Блокировка позволяет регулировать возможности доступа к открытому файлу другими процессами и может принимать значения Shared (совместный доступ), Lock Read, Lock Write или Lock Read Write. Число Номер файла может изменяться в диапазоне от 1 до 255. Длина – число байт, меньшее или равное 32767, имеющее смысл размера [буфера](#).

Команда закрывает файл или файлы с указанными номерами и имеет формат:

Close [Список номеров файлов]

Операторы Reset и FreeFile соответственно закрывают все активные файлы и возвращают свободный номер, который может быть использован при открытии файла.

Непосредственная запись в файл последовательного доступа может осуществляться командами Print для форматированных и Write для неформатированных данных. Аргументами в обоих случаях являются НомерФайла и СписокВывода. Отличие заключается в том, что в первом случае разделителем записей является символ точка с запятой (;), а во втором

разделители добавляются автоматически после каждого элемента записи, а строки автоматически заключаются в кавычки.

Ввод данных из файла осуществляется командой `Input`. Ее аргументами являются номер файла и список переменных, разделенных запятыми, в которых должны быть размещены данные из файла. Для определения конца файла используется инструкция `EOF(НомерФайла)`, которая принимает значение `True` при достижении конца файла. Отметим, что существует одноименная функция `Input(Число, НомерФайла)`, возвращающая строку размером `Число` символов при каждом обращении.

При работе с файлами произвольного доступа используются команды:

`Put [#]НомерФайла, [Номер записи], ИмяПеременной`

`Get [#]НомерФайла, [Номер записи], ИмяПеременной`

Значение `Номер записи` имеет смысл номера байта (`Binary`) или номера записи (`Random`).

Функции `Seek(НомерФайла)` и `LOF(НомерФайла)` возвращают значение типа `Long`, представляющее собой соответственно текущее положение указателя и размер открытого файла. Функция `FileLen(Путь)` выдает значение типа `Long`, содержащее размер файла в байтах.

2. Порядок выполнения работы.

1. Включите компьютер. Загрузите *Windows*. Загрузите *Excel* и выполните команду *Файл/Открыть*. При стандартной настройке на экране появится окно *Открытие документа*, а в окне *Папка* появится название основной рабочей папки *Мои документы*. Если по каким-то причинам установилась другая папка, нажмите кнопку меню *Папка* и выберете папку *Мои документы*. Ниже будет показано содержимое папки *Мои документы*. Найдите папку с номером вашей группы. Откройте эту папку и найдите файл, соответствующий вашей фамилии, установите на него курсор. Нажмите кнопку *Открыть* и убедитесь, что в верхней левой области экрана после текста *Microsoft Excel* появилось название рабочей папки с вашей фамилией.

2. Перейдите на следующий лист рабочего поля. Если свободные листы отсутствуют, выполните команду *Вставить/Лист*.

3. Последовательностью команд *Сервис/Макрос/Редактор Visual Basic* запустите редактор VBA.

4. Выполните команду *Вставка/Модуль* и создайте новый модуль. Выберите его значок в окне проекта, чтобы активизировать окно редактора для создания собственной функции. В открывшемся окне напишите на VBA программу функции, демонстрирующую возможности вывода в файл информации в соответствии со своим индивидуальным заданием.

5. Проверьте работоспособность созданной вам функции из рабочего листа *Excel*.

6. Сохраните результаты выполнения работы в файле на диске.

3. Порядок оформления отчета

Отчетом о лабораторной работе является файл с именем, совпадающим с фамилией студента с результатами работы в папке *Мои документы/номер группы*.

4. Варианты заданий к выполнению лабораторной работы.

Напишите функцию или набор функций, осуществляющие следующие операции с файлами.

1. Запись и чтение файла последовательного доступа, данные форматированные.

2. Запись и чтение файла последовательного доступа, данные неформатированные.

3. Запись и чтение файла произвольного доступа, данные форматированные.

4. Запись и чтение заданной записи (записей) файла произвольного доступа, данные форматированные.

5. Запись, чтение и добавление записи (записей) файла произвольного доступа, данные форматированные.

6. Копирование файла последовательного доступа, данные неформатированные.

7. Копирование файла последовательного доступа, данные форматированные.
8. Совместная работа с файлом последовательного доступа, данные неформатированные.
9. Совместная работа с файлом последовательного доступа, данные форматированные.
10. Совместная работа с файлом произвольного доступа, данные форматированные.

5. Контрольные вопросы.

1. В чем отличие форматированной и не форматированной записи в файл?
2. Какие существуют возможности для организации совместного доступа к файлу?
3. Как можно найти нужную запись в файле?
4. Как копировать файлы?
5. Как можно изменять содержимое файла?
6. Что такое двоичный файл и где он используется?
7. Что происходит при выполнении операции закрытия файла?
8. Как можно определить конец файла?
9. Перечислите основные типы файлов VBA.

СОЗДАНИЕ КЛАССА

Цель работы: изучить методы создания пользовательских классов и научиться работать с объектами.

1. Методические указания

Язык VBA имеет средства, обеспечивающие возможности создания собственного пользовательского класса. С этой целью в состав компилятора включена специальная форма *Class Module*, позволяющая конструировать собственный класс. Для его создания могут использоваться специальные процедуры инициализации и удаление класса. А также процедуры задания имен свойств числовых данных и объектов, а также считывания значения свойств:

```
Private Sub Class_Initialize()
```

```
[Public или Private] [Static] Property Let Имя [(Список аргументов)]
```

```
[Public или Private] [Static] Property Set Имя [(Список аргументов)]
```

```
[Public или Private] [Static] Property Get Имя [(Список аргументов)]
```

```
Private Sub Class_Terminate()
```

Процедура `Class_Initialize` выполняется один раз перед началом работы с объектом и может использоваться для задания начальных значений свойствам и переменным. При завершении работы с классом память может быть освобождена процедурой `Class_Terminate`. Процедуры `Property Let` и `Property Set` задают конкретные значения свойств соответственно числовым данным и объектам. В составе модуля может быть неограниченно много подобных процедур, каждая из которых задает свое значение конкретному свойству `Имя`. Процедура `Property Get` позволяет считать значение конкретного свойства `Имя` и используется собственно для программирования свойства класса.

Обращение к классу производится из обычного программного модуля, включенного в состав проекта. Запрос на создание объекта оформляется в виде инструкции

```
Set Имя_Объекта =[New] Тип_Объекта_Метода
```

Отказ от дальнейшей работы с объектом оформляется как

Set Имя_Объекта = Nothing

Ключевое слово New означает, что объект создается впервые.

В качестве примера рассмотрим последовательность действий, выполняемых при создании класса. Предположим, что необходим объект, позволяющий выполнить процедуру отыскания минимального или максимального элемента массива. Процесс создания объекта начинается с объявления глобальных переменных в новом окне *Class Module*. После запуска нового окна создания модуля в окне *Свойства* необходимо задать имя создаваемого класса (в нашем примере Экстремум).

Option Explicit 'Задание режима обязательного описания переменных

'Описание общих переменных класса

Dim title As String

Dim ВидЭкстремума As Boolean, Таблица() As Double, _

РазмерТаблицы As Integer

Dim max, min As Double

Процедура инициализации класса имеет вид:

Private Sub class_initialize()

'Задание начальных значений

max = 1.E+308

min = -1.E+308

title = "Экстремум"

ВидЭкстремума = True 'По умолчанию ищется максимум

РазмерТаблицы = 1

```
ReDim Таблица(1 To РазмерТаблицы)
```

```
End Sub
```

Свойство задания размера таблицы описано в следующем модуле:

```
Public Property Let ЗадатьРазмерТаблицы(ByVal НовыйРазмерТаблицы As Integer)
```

```
If (НовыйРазмерТаблицы < 1 Or НовыйРазмерТаблицы > 32000) Then
```

```
MsgBox ("НовыйРазмерТаблицы " & Str(НовыйРазмерТаблицы) & "выходит за пределы допустимого диапазона!"), vbOKOnly, title
```

```
Else
```

```
РазмерТаблицы = НовыйРазмерТаблицы
```

```
ReDim Таблица(1 To РазмерТаблицы)
```

```
End If
```

```
End Property
```

Свойство задания рабочей таблицы запрограммировано в следующем виде:

```
Public Property Let ЗадатьТаблицу(ByRef Tabl As Variant)
```

```
Dim i As Integer
```

```
For i = 1 To РазмерТаблицы
```

```
Таблица(i) = Tabl(i)
```

```
Next i
```

```
End Property
```

Отметим, что рабочий массив передается свойству как переменная типа Variant. Свойство задания вида поиска оформлено в виде функции:

```
Public Property Let ЗадатьВидПоиска(ByVal МаксимумИлиМинимум As Boolean)
```

```
If МаксимумИлиМинимум And ВидЭкстремума Then
```

```
MsgBox "Поиск максимума уже задан", vbOKOnly, title
```

```
Else
```

```
If (Not МаксимумИлиМинимум) And (Not ВидЭкстремума) Then
```

```
MsgBox "Поиск минимума уже задан", vbOKOnly, title
```

```
Else
```

```
If (Not МаксимумИлиМинимум) And ВидЭкстремума Then
```

```
MsgBox "Задается поиск минимума", vbOKOnly, title
```

```
Else
```

```
If МаксимумИлиМинимум And (Not ВидЭкстремума) Then
```

```
MsgBox "Задается поиск максимума", vbOKOnly, title
```

```
Else
```

```
End If
```

```
ВидЭкстремума = МаксимумИлиМинимум
```

```
End If
```

```
End If
```

```
End If
```

End Property

Наконец свойство определения величины экстремума оформлено в следующем виде:

```
Public Property Get Величина_экстремума() As Double
```

```
Dim extr As Double, i As Integer
```

```
If ВидЭкстремума Then
```

```
extr = min
```

```
Else
```

```
extr = max
```

```
End If
```

```
For i = 1 To РазмерТаблицы
```

```
If ВидЭкстремума Then
```

```
If Таблица(i) > extr Then
```

```
extr = Таблица(i)
```

```
Else
```

```
End If
```

```
Else
```

```
If Таблица(i) < extr Then
```

```
extr = Таблица(i)
```

```
Else
```

End If

End If

Величина_экстремума = extr

Next i

End Property

Пример программы, использующей созданный класс, представлен в следующем фрагменте:

Sub main()

Dim Объект1, Объект2 As Экстремум

Dim AA(10), res As Double, i As Integer

Dim ss As Variant

Set Объект1 = New Экстремум

With Объект1

.ЗадатьРазмерТаблицы = 10

.ЗадатьВидПоиска = True

End With

For i = 1 To 10

AA(i) = i

Next i

ss = AA()

With Объект1

.ЗадатьТаблицу = ss

res = .Величина_экстремума

End With

MsgBox "Экстремум равен" & Str(res)

ss(4) = 1000

Объект1.ЗадатьТаблицу = ss

res = Объект1.Величина_экстремума

MsgBox "Экстремум равен" & Str(res)

Set Объект2 = New Экстремум

Объект2.ЗадатьВидПоиска = False

Объект2.ЗадатьТаблицу = ss

res = Объект2.Величина_экстремума

MsgBox "Экстремум равен" & Str(res)

End Sub

Таким образом, созданный класс может использоваться в самых разнообразных приложениях.

2. Порядок выполнения работы.

1. Включите компьютер. Загрузите *Windows*. Загрузите *Excel* и выполните команду *Файл/Открыть*. При стандартной настройке на экране появится окно *Открытие документа*, а в окне *Папка* появится название основной рабочей папки *Мои документы*. Если по каким-то причинам установилась другая папка, нажмите кнопку меню *Папка* и выберете папку *Мои документы*. Ниже будет

показано содержимое папки *Мои документы*. Найдите папку с номером вашей группы. Откройте эту папку и найдите файл, соответствующий вашей фамилии, установите на него курсор. Нажмите кнопку *Открыть* и убедитесь, что в верхней левой области экрана после текста *Microsoft Excel* появилось название рабочей папки с вашей фамилией.

2. Перейдите на следующий лист рабочего поля. Если свободные листы отсутствуют, выполните команду *Вставить/Лист*.

3. Последовательностью команд *Сервис/Макрос/Редактор Visual Basic* запустите редактор VBA.

4. Разработайте собственный объект, реализующий операцию поиска экстремума. Дополните объект свойствами в соответствии с индивидуальными заданиями.

5. Проверьте работоспособность созданной вам функции из рабочего листа *Excel*.

6. Сохраните результаты выполнения работы в файле на диске.

3. Порядок оформления отчета

Отчетом о лабораторной работе является файл с именем, совпадающим с фамилией студента с результатами работы в папке *Мои документы/номер группы*.

4. Варианты заданий к выполнению лабораторной работы.

1. Дополните созданный вами объект свойством расчета количества локальных экстремумов.

2. Разработайте свойство отыскания номера элемента массива, в котором находится глобальный максимум.

3. Разработайте свойство отыскания номеров элементов массива, в которых находятся локальные максимумы.

4. Разработайте свойство, предназначенное для упорядочения локальных максимумов по мере их возрастания (убывания).

5. Дополните объект свойством построения графика.
6. Разработайте свойство сортировки (ранжирования) элементов массива по убыванию (возрастанию).
7. Разработайте свойство, позволяющее определить диапазон номеров элементов массива, в которых находятся элементы, образующие глобальный максимум.
8. Разработайте свойство, позволяющее определить диапазон номеров элементов массива, в которых находятся локальные максимумы.
9. Дополните объект свойством построения [гистограммы](#).
10. Разработайте свойство, позволяющее редактировать информацию в массиве.

5. Контрольные вопросы.

1. В чем отличие модуля класса от обычного модуля?
2. Как определить возможные свойства объекта?
3. Перечислите основные типы процедур, позволяющие создавать классы в VBA.
4. Как создать функцию, использующую различные свойства объекта?
5. Как можно изменить свойства объекта?
6. Как можно расширить набор используемых свойств?

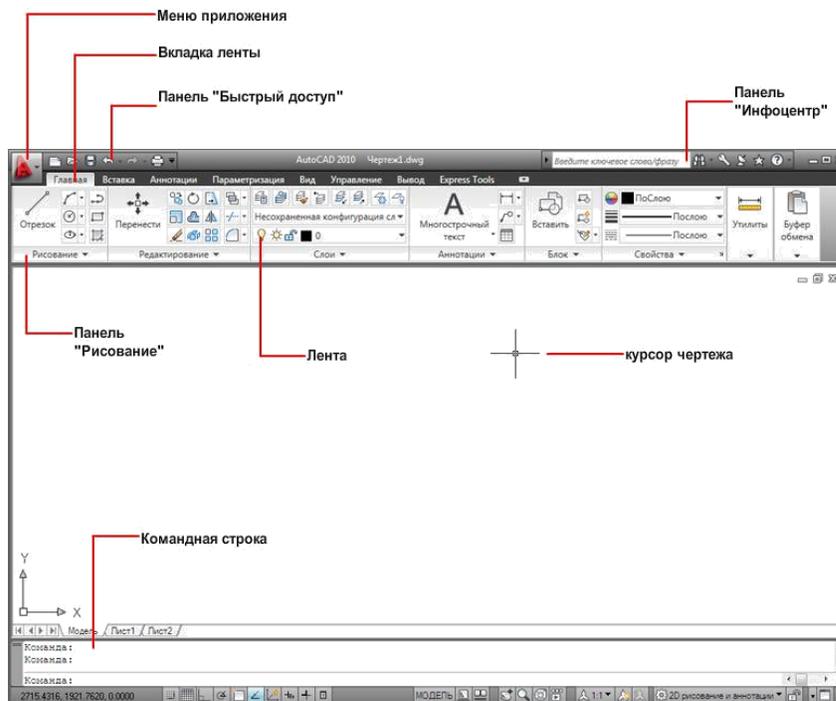
Литература

1. Объектно-ориентированный анализ и проектирование с примерами приложений на C++, 2-е изд./Пер. с англ. – М.: «Издательство Бином», СПб.: «Невский Диалект», 2000. – 560 с.
2. Гарнаев VBA. – СПб.: БХВ – Санкт-Петербург, 1999, - 512 с.

Начальные сведения

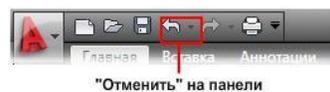
Запуск AutoCAD

- 1 Для запуска AutoCAD перейдите в меню "Пуск" (Windows) > "(Все) программы" > Autodesk > AutoCAD 2010 > AutoCAD 2010-Русский.
- 2 Ознакомьтесь с отмеченными далее основными областями диалогового окна AutoCAD.



Команда "Отменить": как начать заново

Если вы ошиблись и хотите начать упражнение заново, можно воспользоваться командой "Отменить". Щелкните значок "Отменить" в панели "Быстрый доступ", расположенной в верхней части окна приложения. Для отмены результата выполнения всех предыдущих команд потребуется щелкнуть значок несколько раз



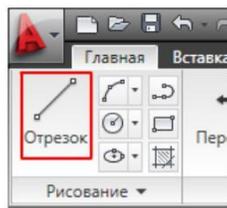
ESC - отмена команды

При непреднамеренном щелчке в области рисования, при случайном запуске команды, а также когда отображается контекстное меню, можно отменить выполненное действие, нажав клавишу ESC.

Построение отрезков

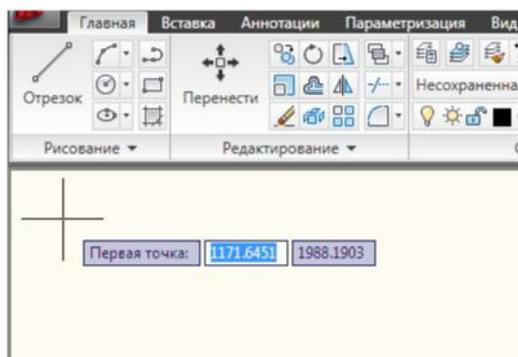
Построение отрезка

- 1 Щелкните на ленте вкладку "Главная" ► панель "Рисование" ► "Отрезок".

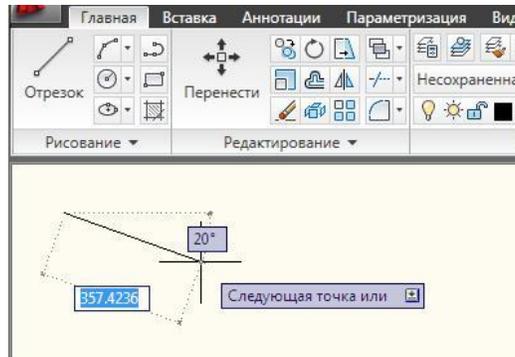


- 2 Переместите курсор вниз в пустую область чертежа. Отображается перекрестье с тремя текстовыми полями. Поле "Первая точка" представляет собой поле динамической подсказки. Динамическая подсказка позволяет сосредоточиться на работе и исключает необходимость ввода в командной строке.

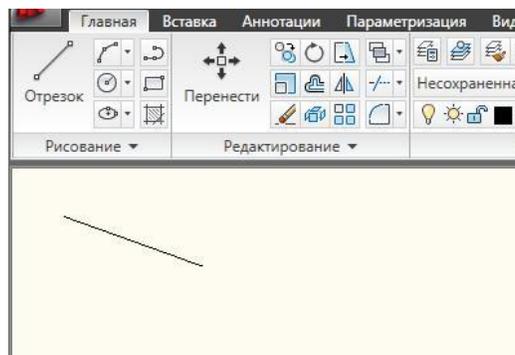
ПРИМЕЧАНИЕ Если поле динамической подсказки не отображается, нажмите клавишу F12.



- 3 В двух других текстовых полях отображаются данные о местоположении курсора (координаты X и Y). Переместите курсор и обратите внимание на изменение координат.
- 4 В пустой области чертежа щелчком укажите начальную точку первого отрезка. В поле динамической подсказки отображается текст: "Следующая точка или".



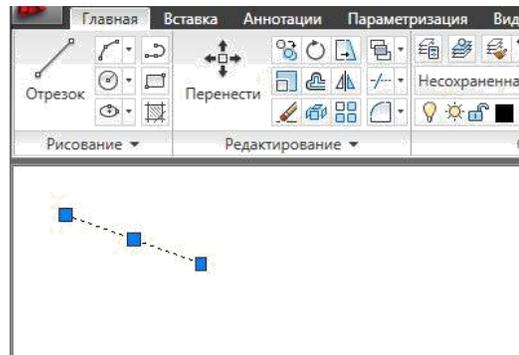
- 5 Переместите курсор в любом направлении и щелчком укажите конечную точку отрезка. Отображаются другие текстовые поля с дополнительной информацией об отрезке. В данное время эти поля можно проигнорировать.
- 6 Для завершения построения отрезка нажмите клавишу ENTER. Ваш первый отрезок построен!



Выбор и удаление отдельного отрезка

Прежде чем перейти к построению последовательности отрезков, выберите и удалите только что созданный отрезок.

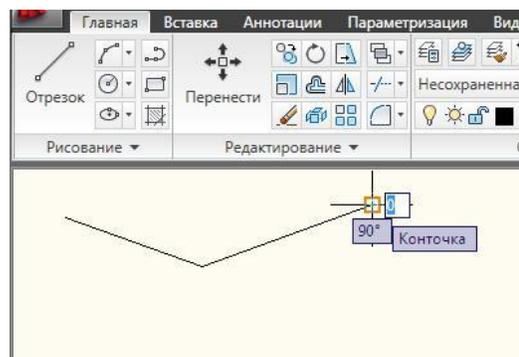
- 1 Наведите курсор на отрезок, чтобы выделить его, и отрезок будет отображаться пунктиром более темного цвета.
- 2 Щелчком выберите отрезок. Когда отрезок выбран, отображаются три квадратных ручки выбора, и отрезок становится пунктирным.



- 3 Для удаления отрезка нажмите клавишу DELETE. В этом уроке было показано, как следует выбирать отдельный объект и удалять его.

Построение последовательности соединенных отрезков

- 1 Щелкните на ленте вкладку "Главная" ➤ панель "Рисование" ➤ "Отрезок".
- 2 Переместите курсор вниз в пустую область чертежа. Щелчком задайте первую точку отрезка.
- 3 Переместите курсор и щелчком укажите следующую точку отрезка. Построен первый отрезок из последовательности соединенных отрезков.
- 4 Переместите курсор и щелчком задайте следующую точку. Построен второй отрезок из последовательности соединенных отрезков.

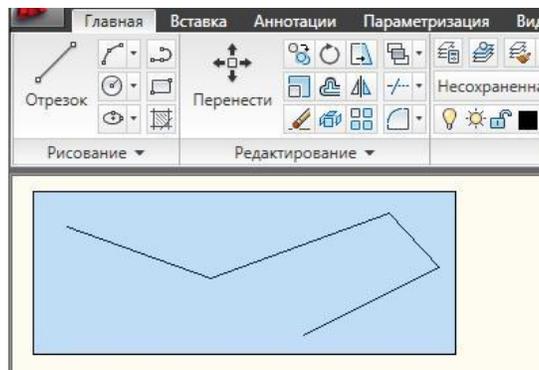


- 5 Продолжайте перемещать курсор и щелчками задавать точки и добавлять отрезки.
- 6 Для завершения построения последовательности отрезков нажмите клавишу ENTER.

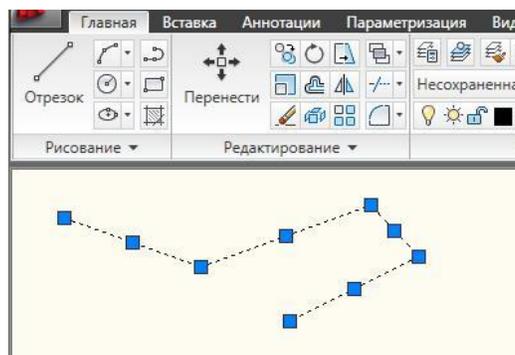
Выбор и удаление последовательности отрезков

Прежде чем перейти к соединению последовательности отрезков для создания объекта, удалите только что нарисованные отрезки. Далее будет показано, как выбрать все отрезки одновременно с помощью рамки выбора.

- 1 Установите курсор выше и левее отрезков.
- 2 Щелкните и при нажатой кнопке мыши перетащите курсор вправо и вниз, формируя вокруг отрезков рамку выбора.



- 3 Щелкните, когда все отрезки будут находиться внутри рамки. После щелчка отображаются квадратные ручки выбора, отрезки становятся пунктирными.



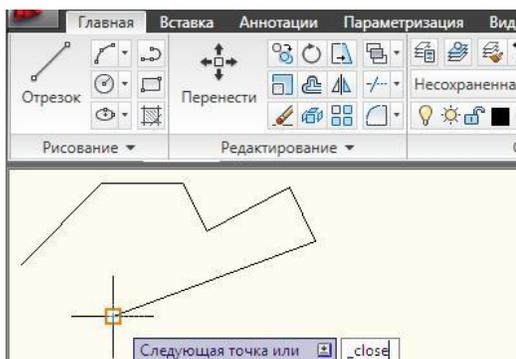
- 4 Для удаления отрезков нажмите клавишу DELETE.

Соединение последовательности отрезков для формирования объекта

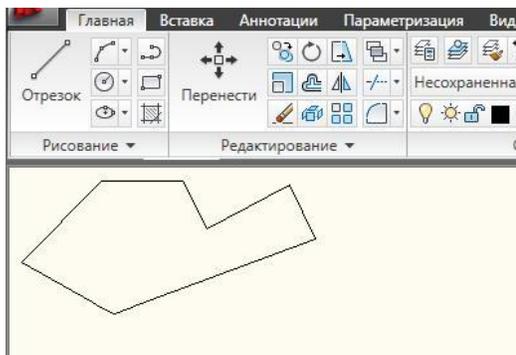
- 1 Щелкните на ленте вкладку "Главная" ► панель "Рисование" ► "Отрезок".



- 2 Переместите курсор вниз в пустую область чертежа. Щелчком задайте первую точку отрезка.
- 3 Переместите курсор и щелчком укажите следующую точку отрезка.
- 4 Продолжайте перемещать курсор и создавать новые отрезки.
- 5 Указав последнюю точку, без перемещения курсора введите с клавиатуры команду ЗАКРЫТЬ. При вводе команды ЗАКРЫТЬ она отображается в небольшом текстовом поле в окне чертежа.



- 6 Для завершения команды ЗАКРЫТЬ нажмите клавишу ENTER. Конец последнего построенного отрезка соединяется в начале первого отрезка. Такой способ непосредственного ввода команды называется динамическим вводом. Это быстрее и проще, чем вводить команду в командной строке.



7 Прежде чем перейти к следующему уроку, нажмите кнопку  ➤ "Закреть", чтобы закрыть чертеж. Сохранять чертеж не обязательно.

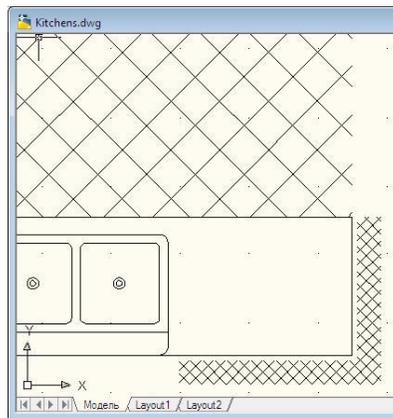
Зумирование и панорамирование с помощью колесика мыши

- 1 Щелкните  ► "Открыть" ► "Чертеж" и выберите в диалоговом окне выбора файлов файл *kitchens.dwg*. Возможно, потребуется перейти в папку *C:\Мои документы\Tutorials*. Это один из файлов, загруженный в начале данного учебного пособия.
- 2 Нажмите "Открыть".
- 3 Убедитесь, что активным является пространство модели. В противном случае щелкните вкладку "Модель" в левом нижнем углу окна программы.

Увеличение и уменьшение изображения с помощью колесика мыши

Самый простой способ увеличения и уменьшения изображения заключается в использовании колесика мыши.

- 1 Установите курсор в области рисования, затем, вращая колесико мыши вверх, увеличьте изображение фрагмента чертежа.



- 2 Для уменьшения изображения вращайте колесико мыши вниз.

Отображение всего чертежа в области рисования с помощью колесика мыши

- 1 С помощью колесика мыши увеличьте изображение фрагмента чертежа.
 - 2 Для отображения всего чертежа в области рисования дважды щелкните колесиком мыши.
-

СОВЕТ Если при увеличении изображения дуги и окружности становятся несглаженными или при невозможности выполнить увеличение или уменьшение, можно регенерировать отображение. Введите в поле динамического ввода ВСЕРЕГЕН и нажмите ENTER.

Панорамирование чертежа с помощью колесика мыши

Панорамирование чертежа используется для изменения положения вида без изменения масштаба изображения.

- 1 Установив курсор в области рисования, нажмите и удерживайте нажатым колесико мыши. Отображается значок в форме руки.
- 2 Для панорамирования чертежа перетащите курсор.
- 3 Чтобы прекратить панорамирование, отпустите колесико мыши.
- 4 Увеличьте изображение с помощью колесика мыши, затем выполните панорамирование. Обратите внимание, что при панорамировании масштаб изображения не изменяется.
- 5 Прежде чем перейти к следующему уроку, нажмите кнопку  > "Закреть", чтобы закрыть чертеж. Не сохраняйте чертеж.

Использование Инфоцентра для получения интерактивной справки и другой информации

Панель "Инфоцентр" расположена в верхнем правом углу окна AutoCAD.



Инфоцентр можно использовать для поиска информации в разнообразных источниках с помощью одного запроса. Также можно легко получить доступ к обновлениям продукта и к уведомлениям.

Инфоцентр можно использовать в следующих целях:

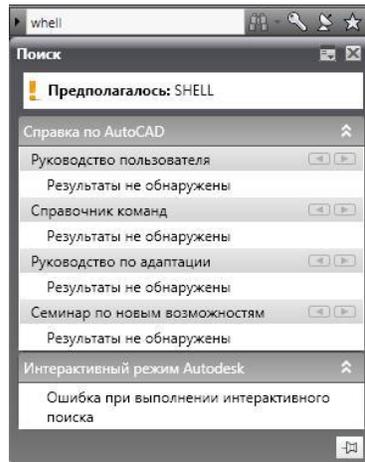
- для поиска информации по ключевым словам или фразам;
- для доступа к услугам подписки из панели "Центр подписки";
- для доступа к обновлениям программных продуктов и уведомлениям из панели "Коммуникационный центр";
- для доступа к сохраненным разделам в панели "Избранное";
- для доступа к разделам интерактивной справки.

Поиск с помощью Инфоцентра

- 1 В текстовом поле Инфоцентра "Поиск" введите CAD. Щелкните раскрывающийся список рядом со значком поиска (бинокль) и выберите область поиска. Отображаются результаты поиска по слову "CAD" для выбранной области поиска.

2 Для просмотра результатов поиска нажмите кнопку "Далее" (стрелка "вправо") или "Назад" (стрелка "влево").

При ошибочном вводе слова в панели отображается подсказка с предлагаемыми вариантами написания.

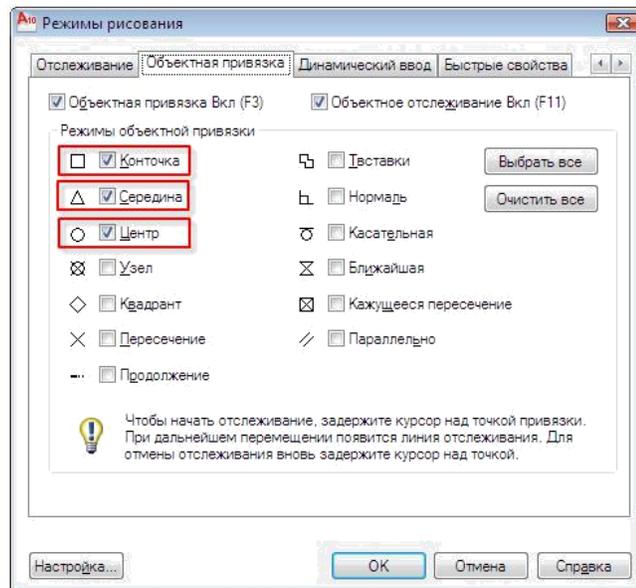


Задание режимов рисования

- 1 Щелкните правой кнопкой мыши значок "Объектная привязка", расположенный в строке состояния в нижней части окна, и выберите "Настройки".



- 2 Открывается диалоговое окно "Режимы рисования". Убедитесь, что на вкладке "Объектная привязка" установлен флажок "Объектная привязка вкл.". Для задания режимов объектной привязки установите флажки "Контточка", "Середина точка" и "Центр". Убедитесь, что другие флажки не установлены. Нажмите кнопку "ОК".

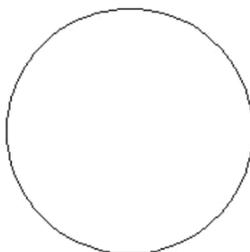
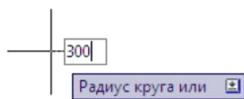


Построение окружности

"Рисование" > раскрывающееся меню "Круг" > "Центр, радиус"



- 1 Установите курсор-перекрестье в области рисования.
- 2 Щелчком левой кнопкой мыши задайте центр окружности.
- 3 Отображается поле динамического ввода с запросом "Радиус круга". Введите с клавиатуры значение 3.25 и нажмите ENTER для создания окружности с радиусом 3,25.



Применение образца штриховки

Замкнутый контур можно заполнить стандартным образцом штриховки, образцом из простых линий с использованием текущего типа линий или пользовательским образцом штриховки. Один из типов штриховки, при использовании которого область закрашивается сплошным цветом, называется заливкой.

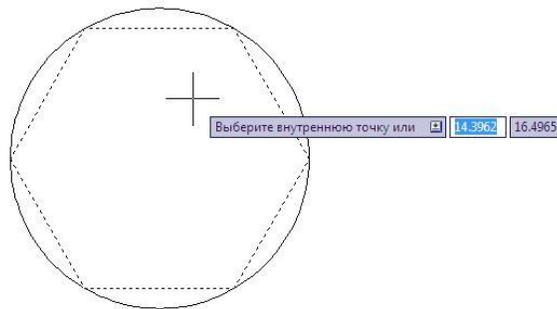
Использование стандартного образца штриховки

- Щелкните на ленте вкладку "Главная" > панель "Рисование" > "Штриховка".



- В диалоговом окне "Штриховка и градиент" выберите "Добавить: точки выбора".

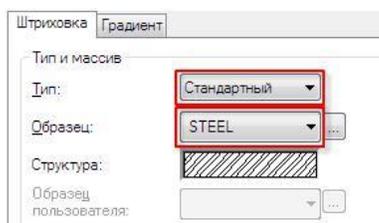
- Щелчком *внутри* многоугольника выберите его. Сторону многоугольника выбирать не следует. Нажмите клавишу ENTER.



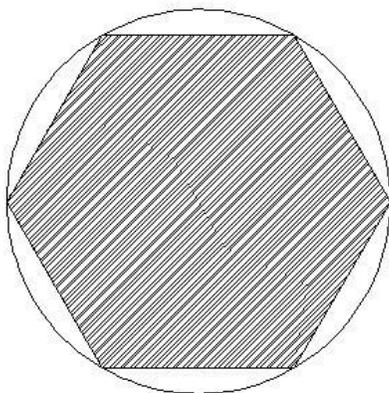
3 В диалоговом окне "Штриховка и градиент" на вкладке "Штриховка" в группе "Тип и массив" с помощью раскрывающегося меню задайте следующие параметры:

Тип: Стандартный

Образец: Сталь



4 Нажмите "ОК". Многоугольник заполняется выбранным образцом штриховки.



6 Щелкните  ► "Сохранить", затем закройте чертеж.

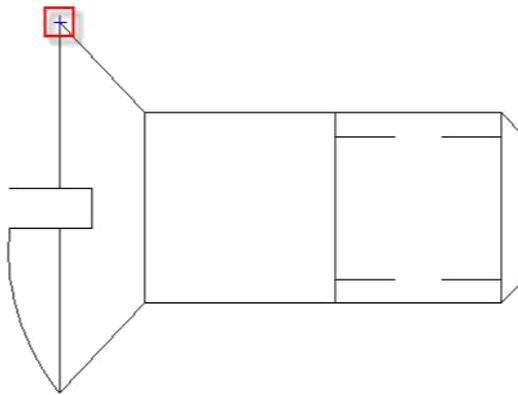
Построение дуги

Построение дуги

- 6 Щелкните на ленте вкладку "Главная" ► панель "Рисование" ► "3 точки".

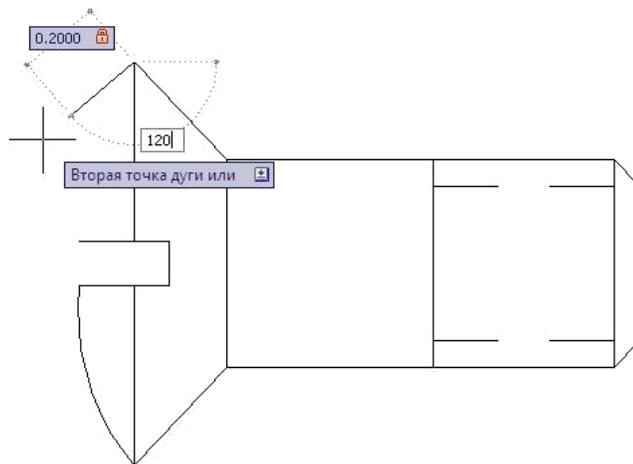


- 6 В поле динамического ввода курсора отображается сообщение "конечная точка".

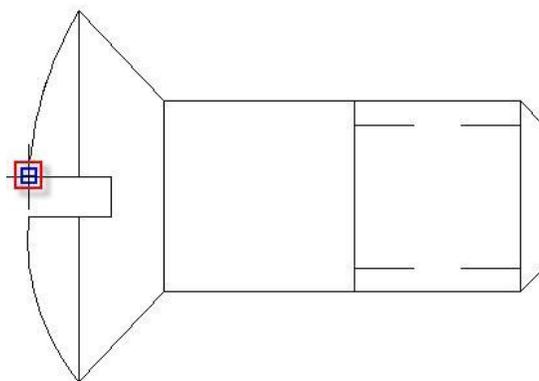


ПРИМЕЧАНИЕ Если цветной квадрат не отображается, убедитесь, что режимы рисования заданы так, как описано в уроке 1.

- 7 Щелкните вблизи центра цветного квадрата. Указанная точка определяет первую точку дуги. Квадрат является визуальной подсказкой, помогающей выполнить привязку первой точки дуги к конечной точки существующего отрезка. Данная функция позволяет создавать точные чертежи.
- 1 Переместите курсор в сторону от первой точки. Затем введите местоположение второй точки, а также укажите угол дуги. Эти значения вводятся с использованием динамического ввода.
- 2 Сначала с клавиатуры введите 0.2000 для задания второй точки дуги. Обратите внимание, что это значение отражается в поле динамического ввода. Не нажимайте клавишу ENTER. Вместо этого нажмите клавишу TAB.
- 3 Нажатие клавиши TAB позволяет перейти в поле динамического ввода, где отображается запрос на задание угла дуги. Введите 120. Нажмите ENTER для задания второй точки дуги.



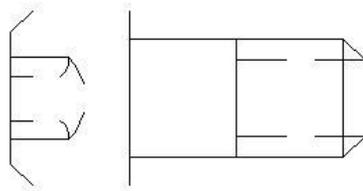
- 1 Переместите курсор в точку, показанную на следующем рисунке. Когда появится цветной квадрат и в поле динамического ввода курсора будет выведено сообщение "конечная точка", щелкните в центре квадрата для завершения построения дуги.



- 12 Щелкните  ➤ "Сохранить", затем закройте чертеж.

Построение полилинии

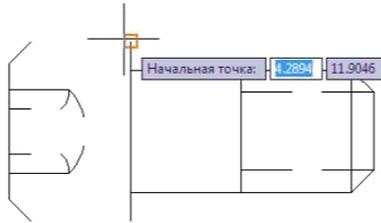
На этом уроке будет показано, как можно создать объект - *полилинию*. Полилиния представляет собой последовательность сегментов, рассматриваемых как единый объект. Сегменты, из которых состоит полилиния, могут быть прямолинейными или криволинейными. На этом уроке будет показано, как вычерчивается полилиния с прямыми сегментами.



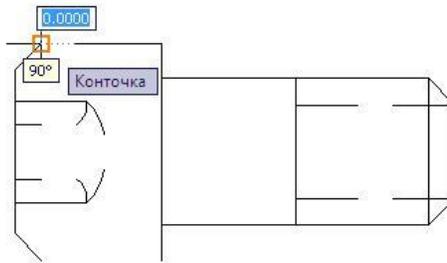
2 Щелкните на ленте вкладку "Главная" ➤ панель "Рисование" ➤

"Полилиния". 

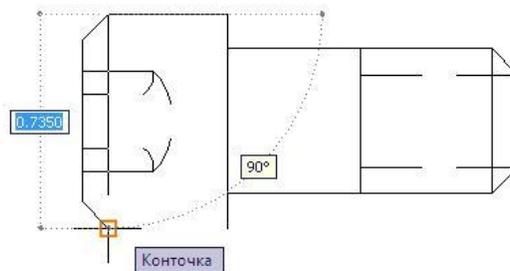
5 Наведите курсор на чертеж и переместите его в положение, указанное окрашенным квадратом на следующем рисунке.



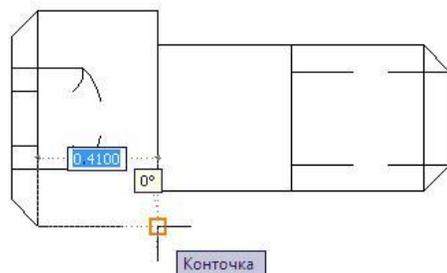
- Щелкните вблизи центра цветного квадрата. Указанная точка определяет первую точку полилинии.
- Переместите курсор в положение, указанное на следующем рисунке цветным квадратом:



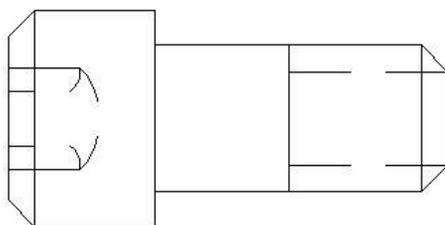
- Щелкните вблизи центра цветного квадрата. Построен первый отрезок полилинии.
- Переместите курсор в положение, указанное на следующем рисунке цветным квадратом:



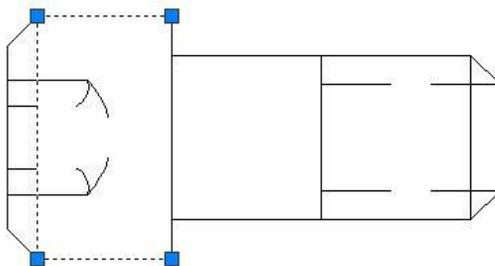
- 1 Щелкните вблизи центра цветного квадрата.
Построен второй отрезок полилинии.
- 1 Переместите курсор в положение, указанное на следующем рисунке цветным квадратом:



- 2 Щелкните вблизи центра цветного квадрата.
Построен третий отрезок полилинии.
- 2 Нажмите клавишу ENTER для завершения построения полилинии.



- 14 Щелчком на одном из сегментов выберите полилинию.
Обратите внимание, что при выборе одного линейного сегмента выбирается вся полилиния.



- 15 Щелкните  ► "Сохранить", затем закройте чертеж.

Задание параметров сетки и шаговой привязки

Такие компоненты, как сетка и шаговая привязка, позволяют настроить структуру, которую можно использовать в качестве основы для построений.

- Сетка представляет собой прямоугольную комбинацию точек, занимающую область в заданных пользователем границах сетки. Сетка заменяет листок бумаги, расчерченный в клетку, который подкладывают под чертеж для облегчения построений. Сетка позволяет выравнивать объекты и оценивать расстояние между ними. Сетка не выводится на печать.
- Функция привязки "шаг" ограничивает перемещение курсора в форме перекрестья заданным пользователем интервалом. При включенном режиме шаговой привязки курсор фиксируется на невидимой сетке (как бы прилипает к ней). Режим шаговой привязки удобен для точного задания местоположения точек с помощью курсора.

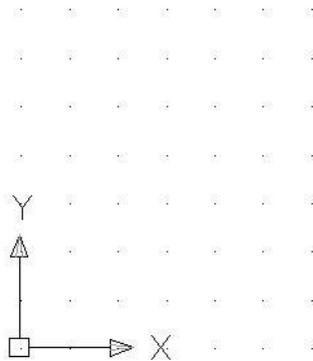
Отображение сетки

В строке состояния нажмите кнопку "Отображение сетки" для вывода ее на экран.



ПРИМЕЧАНИЕ Если кнопка включения сетки не отображается, щелкните правой кнопкой мыши в строке состояния и выберите пункт "Переключатели режимов" > "Сетка (F7)".

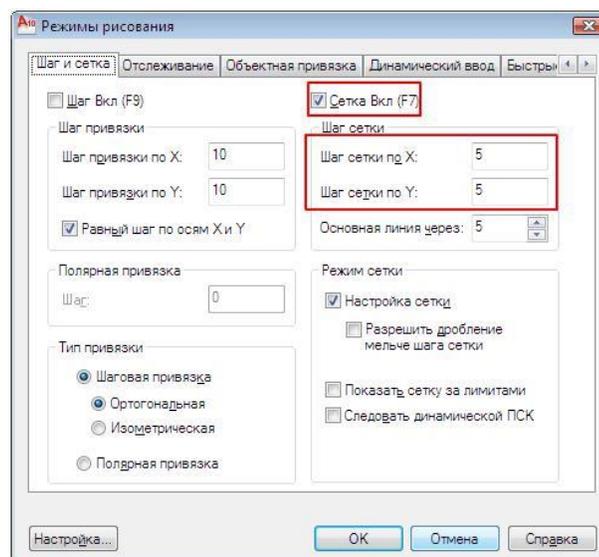
В области рисования отображается сетка. Обратите внимание на то, что она показана в ограниченной области, в лимитах сетки.



Настройка шага сетки

В строке состояния щелкните правой кнопкой мыши на кнопке "Сетка". Выберите "Режимы".

В диалоговом окне "Режимы рисования", на вкладке "Шаг и сетка" убедитесь в том, что установлен флажок "Сетка Вкл (F7)"

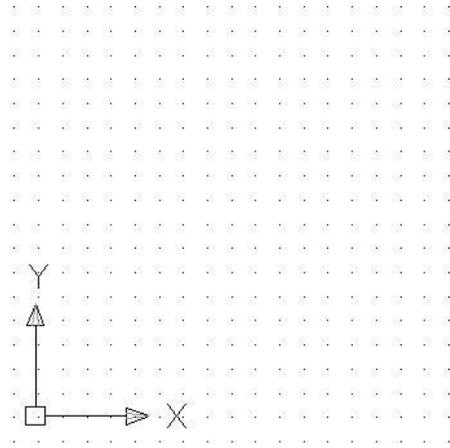


В разделе "Шаг сетки" выполните следующие действия:

В поле "Шаг сетки по X" введите 0.0500, чтобы задать шаг сетки по горизонтали в принятых единицах.

В поле "Шаг сетки по Y" введите 0.0500, чтобы задать шаг сетки по вертикали в принятых единицах.

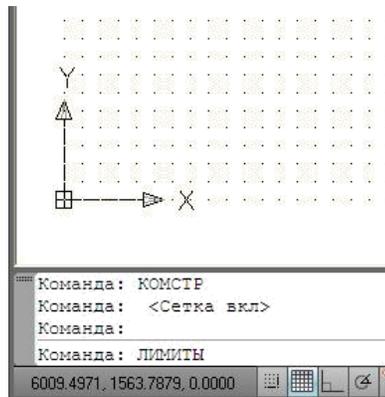
Нажмите "ОК".



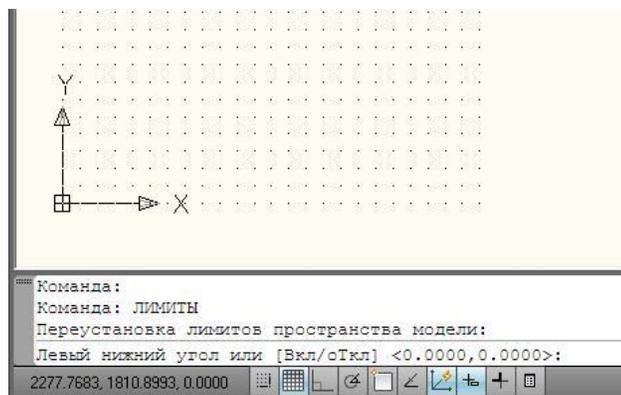
Расстояние между точками сетки становится меньше.

Настройка лимитов сетки

Область чертежа, на которой отображается сетка, определяется командой ЛИМИТЫ. Введите слово ЛИМИТЫ в командной строке и нажмите клавишу Enter.



В командной строке нажмите Enter для подтверждения принятых по умолчанию значений координат (0,0), которые определяют левый нижний угол лимитов чертежа.



В командной строке введите 480,360 и нажмите Enter, чтобы задать верхний правый угол лимитов сетки. Зона отображения сетки увеличивается и занимает большую площадь в области рисования.

Включение режима шаговой привязки

В строке состояния нажмите кнопку "Шаговая привязка" для включения соответствующего режима.



ПРИМЕЧАНИЕ Если кнопка шаговой привязки не отображается, щелкните правой кнопкой мыши в строке состояния и выберите пункт "Переключатели режимов" ► "Шаг(F9)".

При включенном режиме шаговой привязки перемещайте курсор в области рисования. Обратите внимание на то, что выполняется привязка курсора к точкам, расположенным с равным интервалом в области рисования.

Настройка шаговой привязки

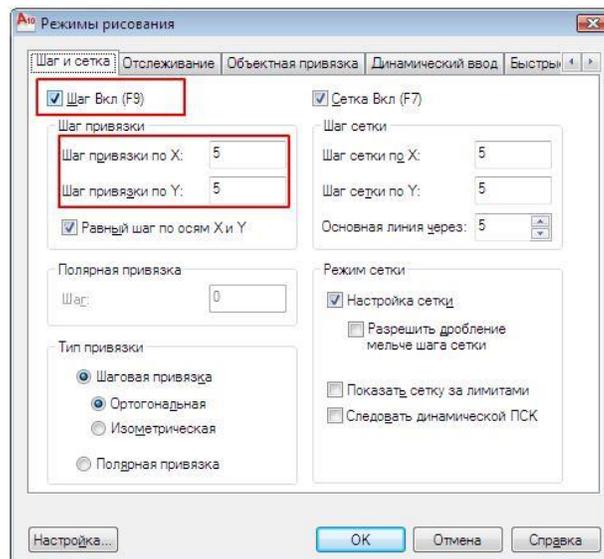
В строке состояния щелкните правой кнопкой мыши на кнопке "Шаговая привязка" и выберите пункт "Режимы".

В диалоговом окне "Режимы рисования", на вкладке "Шаг и сетка" убедитесь в том, что установлен флажок "Шаг Вкл (F9)"

В разделе "Шаг привязки" выполните следующие действия:

В поле "Шаг привязки по X" введите 0.500, чтобы задать значение шага привязки по горизонтали в принятых единицах.

В поле "Шаг привязки по Y" введите 0.500, чтобы задать значение шага привязки по вертикали в принятых единицах.



Нажмите "ОК".

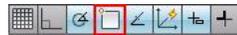
Точная привязка к точкам объектов

Использование объектной привязки представляет собой наилучший способ задания точного местоположения на объекте без необходимости ввода координат. Например, можно использовать объектную привязку для построения отрезка до точного центра окружности, до конечной точки другого линейного сегмента или по касательной к дуге.

Объектную привязку можно включать во время любого запроса указания точек. По умолчанию при перемещении курсора над объектной привязкой на объекте отображаются маркер и подсказка. Для объектной привязки имеются специальные средства визуализации, называемые автопривязкой. Эти средства повышают наглядность и эффективность использования объектной привязки.

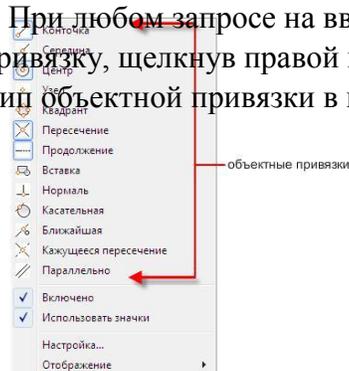
Отображение и активизация объектной привязки

2 В строке состояния нажмите кнопку "Объектная привязка" для включения соответствующего режима.

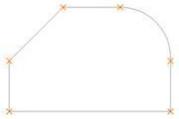
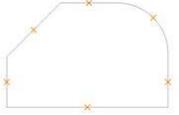
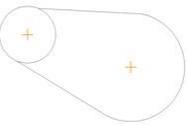


ПРИМЕЧАНИЕ Если кнопка "Объектная привязка" не отображается, щелкните правой кнопкой мыши в строке состояния и выберите пункт "Переключатели режимов" > "Объектная привязка (F3)".

6 При любом запросе на ввод точки можно задать единичную объектную привязку, щелкнув правой кнопкой мыши при нажатой клавише Shift и выбрав тип объектной привязки в меню "Объектная привязка".



Часто используемые типы объектной привязки приведены в следующей таблице.

Объектная привязка	Назначение привязки
<p>Конточка</p>  	Конечные точки объектов
<p>Середина</p>  	Точки на середине сегментов объектов
<p>Пересечение</p>  	Точки пересечения линий объектов или, для единичных объектных привязок, места, в которых эти линии пересеклись бы при удлинении объектов
<p>Центр</p>  	Центры окружностей, дуг или эллипсов
<p>Квадрант</p> 	Квадранты окружностей, дуг или эллипсов, то есть точки пересечения этих примитивов с центровыми осями, параллельными осям текущей системы

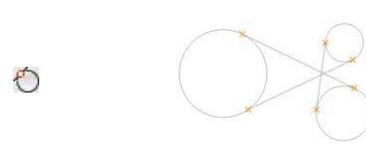
ыкоор-
динат.



Нормаль



Точки на объектах,
располо-
женныенаперпендик
уляре,
опущенном на
объекты из
последней заданной
точки

Объектная привязка	Назначение привязки
<p data-bbox="527 525 649 567">Касание</p> 	<p data-bbox="1071 493 1360 819">Точка на окружности или дуге, которая, при соединении с последней точкой, образует касательную к объекту</p>

Задание объектной привязки

Постройте отрезок в центре области рисования.

На ленте выберите вкладку "Главная" ► панель "Рисование" ► раскрывающийся список "Круг" ► "Центр, радиус". 

При нажатой клавише Shift щелкните правой кнопкой мыши в области рисования и выберите в меню "Объектная привязка" пункт "Середина". Эта операция позволяет указать точку центра окружности.

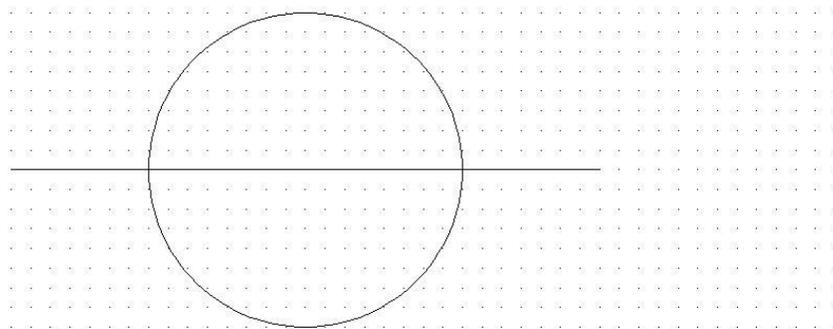
Наведите курсор на середину отрезка.

При включенном режиме автопривязки курсор должен автоматически фиксироваться в ближайшем местоположении объектной привязки "середина". Кроме того, должны выводиться маркер и подсказка.



Укажите точку в области рисования для привязки к середине отрезка. Задайте значение радиуса окружности:

2.



Задание действующих типов объектной привязки

Чтобы использовать один и тот же тип объектной привязки неоднократно, можно назначить такой режим автопривязки постоянно действующим. Выбранная привязка останется активной, пока не будет отключена. В качестве действующей объектной привязки можно задать несколько типов объектной привязки, например "Конточка" и "Центр". Выполняемые объектные привязки можно включать и отключать в строке состояния.

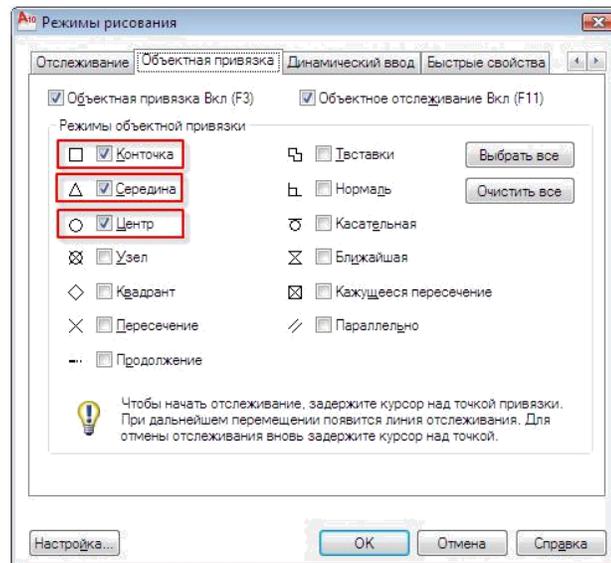
4 В строке состояния щелкните правой кнопкой мыши на кнопке "Объектная привязка" и выберите пункт "Режимы".

5 В диалоговом окне "Режимы рисования", на вкладке "Объектная привязка" нажмите "Очистить все", а затем установите следующие флажки:

"Конточка";

"Середина";

"Центр".



6

Нажмите "OK".

7

Постройте несколько отрезков и окружностей с использованием объектной привязки для точного задания местоположения точек.

Задание углов и расстояний

При построении отрезков или перемещении объектов можно использовать полярное отслеживание, чтобы ограничить перемещение курсора с учетом заданного  углового шага (значение по умолчанию - 90 градусов).

Например, включив полярное отслеживание перед началом построений, можно создать ряд перпендикулярных отрезков. Так как отрезки связаны зависимостью

1 горизонтальной и вертикальной осями, можно выполнять построения быстрее, поскольку перпендикулярность линий гарантируется.

Отображение и активизация полярного отслеживания

2 В строке состояния нажмите кнопку "Полярное отслеживание" для включения соответствующего режима.

ПРИМЕЧАНИЕ Если кнопка "Полярное отслеживание" не отображается, щелкните правой кнопкой мыши в строке состояния и выберите пункт "Переключатели режимов" > "Полярное отслеживание (F10)".

Использование полярного отслеживания

1 В строке состояния щелкните правой кнопкой мыши на кнопке "Полярное отслеживание" и выберите пункт "Режимы".

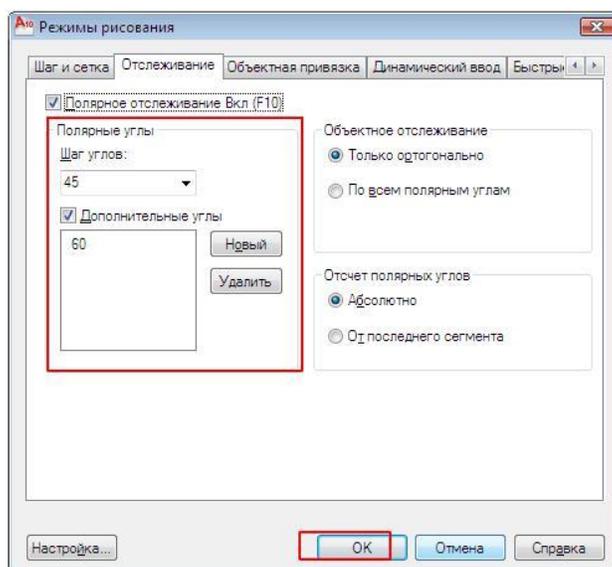
1 В диалоговом окне "Режимы рисования", на вкладке "Полярное отслеживание" убедитесь в том, что установлен флажок "Полярное отслеживание Вкл (F10)".

2

В разделе "Полярные углы" выполните следующие действия.

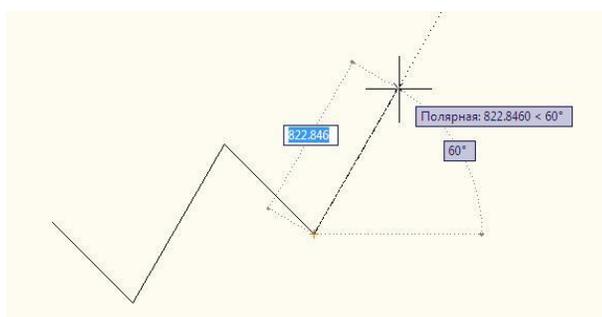
В раскрывающемся списке "Шаг углов" выберите значение "45".

Нажмите кнопку "Новый" и для добавления нового значения полярного угла введите 60. Дополнительные углы задаются абсолютным значением, а не как приращение.



1

Нажмите "ОК". Постройте несколько отрезков под углом 45 и 60 градусов друг к другу. Обратите внимание, что выполняется привязка отрезков к направлениям, заданным углами. Удалите отрезки перед переходом к выполнению следующего упражнения.



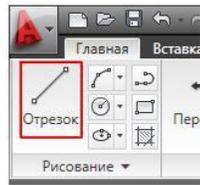
Построение объектов с помощью полярной привязки

Чтобы задать значения координат, можно использовать непосредственный ввод расстояния. Для этого перемещением курсора следует задать направление, а затем ввести значение от предыдущей точки. При включенном полярном отслеживании использование непосредственного ввода расстояния позволяет строить отрезки заданной длины под заранее определенным углом.

1 Убедитесь в том, что режим полярного отслеживания включен.

2 Выберите вкладку "Главная" ► панель "Рисование" ►

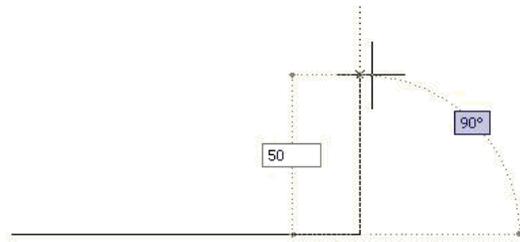
"Отрезок". 



5 Щелчком в области рисования для задайте первую точку и переместите курсор вправо (0 градусов).

3 Введите 10 в поле динамического ввода, чтобы задать расстояние, и нажмите клавишу Enter.

4 Переместите курсор вверх под углом 90 градусов, введите 5 для задания следующей точки и нажмите клавишу Enter.

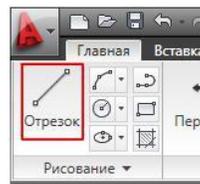


8 Повторите описанное действие еще несколько раз и нажмите Enter для завершения операции. Удалите отрезки, прежде чем перейти к выполнению следующего упражнения.

Разовое задание угла отслеживания

Если значение угла не будет использоваться часто, можно ввести переопределение угла. Например, если начинается построение отрезка в точке с координатами (5,5) и необходимо, чтобы отрезок проходил под углом 10 градусов и имел длину 20 единиц, требуется выполнить следующие действия.

1 Выберите вкладку "Главная" > панель "Рисование" > "Отрезок".



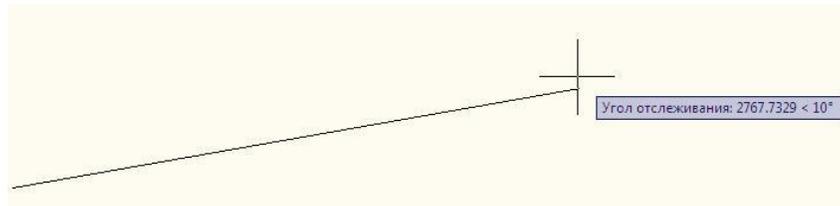
9
клавишу

В поле запроса введите #5,5 для задания первой точки и нажмите Enter.

■
нажмите

В поле запроса введите <10, чтобы задать переопределение угла, и

Enter.



1 Переместите курсор в требуемом направлении.

2 |

15 Введите 20 и нажмите клавишу Enter. Построен отрезок длиной
20 единиц под углом 10 градусов.

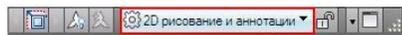
Смена рабочего пространства

Рабочее пространство представляет собой конфигурацию окна программы AutoCAD, которое состоит из закрепляемых окон, меню, панелей инструментов и других элементов пользовательского интерфейса. Можно использовать стандартные рабочие пространства или создавать собственные.

При использовании рабочего пространства отображаются только необходимые для конкретной задачи меню, панели инструментов, вкладки ленты и палитры.

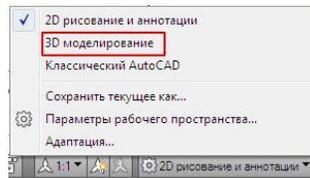
Смена рабочего пространства

- 1 В нижней части окна расположена строка состояния. Ниже показан значок рабочего пространства в строке состояния.



ПРИМЕЧАНИЕ Если этот значок не отображается, щелкните правой кнопкой мыши в пустой области строки состояния и выберите "Рабочие пространства". Если значок по-прежнему не отображается, необходимо развернуть окно AutoCAD во весь экран.

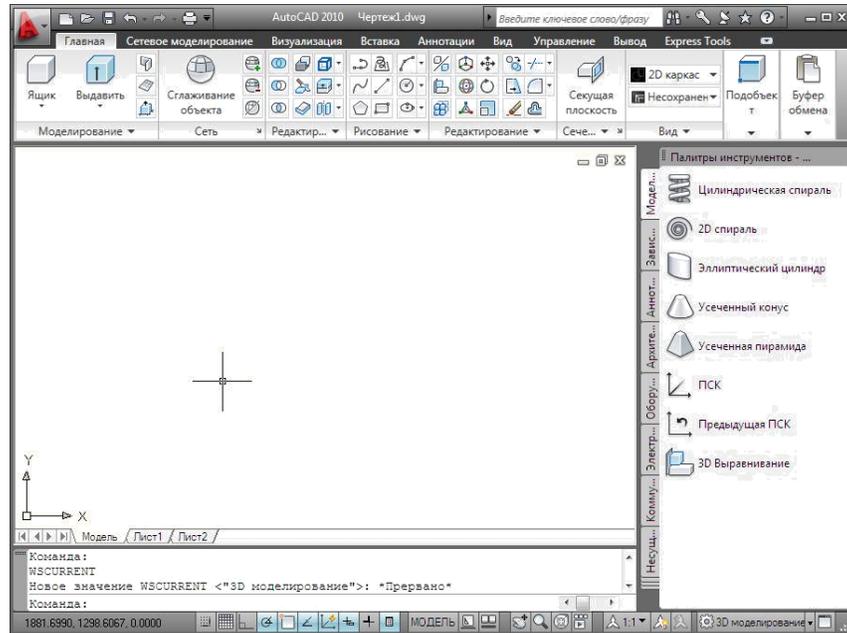
- 4 Щелкните стрелку рядом с раскрывающимся списком, чтобы вывести на экран меню стандартных рабочих пространств.
- 5 В меню "Рабочие пространства" выберите "3D моделирование".



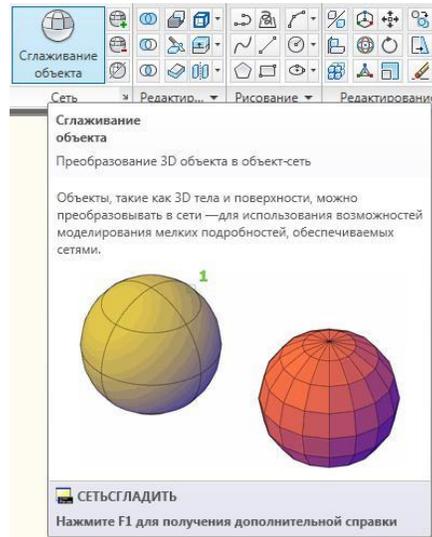
- 6 Отображается рабочее пространство "3D моделирование". Значок рабочего пространства в строке состояния указывает, что активным является рабочее пространство "3D моделирование". В этом рабочем пространстве можно использовать различные команды и инструменты, необходимые для создания 3D чертежей. Например,

обратите внимание на расположенные на ленте команды
3D моделирования.

ПРИМЕЧАНИЕ Выбранное рабочее пространство сохраняется при закрытии
и повторном открытии AutoCAD. Для возврата в рабочее пространство "2D
рисование и аннотации" его необходимо выбрать.



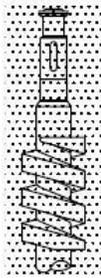
2 Для вызова справочной информации о командах на ленте установите курсор на одной из команд и задержите его на ней в течение секунды. На рисунке показана справочная информация для команды "Сглаживание объекта".



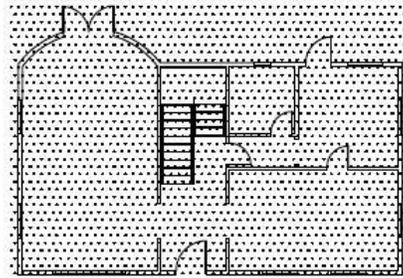
3 Для перехода в рабочее пространство "2D
рисование и аннотации" воспользуйтесь значком
рабочих пространств в строке состояния.

Выбор единиц чертежа

Прежде чем приступить к работе с чертежом, необходимо решить, какие единицы чертежа будут использоваться. Расстояния в AutoCAD измеряются в единицах чертежа. На чертеже одна единица чертежа может соответствовать одному дюйму, одному миллиметру, одному метру или одной миле.



Вал
1 единица = 1 мм
(шаг сетки = 2 мм)



План офиса
1 единица = 1 дюйм
(шаг сетки = 12 дюймов)

Прежде чем приступить к черчению, выберите единицу чертежа. В AutoCAD не предусмотрена настройка, определяющая длину единицы чертежа.

После выбора единицы чертежа необходимо задать формат единиц чертежа.

Параметры формата для линейных единиц

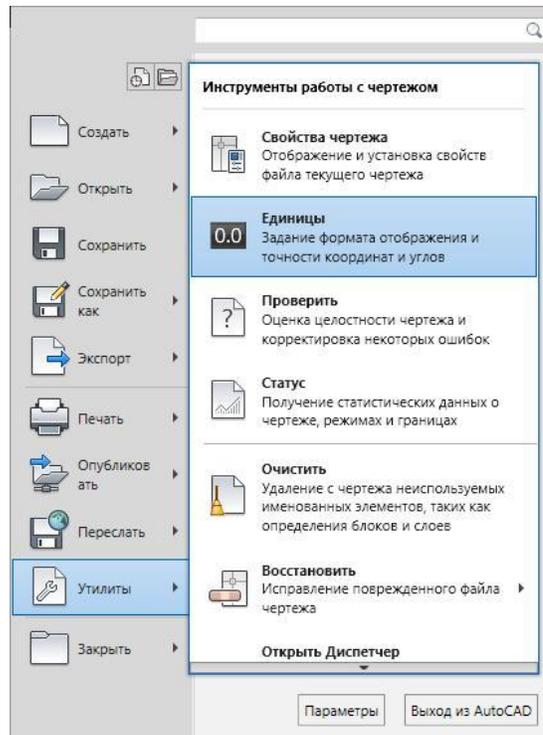
- 1 Архитектурные. Значение длины, равное 15,5 единицам, отображается как 1'-3 1/2"
- 1 Десятичные. Значение длины, равное 15,5 единицам, отображается как 15.5000
- 1 Инженерные. Значение длины, равное 15,5 единицам, отображается как 1'-3.5"

Например, инженерам-механикам, использующим в своей работе данные в миллиметрах, необходимо для линейных единиц задать десятичный формат. Архитектор, использующий значения в футах и дюймах, должен задать архитектурный формат.

Формат единиц чертежа определяет только стиль отображения единиц чертежа на экране, например при отображении координат и значений в диалоговых окнах и запросах.

Задание формата единиц чертежа

3 Щелкните  > "Утилиты" > "Единицы".



4 В диалоговом окне "Единицы чертежа" в области "Длина" выберите следующие значения:

Тип: Архитектурные

Точность: 0'-0 1/8''

А10 Единицы чертежа

Линейные	Угловые
Формат: Архитектурные	Формат: Десятичные градусы
Точность: 0-0 1/8"	Точность: 0
	<input type="checkbox"/> По часовой стрелке

Масштаб вставки

Единицы для изменения вставленных элементов:
Дюймы

Пример
1 1/2", 2", 0"
3" < 45,0"

Освещение

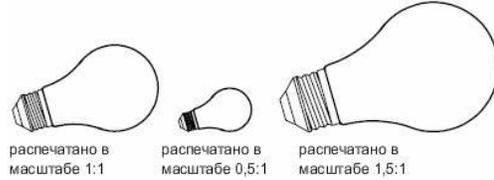
Единицы задания интенсивности освещения:
Международные

OK Отмена Направление... Справка

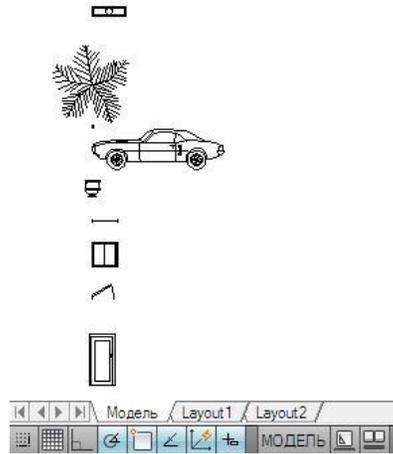
Задание масштаба печати

На этом примере будет показано, как задается масштаб для вывода чертежа на печать.

Например, величина одной единицы чертежа, как правило, соответствует одному миллиметру или одному метру в метрических чертежах, в то время как британской системе единиц она обычно соответствует одному дюйму или одному футу.



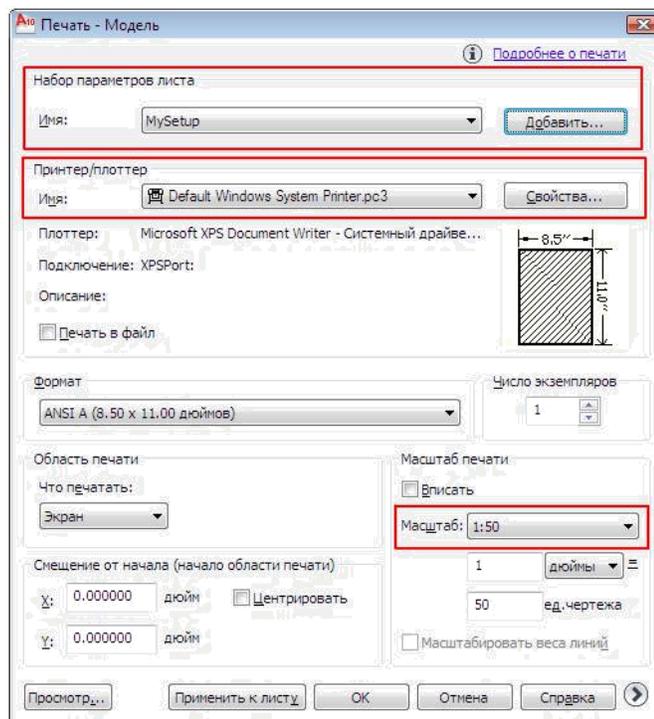
Печать с использованием нестандартного масштаба



4 Убедитесь, что в нижней части окна чертежа выбрана вкладка "Модель".

6 Щелкните  > "Печать" > "Печать"  .

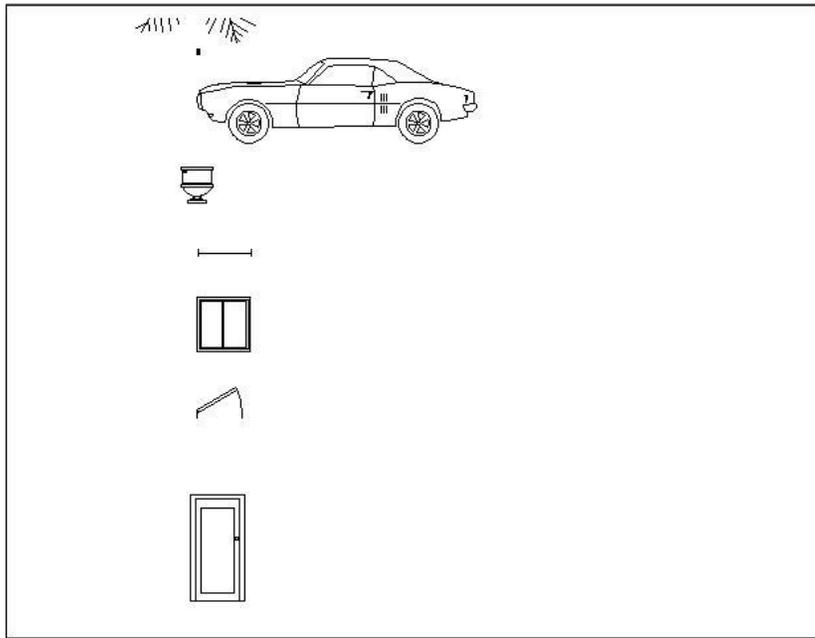
1 В диалоговом окне "Печать" в разделе "Параметры листа" нажмите кнопку "Добавить".



- 3 В диалоговом окне "Добавление набора параметров листа" введите MySetup. Нажмите "ОК".
- 4 В разделе "Принтер/плоттер" в списке "Имя" выберите принтер, на который будет выведен текущий лист.
- 5 В разделе "Масштаб печати" в раскрывающемся списке "Масштаб" выберите значение 1:50.

ПРИМЕЧАНИЕ Если установлен флажок "Вписать", список "Масштаб" не активен.

- 6 Нажмите кнопку "Просмотр". Если открывается диалоговое окно "Подтверждение масштаба печати", выберите "Продолжить". В окне предварительного просмотра чертеж отображается с выбранным масштабом 1:50.



- 7 В окне предварительного просмотра щелкните "Печать"  для вывода чертежа на печать.
- 7 Щелкните  для закрытия окна предварительного просмотра и нажмите "Отмена" для закрытия диалогового окна "Печать".

Выбор масштаба при печати чертежа в соответствии с форматом листа бумаги

- 1 Щелкните  > "Печать" > "Печать". 
- 2 В диалоговом окне "Печать" в группе "Масштаб печати" установите флажок "Вписать".

Формат: ANSI A (8.50 x 11.00 дюймов) Число экземпляров: 1

Область печати
Что печатать: Экран

Смещение от начала (начало области печати)
X: 0.000000 дюйм Центрировать
Y: 0.000000 дюйм

Масштаб печати
 Вписать
Масштаб: Польз. 1 дюймы 130.3 ед. чертежа
 Масштабировать веса линий

Полученный масштаб будет вычислен автоматически. В соответствующем поле отображается соотношение между единицами печати и единицами чертежа, представляющее собой нестандартный масштаб.

1 Нажмите "ОК" для вывода чертежа на печать.

СОВЕТ Чтобы увидеть, как будет выглядеть распечатанный чертеж, в диалоговом окне "Печать" щелкните "Просмотр".

Основные сведения о пространствах модели и листа

Существует две отдельных рабочих среды или "пространства", в которых можно создавать объекты на чертеже: *пространство модели* и *пространство листа*.

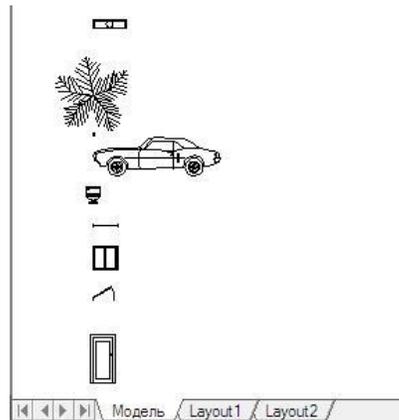
Для перехода из пространства модели в пространство листа и обратно используются вкладки в нижней части области рисования. Вкладка "Модель" служит для работы в пространстве модели, одна или несколько вкладок "Лист" - для работы в пространстве листа.

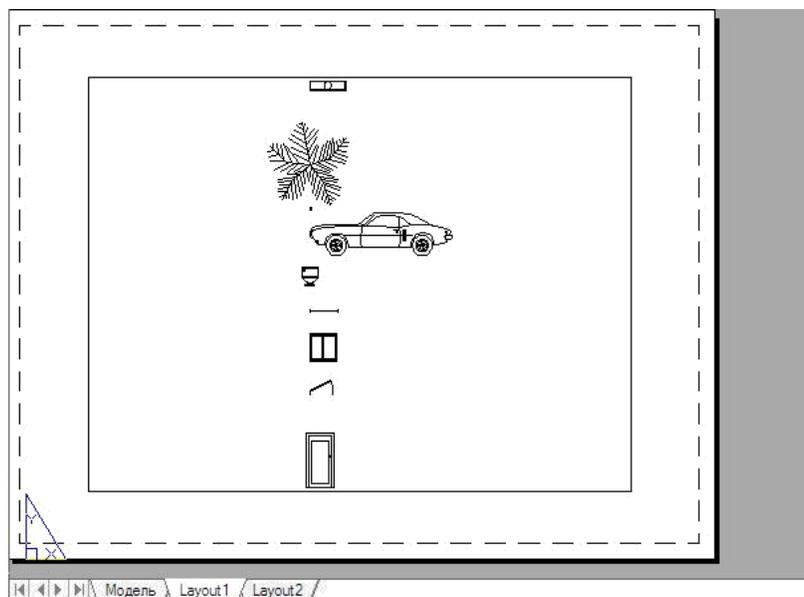
ПРИМЕЧАНИЕ Эти вкладки могут быть скрыты и отображаться в виде кнопок в строке состояния.

В пространстве модели модель создается в масштабе 1:1. В пространстве листа можно создать один или несколько *видовых экранов листа*, задавать размеры, создавать примечания и основную надпись, представляющую лист чертежа.

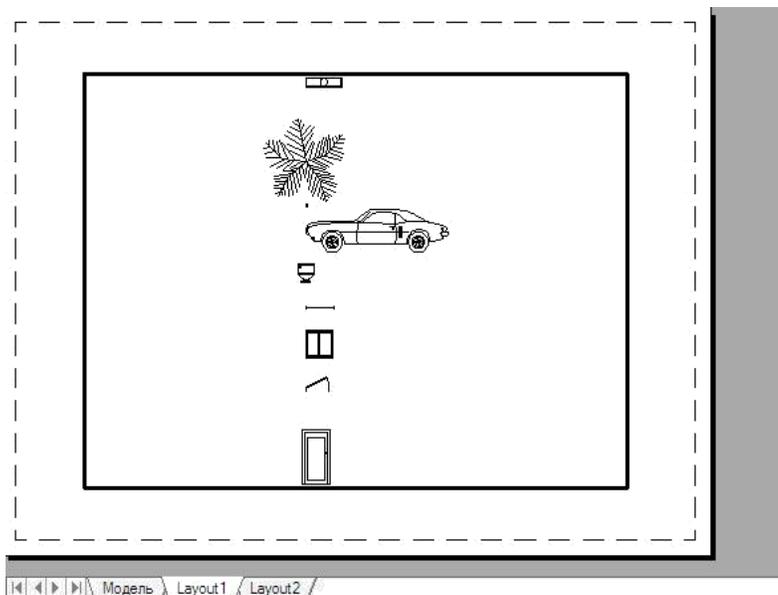
Переход между пространством модели и пространством листа

- 1 Щелкните вкладку "Лист1" для перехода в пространство листа. Пространство листа является активным, если в нижнем углу отображается значок (показан ниже синим цветом). Если пространство листа не активно, дважды щелкните левой кнопкой мыши в пустой области вне прямоугольника.





- 5 Можно переходить из пространства модели в пространство листа и обратно. Для перехода в пространство модели дважды щелкните левой кнопкой мыши в середине прямоугольника. Обратите внимание, что граница внутренней прямоугольной области стала толще и исчез синий значок, указывающий на пространство листа.



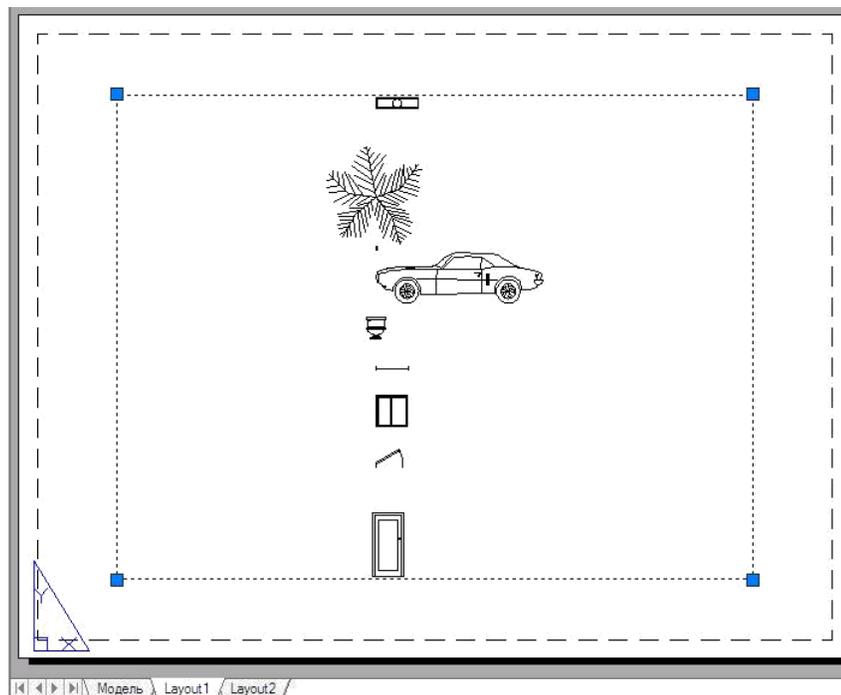
- 4 В пространстве модели прямоугольная область с утолщенной границей соответствует видовому экрану листа. Видовые экраны листов можно использовать для вызова пространства модели из пространства листа. Видовой экран листа представляет собой своего рода кадр изображения с "фотографией" модели в пространстве модели.
- 5 Попробуйте несколько раз перейти из пространства модели в пространство листа и обратно. Для этого необходимо дважды щелкнуть внутри видового экрана (для перехода в пространство модели) и вне видового экрана в пустой области (для перехода в пространство листа).

ПРИМЕЧАНИЕ Если перейти в пространство модели не удастся, щелкните кнопку с надписью "ЛИСТ" в строке состояния. Надпись на кнопке должна измениться на "МОДЕЛЬ".

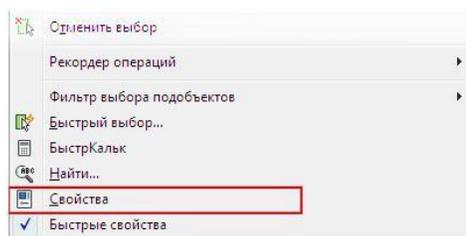
Задание масштаба видового экрана

Выберите вкладку "Лист1", расположенную справа от вкладки "Модель". Убедитесь, что активным является пространство листа.

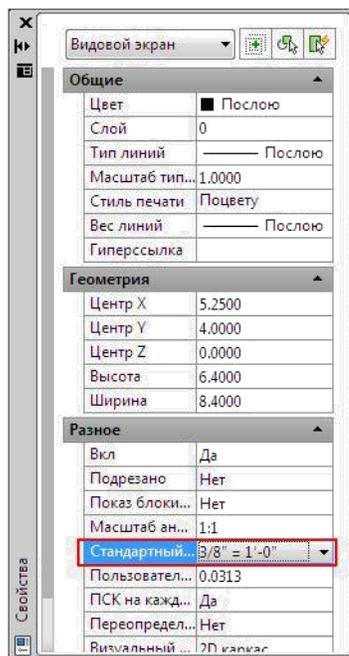
- Щелкните границу видового экрана.



2 Щелкните правой кнопкой мыши и выберите "Свойства".



16В палитре свойств выберите "Стандартный масштаб", затем выберите в списке $3/8'' = 1'-0''$.



Выбранный масштаб применяется к видовому экрану.

