

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

ИНФОРМАТИКА

*Методические рекомендации к лабораторным работам
для студентов направлений подготовки 09.03.01 «Информатика
и вычислительная техника» и 09.03.04 «Программная инженерия»
дневной формы обучения*



Могилев 2019

УДК 004
ББК 32.973
И 74

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления»
«13» ноября 2018 г., протокол № 5

Составитель канд. техн. наук, доц. В. П. Василевский

Рецензент канд. техн. наук, доц. И. В. Лесковец

Методические рекомендации к лабораторным работам предназначены для студентов направлений подготовки 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия» дневной формы обучения.

Учебно-методическое издание

ИНФОРМАТИКА

Ответственный за выпуск

А. И. Якимов

Технический редактор

А. А. Подошевка

Компьютерная верстка

Е. С. Лустенкова

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.

Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 31 экз. Заказ №

Издатель и полиграфическое исполнение:

Межгосударственное образовательное учреждение высшего образования

«Белорусско-Российский университет».

Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий

№ 1/156 от 24.01.2014.

Пр. Мира, 43, 212000, Могилев.

© Белорусско-Российский
университет, 2019



Содержание

Введение.....	4
1 Текстовый процессор Microsoft Word.....	5
1.1 Лабораторная работа № 1. Основные навыки работы. Редактирование и форматирование текста.....	5
1.2 Лабораторная работа № 2. Создание, редактирование и форматирование таблиц.....	7
1.3 Лабораторная работа № 3. Использование слияния для создания однотипных документов.....	8
2 Табличный процессор Microsoft Excel.....	11
2.1 Лабораторная работа № 4. Ввод и редактирование данных.....	11
2.2 Лабораторная работа № 5. Использование встроенных функций.....	12
2.3 Лабораторная работа № 6. Диаграммы, графики.....	14
2.4 Лабораторная работа № 7. Решение уравнений и систем уравнений. Задачи оптимизации.....	17
2.5 Лабораторная работа № 8. Базы данных. Основные понятия.....	21
2.6 Лабораторная работа № 9. Базы данных. Методы обработки информации.....	23
2.7 Лабораторная работа № 10. Макросы.....	25
3 Основы языка VBA.....	27
3.1 Лабораторная работа № 11. Интерфейс редактора VBA. Программы линейной структуры.....	27
3.2 Лабораторная работа № 12. Операторы ветвления. Объекты Excel...	30
3.3 Лабораторная работа № 13. Операторы цикла.....	33
3.4 Лабораторная работа № 14. Пользовательские процедуры и функции.....	37
3.5 Лабораторная работа № 15. Переключатели.....	41
3.6 Лабораторная работа № 16. Списки.....	43
Список литературы.....	45



Введение

Цель дисциплины – получение знаний и навыков использования студентами специальностей «Информатика и вычислительная техника» и «Программная инженерия» средств, методов и приемов, развивающих практические навыки, которые востребуются сразу после включения в профессиональную деятельность.

Отличие информатики от других технических дисциплин, изучаемых в высшей школе, состоит в том, что ее предмет изучения меняется ускоренными темпами. При этом в среднем за полтора года удваиваются основные технические параметры аппаратных средств, за два-три года меняются поколения программного обеспечения.

В методических рекомендациях изложены основные понятия приложений MS Office Word, MS Office Excel и сведения, достаточные для выполнения лабораторных работ по этим приложениям.

Лабораторный практикум содержит также работы, в которых используется для выполнения заданий визуальная среда быстрого проектирования программ для Windows: Microsoft Visual Basic (VBA).

Основной формой защиты лабораторных работ является *собеседование*.

Для защиты лабораторных работ:

- предоставить преподавателю файлы с результатами выполненных заданий;
- ответить на вопросы по тематике защищаемой работы.

Созданные при выполнении заданий файлы сохраняются в рабочей папке студента.



1 Текстовый процессор Microsoft Word

1.1 Лабораторная работа № 1. Основные навыки работы. Редактирование и форматирование текста

Цель: освоить средства создания и обработки текстовых документов.

Общее название программных средств создания, редактирования и форматирования текстовых документов – *текстовые процессоры*. В настоящее время наибольшее распространение имеет текстовый процессор Microsoft Word. Различные версии MS Word отличаются, в основном, структурой и элементами пользовательского интерфейса.

Рабочее окно процессора имеет структуру, характерную для приложений ОС Windows. Его основные элементы управления: панель быстрого запуска, лента, на которой расположены вкладки, содержащие панели инструментов, рабочее поле и строка состояния.

Порядок работы в текстовом редакторе Word

Создание документа.

Новый документ создается командой меню **Файл**⇒**Создать**. В результате на экране появляется диалоговое окно **Создать**, в котором пользователю можно выбрать шаблон документа и нажать кнопку **ОК**.

Установка параметров страницы.

Перед набором текстового документа можно установить параметры страницы документа (по умолчанию используются параметры, заданные в шаблоне).

Для этого надо использовать вкладку **Разметка страницы**. Откроются группы **Параметры страницы**, **Фон страницы**, **Абзац** и **Упорядочить**.

Применяя имеющиеся в указанных группах команды, можно изменить параметры, заданные в шаблоне (по умолчанию используется шаблон **Normal.dot**).

Установка шрифта и его параметров.

Для этого используется вкладка **Главная**. Команды для изменения параметров шрифта размещены в группе **Шрифт**.

Если необходимо переустановить параметры шрифта в набранном фрагменте текста, то предварительно этот фрагмент выделяется.

Установка параметров форматирования абзаца.

Для достижения этой цели необходимо установить курсор ввода в нужный абзац, открыть контекстное меню и выбрать команду **Абзац**. В диалоговом окне установить требуемые значения в полях **Выравнивание**, **Отступ**, **Интервал**.

Эти же параметры можно менять, используя команды **Разметка страницы**⇒**Абзац** и **Главная** ⇒**Абзац**.



Сохранение документа.

Осуществляется выбором команды **Файл⇒Сохранить** (сохранить под старым именем) или **Файл⇒Сохранить как** (сохранить с новыми параметрами – указываются новое полное имя файла, папка, имя, тип и т. д.).

В результате на экране появляется диалоговое окно **Сохранение документа**. Это окно содержит поля, в которых задаются имя файла и маршрут к нему (его адрес) и указывается тип сохраняемого файла.

Выделение и правка текста.

Все основные операции с документами осуществляются над выделенными элементами. При этом выделенный участок документа изменяет свой цвет (если фон окна был, например, белым, а буквы черными, то после выделения фон станет черным, а буквы белыми).

Задание

1 Создайте новый документ (не вводя пока текста), имеющий следующие параметры страницы:

- размер бумаги – А4;
- верхнее поле – 2,5 см;
- нижнее поле – 2,5 см;
- левое поле – 2 см;
- правое поле – 1,5 см.

2 Сохраните созданный документ в своей папке (имя файла определите сами, например **Lab_W_1**).

3 Скопируйте в новый документ исходный текст (файл с текстом выдаст преподаватель).

4 Создайте элемент автотекста, включив в него содержание третьего абзаца текста.

5 Создайте элемент автозамены, который при наборе ваших инициалов вставляет в текст вашу полную фамилию (например, ввод ИАН заменяется текстом: Иванов Андрей Николаевич).

Контрольные вопросы

- 1 Основные элементы интерфейса MS Word 2010.
- 2 Базовые приемы работы в текстовом процессоре MS Word.
- 3 Методы создания нового документа.
- 4 Варианты сохранения документа.
- 5 Специальные способы ввода текста.



1.2 Лабораторная работа № 2. Создание, редактирование и форматирование таблиц

Цель: приобрести навыки работы с таблицами в текстовом процессоре MS Word.

Таблицы предназначены для упорядочивания данных и создания макетов страницы с последовательно расположенными столбцами текста или графики. Для работы с таблицами используется группа **Таблица**, размещенная во вкладке **Вставка**.

Наиболее оптимальный способ создания таблицы — использование команды **Вставить таблицу**.

После выполнения этой команды появляется диалоговое окно, в котором можно указать число столбцов и число строк, а также способ автоподбора ширины столбцов.

Для выполнения редактирования и форматирования созданной ранее таблицы необходимо установить курсор в любой ячейке таблицы, нажать левую клавишу мыши.

На ленте появляется пункт меню **Работа с таблицами**, содержащий вкладки **Конструктор** и **Макет**.

Используя команды, размещенные во вкладке **Конструктор**, можно выполнить операции по изменению стиля таблицы, параметров стиля, настройке границ выделенного фрагмента.

Используя команды, размещенные во вкладке **Макет**, можно добавлять (удалять) столбцы и строки, объединять, разделять и устанавливать размеры ячейки, а также выполнять операции по форматированию содержимого ячеек.

Задание

Создайте расписание занятий своей группы на три дня недели в соответствии с предложенным вариантом (таблица 1.1) и указаниями к выполнению задания.

Результаты сохраните в рабочей папке в файле «Lab_W_3».

Таблица 1.1

Номер варианта	День недели		
	1	Понедельник	Вторник
2	Пятница	Понедельник	Вторник
3	Понедельник	Вторник	Пятница
4	Четверг	Вторник	Среда
5	Понедельник	Среда	Пятница
6	Четверг	Пятница	Среда
7	Вторник	Среда	Четверг
8	Пятница	Вторник	Среда

Окончание таблицы 1.1

Номер варианта	День недели		
	9	Вторник	Среда
10	Пятница	Среда	Четверг
11	Вторник	Четверг	Понедельник
12	Вторник	Пятница	Понедельник
13	Среда	Четверг	Пятница
14	Пятница	Понедельник	Среда

Указания к выполнению задания

1 Вставьте таблицу, состоящую из семи столбцов и семи строк.

2 В первой строке объедините ячейки, принадлежащие соответственно второму и третьему, четвертому и пятому, шестому и седьмому столбцам, используя команду **Объединить ячейки**.

3 Аналогичным образом объедините ячейки первого столбца, принадлежащие первой и второй строкам.

4 Заполните расписание занятий в своей группе в соответствующие дни недели.

Контрольные вопросы

1 Назначение таблиц, создаваемых в текстовом процессоре MS Word.

2 Какие режимы работы используются для редактирования и форматирования таблиц?

3 Назначения команд, размещенных во вкладке Конструктор.

4 Назначения команд, размещенных во вкладке Макет.

1.3 Лабораторная работа № 3. Использование слияния для создания однотипных документов

Цель: освоить этапы создания документов, содержащих постоянную информацию и блоки переменных данных.

Слияние относится к процессам, ускоряющим создание серии однотипных документов, содержащих блоки постоянной информации и уникальные элементы.

Создаётся документ, содержащий одинаковую (постоянную) информацию для всех экземпляров документов. Данный документ будем называть **основным документом**.

Процесс **слияния** предполагает наличие ещё одного документа – **источника данных** для заполнения уникальных элементов **основного документа**.

Таким образом, **слияние** – это объединение информации **Основного документа** с **Источником данных**. В результате слияния получаем новый документ,



который назовём **Составной документ**.

Текстовый процессор **MS Word** имеет ряд вкладок, среди которых и вкладка **Рассылки**, где сосредоточены все операции по **слиянию**.

Процесс слияния можно разделить на ряд укрупнённых этапов:

- создание **Основного документа** и **Источника данных**;
- подключение к **Основному документу** **Источника данных**;
- размещение в **Основном документе** **полей слияния**;
- выполнение **слияния** и сохранение полученного **Составного документа** или же вывод на печать результатов **слияния**.

Задание

Следуйте нижеприведённым инструкциям для освоения процесса **слияния** двух документов.

1 Создайте в рабочей папке папку с именем **Слияние**.

2 Создайте **основной документ** в текстовом процессоре **MS Word**. Примерная форма основного документа представлена далее.

Извещение

Студенту **ФИО**

Вид аттест по *Предмет*

состоится Дата

в аудитории Место пр в *Время*

В тексте *приведены* условные обозначения меток, которые будут вставлены в основной документ при объединении его с источником данных. В тексте на месте меток можно оставить пробелы, а выделенные курсивом слова не вводить.

3 Сохраните документ в папке **Слияние** под именем **Извещение**.

4 Создайте **источник данных**.

Источник данных можно создавать в среде **MS Word**, **MS Excel** или **MS Access**, **MS Outlook** и т. д.

5 Создайте таблицу следующего вида (рисунок 1.1).

№	ФИО	<u>Вид аттест</u>	<u>Предмет</u>	<u>Место пр</u>	Дата	Время
	Иванов А.И.	зачет	Информатика	325	25.12.09	10-00
	Ковалев П.Л.	Экзамен	Химия	412	06.01.10	9-00

Рисунок 1.1

Заполните в таблице 5–7 строк с информацией по вашему выбору.

6 Сохраните файл, содержащий таблицу, в папке **Слияние** под именем **Студент**.

7 Перейдите в документ **Извещение** (основной документ).

8 Подключите источник данных к основному документу. Для этого в файле **Извещение** перейдите во вкладку **Рассылки**. Большинство элементов вкладки будут недоступны. Нажмите кнопку **Выбрать получателей** и выберите команду

Использовать существующий список.

В окне **Выбор источника данных** найдите файл источника данных (**Студент**). Далее дважды щелкните по значку файла источника данных (**Студент**) или выделите его и нажмите кнопку **Открыть**.

9 В появившемся окне **Выделить таблицу** выберите таблицу, из которой будут извлекаться данные для слияния. Этим шагом заканчивается процесс подключения источника данных к основному документу. Обратите внимание, что после подключения стали доступны элементы вкладки **Рассылки**.

10 Вставьте поля слияния в основной документ. Для этого установите курсор в место основного документа, куда требуется вставить **поле слияния**, щелкните по стрелке кнопки **Вставить поле слияния** и выберите из списка вставляемое поле.

Таким же образом вставьте в основной документ все необходимые **поля слияния**. Основной документ приобретет следующий вид.

Извещение

**Студенту «ФИО»
«Вид аттест» по «Предмет»
состоится «Дата»
в аудитории «Место пр» в «Время»**

11 Основной документ готов к слиянию. Для удобства просмотра документа можно затенить поля слияния (кнопка **Выделить поля слияния**). Вместо наименований полей можно отобразить их значения (кнопка **Просмотр результатов**). Пользуясь кнопками **Следующая запись**, **Предыдущая запись** и полем **Перейти к записи**, можно провести предварительный просмотр результата объединения документов.

12 Выполните слияние в новый документ. Для этого нажмите кнопку **Найти и объединить** и выберите команду **Изменить отдельные документы**. В окне **Составные новые документы** выберите записи источника данных, которые следует использовать для слияния. Выбираем **Все**.

13 Результатом слияния будет новый документ, который содержит текст основного документа со вставленными значениями полей из источника данных. Текст основного документа повторяется столько раз, сколько записей было выбрано для слияния. Каждый фрагмент документа заканчивается разрывом раздела.

14 Полученный документ можно напечатать и/или сохранить для дальнейшего использования.

15 Сохраните **составной документ** в папке **Слияние** под именем **Пакет**.

Контрольные вопросы

- 1 Назначение слияния.
- 2 Документы, используемые при выполнении слияния.
- 3 Программные средства, используемые для создания источника данных.



2 Табличный процессор Microsoft Excel

Программа **Microsoft Excel** предназначена для работы с таблицами данных. При формировании таблиц выполняется ввод, редактирование и форматирование данных.

Документ **Excel** называется *рабочей книгой*. Книга состоит из *рабочих листов*, имеющих табличную структуру. В окне документа отображается *текущий рабочий лист*, с которым и ведется работа.

Рабочий лист состоит из столбцов и строк, на пересечении которых образуются *ячейки* таблицы. Они являются минимальными элементами для хранения данных.

Отдельная ячейка может содержать текст, число или формулу, а также оставаться пустой. Ввод данных осуществляется непосредственно в текущую (активную) ячейку или в строку формул.

Тип данных, размещаемых в ячейке, определяется автоматически при вводе. Если данные можно интерпретировать как число, программа **Excel** так и делает. В противном случае данные рассматриваются как текст.

2.1 Лабораторная работа № 4. Ввод и редактирование данных

Цель: изучить типы данных, используемых в **Excel**, освоить способы ввода, редактирования и форматирования содержимого таблиц.

Задание

Создайте таблицу, введите данные и формулы. Выполните форматирование. *Выполните следующее.*

1. *Начало работы.*

1.1 Запустите программу **Excel**.

1.2 **Лист 1** переименуйте, присвойте имя **Расходы**.

1.3 Сохраните созданную книгу в рабочей папке, имя файла Lab_1_EX.

2 *Создание таблицы.*

2.1 На листе **Расходы** создайте таблицу по образцу, приведенному на рисунке 2.1.

№	Марка автомобиля	Вид топлива	Цена литра топлива	Расход топлива, л/100 км	Пробег	Стоимость топлива
Итого						

Рисунок 2.1



2.2 Для оформления строки заголовков используйте команду **Формат⇒Ячейки**.

2.3 Введите не менее пяти записей.

2.4 В столбце «**Стоимость топлива**» расходы вычислите, используя формулу (Стоимость топлива = Цена 1л × Расход на 100 км × Пробег / 100).

В строке **Итого** в столбце **Стоимость топлива** введите формулу суммы всех значений данного столбца.

2.5 Установите денежный формат ячеек соответствующих столбцов.

2.6 Выполните оформление таблицы.

2.7 Сохраните файл Lab_1_EX.

Контрольные вопросы

- 1 Назначение электронных таблиц.
- 2 Как определить, какая ячейка текущая?
- 3 Как адресуются ячейки электронной таблицы?
- 4 Основные элементы интерфейса EXCEL.
- 5 Что такое относительный и абсолютный адрес?

2.2 Лабораторная работа № 5. Использование встроенных функций

Цель: освоить выполнение вычислений в рабочих книгах с использованием функций Excel.

Теоретические положения

Функции в Excel применяются для выполнения стандартных вычислений в рабочих книгах. Значения, которые используются для вычисления функций, называются аргументами. В качестве аргументов функций можно использовать константы, ссылки на ячейки, имена диапазонов ячеек, а также другие функции. Значения, возвращаемые функциями в качестве ответа, называются результатами.

Функцию можно использовать как часть формулы, так и автономно размещая в ячейке рабочего листа. Excel содержит более 400 встроенных функций. Вводить с клавиатуры в формулу названия функций и значения входных параметров не всегда удобно и при использовании сложных формул сопровождается возрастанием вероятности возникновения ошибок при вводе. **Мастер функций** позволяет реализовать все возможности использования функций.

Мастер функций можно вызвать следующими способами:

- в строке формул нажать кнопку « f_x »;
- выполнить команду **Вставка(Insert)⇒Функция**.

В результате последовательно откроются диалоговые окна:

- выбора категории и функции;
- ввода аргументов функции.

При вводе аргументов в эти диалоговые окна там же можно просмотреть



результаты и, при необходимости, вызвать справку по используемой функции.

Для уточнения синтаксиса функций, используемых в выражениях, примените информацию, приведенную в справочной системе EXCEL.

При выполнении задания будут использоваться логические функции. Далее приведена краткая информация по синтаксису и особенностях применения некоторых логических функций.

Логические функции ЕСЛИ, И, ИЛИ

ЕСЛИ()

Синтаксис: **ЕСЛИ**(логическое_выражение; значение_если_истина; значение_если_ложь).

И()

Синтаксис: **И**(логич_знач1; логич_знач2; ... ; логич_знач30)

ИЛИ()

Синтаксис: **ИЛИ**(логич_знач1; логич_знач2; ... ; логич_знач30)

Функция **И** используется тогда, когда нужно проверить, выполняются ли несколько условий *одновременно*.

Функция **И** принимает от 1 до 30 аргументов (в Office 2007 и более новых версиях до 256), каждый из которых является логическим значением *ложь* или *истина* либо любым выражением или функцией, которое в результате дает *ложь* или *истина*.

Возвращаемое значение: если все аргументы функции **И** равны *истина*, функция **И** возвращает логическое значение *истина*. Если хотя бы один аргумент имеет значение *ложь*, функция **И** возвращает *ложь*.

Функция **ИЛИ** возвращает логическое значение. Если хотя бы один аргумент функции **ИЛИ** имеет значение *истина*, возвращает *истина*, если все аргументы равны *ложь*, функция **ИЛИ** возвращает *ложь*.

Примечание – Функции **И** и **ИЛИ** почти никогда не используются сами по себе. Обычно их используют в качестве аргумента других функций, например, **ЕСЛИ**.

Рассмотрим пример использования логических функций. Вычислить значение Y в зависимости от значений аргументов:

Y=	$\begin{cases} x^3 + \cos(x) + \ln x & x \geq 1 \text{ и } a+b=5 \\ \cos(x)^3 + \sin(x)^2 - 5a + 1/b^{1/3} \end{cases}$

На рисунке 2.2 представлен фрагмент листа Excel, в ячейки которого внесены несколько вариантов значений аргументов и формулы.



x	a	b	Y1	Y2	Y
2	2	3	1,27700034	-8,5519	1,277
2	3	3	8,27700034	-13,552	-13,552
0	3	1	#ЧИСЛО!	-13	-13
-2	3	2	-7,7229997	-13,452	-13,452
1	5	0	1,54030231	#ДЕЛ/0!	1,5403

Рисунок 2.2

Далее приведены формулы, вычисляющие значения выражений при различных значениях аргументов : **x**, **a** и **b**.

$=X^{F163}+\text{COS}(B3)+\text{LN}(\text{ABS}(B3))$ – формула в столбце Y1, во второй строке;

$=\text{COS}(B3)^3+\text{SIN}(B3)^2-5*C3+1/D3^{(1/3)}$ – формула в столбце Y2, во второй строке;

$=\text{ЕСЛИ}(\text{И}(B3 \geq 1; (C3+D3)=5); E3; F3)$ – формула в столбце Y, во второй строке.

Задание

Вычислите выражения по заданному варианту при различных значениях аргументов.

Контрольные вопросы

- 1 Особенности ввода формул в ячейки рабочего листа.
- 2 Создание формул массива.
- 3 Что такое функция и как её можно задавать?
- 4 Мастер функций. Способы вызова.
- 5 Синтаксис логических функций ЕСЛИ, И, ИЛИ.

2.3 Лабораторная работа № 6. Диаграммы, графики

Цель: освоить способы построения диаграмм и графиков.

Для построения диаграмм на основании табличных данных используется **Мастер диаграмм**.

Перед вызовом **Мастера диаграмм** предпочтительно выделить диапазоны ячеек, содержащих информацию, которая должна использоваться при создании диаграммы.

Используемая информация, содержащаяся в столбцах или строках таблицы, образует ряды данных.

Мастер диаграмм осуществляет пошаговое руководство процессом создания диаграммы. Для вызова Мастера диаграмм используется команда **Вставка** ⇒ **Диаграмма**.



Для построения диаграммы выполните следующие действия.

1 Выделите область данных, по которым будет строиться диаграмма.

2 Загрузите **Мастер диаграмм, Excel** выведет диалоговое окно мастера.

3 Выберите **Тип** диаграммы и **Вид**.

4 Щелкните по кнопке **Далее** или нажмите **Enter**.

Появится следующее диалоговое окно Мастера диаграмм.

5 Проверьте, действительно ли по данным, содержащимся в выбранном интервале клеток, вы хотите построить диаграмму.

6 Щелкните по кнопке **Далее** или нажмите **Enter**.

7 В следующем диалоговом окне можно ввести заголовок для всей диаграммы, так же как и индивидуальные заголовки для независимой переменной (**ось X**) и зависимой переменной (**ось Y**).

8 Щелкните по кнопке **Далее** или **Enter**. В появившемся диалоговом окне необходимо указать место размещения диаграммы на текущем листе или отдельном.

9 Щелкните по кнопке **ГОТОВО**.

Редактировать можно все объекты диаграммы.

Основными объектами диаграммы являются:

- область диаграммы;
- область построения диаграммы;
- ряды данных;
- оси категорий и значений.

После наведения курсора мыши на выбранный объект и вызова контекстного меню появляется меню с набором всех возможных действий с объектом.

Задание

Решите задачу о траектории движения тела.

Рассматривается движение тела, имеющего начальную скорость v_0 и угол φ между вектором v_0 и линией горизонта.

Уравнения движения тела с учетом указанных параметров v_0 , φ и силы тяжести (сопротивление воздуха не учитываем) имеют вид:

$$X = V_0 \cdot \cos \varphi \cdot t;$$

$$Y = V_0 \cdot \sin \varphi \cdot t - g \cdot t^2 / 2,$$

где X – координата по горизонтали (дальность);

Y – координата по вертикали (высота);

t – время;

g – ускорение свободного падения.

Время полета при заданных значениях параметров v_0 и φ

$$t_k = \frac{2v_0 \cdot \sin \varphi}{g}.$$

Выполнить следующее.



1 На листе **Excel** создать расчетную таблицу для определения изменения координат тела в зависимости от времени при заданных значениях v_0 , φ и вычисленного времени полета. Задав конкретное значение v_0 (например, 50, 100...) и изменяя значения угла φ , определить: максимальную дальность полета, максимальную высоту полета, угол, при котором снаряд попадет в точку с координатой **150**.

2 Построить график изменения координат тела для одного из вариантов начальных данных.

На рисунке 2.4 показан график изменения координат тела для варианта данных, представленных на рисунке 2.3.

	A	B	C	D	E	F	G
1							
2			Данные для расчета траектории				
3			Скорость V_0 м/сек			50	
4			Угол φ (градусы)			30	
5			Угол φ (радианы)			0,5235988	
6			Время полета (t_0)			5,1020408	
7							
8			Таблица для расчета траектории				
9			№	t	X	Y	
10			0	0,00	0,00	0,00	
11			1	0,26	11,05	6,06	
12			2	0,51	22,09	11,48	
13			3	0,77	33,14	16,26	
14			4	1,02	44,18	20,41	
15			5	1,28	55,23	23,92	
16			6	1,53	66,28	26,79	
17			7	1,79	77,32	29,02	
18			8	2,04	88,37	30,61	
19			9	2,30	99,42	31,57	
20			10	2,55	110,46	31,89	
21			11	2,81	121,51	31,57	
22			12	3,06	132,55	30,61	
23			13	3,32	143,60	29,02	
24			14	3,57	154,65	26,79	
25			15	3,83	165,69	23,92	
26			16	4,08	176,74	20,41	
27			17	4,34	187,79	16,26	
28			18	4,59	198,83	11,48	
29			19	4,85	209,88	6,06	
30			20	5,10	220,92	0,00	

Рисунок 2.3

Координаты тела

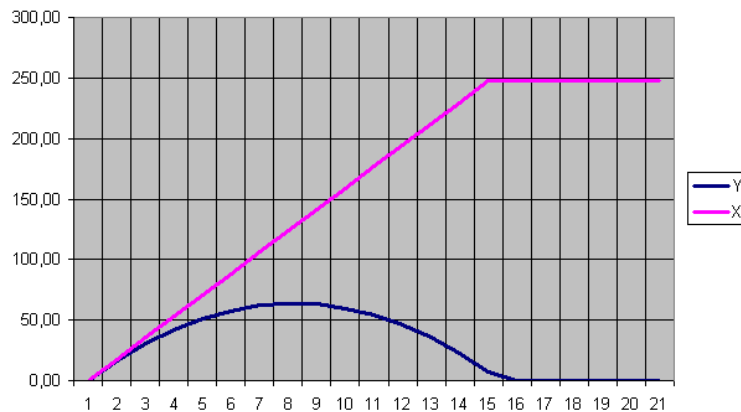


Рисунок. 2.4

Пояснения к данным, представленным на рисунке 2.3

В качестве исходных данных в таблицу «Данные для расчета траектории» вносятся значения v_0 , φ (в градусах).

В расчетных формулах используется значение угла в радианах. В ячейку F5 внесена формула $=\text{ПИ}()/180*\text{F4}$ для пересчета градусов в радианы.

В ячейку F6 внесена формула (см. выше) для расчета времени полета.

В ячейки строки 2 таблицы для расчета траектории (10-я строка на листе) внесены следующие формулы.

$=\text{\$F\$6}/20*\text{C10}$	$=\text{F\$3}*\text{COS}(\text{F\$5})*\text{D10}$	$=\text{F\$3}*\text{SIN}(\text{F\$5})*\text{D10}-9,8*\text{D10}^2/2$
--------------------------------	---	--

Для заполнения формулами остальных строк применяется копирование. Для ссылки на ячейки, которые не должны меняться при копировании, используются абсолютные или смешанные адреса.

Контрольные вопросы

- 1 Средства, используемые для построения диаграмм.
- 2 Способы размещения данных, используемых для построения диаграмм.
- 3 Основные объекты диаграммы.
- 4 Какие адреса используются для ссылки на ячейки, которые не должны меняться при копировании?

2.4 Лабораторная работа № 7. Решение уравнений и систем уравнений. Задачи оптимизации

Цель: изучить методы решения уравнений и систем уравнений, а также команды **Подбор параметра** и **Поиск решения**.

Теоретические положения

Надстройки – это специальные средства, расширяющие возможности программы **Excel**. На практике именно надстройки делают программу **Excel** удобной для использования в научно-технической работе. Хотя эти средства считаются внешними, дополнительными, доступ к ним осуществляется при помощи обычных команд строки меню (Сервис или Данные). Команда использования надстройки обычно открывает специальное диалоговое окно, оформление которого не отличается от стандартных диалоговых окон программы **Excel**.

Надстройка **Подбор параметра** обеспечивает вычисление значения аргумента (параметра) для заданного значения (максимальное, минимальное или заданное значения), изменяет значения величин во влияющих ячейках, учитывая ограничения.



Решение уравнений средствами программы Excel

Рассмотрим использование команды **Подбор параметра**.

Задача. Найдите решение уравнения $x^3 - 3x^2 + x + 1 = 0$.

При применении этого метода в одну ячейку вводится формула, вторая ячейка используется для искомого решения.

На рисунке 2.5 представлен один из вариантов размещения данных на листе Excel.

C3		fx	
		=B3^3-3*B3^2+B3+1	
A	B	C	D
	x	F(x)	
		1,00	

Рисунок 2.5

Для решения уравнения выполните следующее.

1 В ячейку **C3** вносится левая часть уравнения, используя в качестве переменной x ссылку на ячейку **B3**. Формула видна в строке ввода.

2 Выполните команду **Подбор параметра** (**Сервис** \Rightarrow **Подбор параметра** в **Excel 2003** или **Данные** \Rightarrow **Анализ** \Rightarrow **Подбор параметра** в более новых версиях).

3 В поле **Установить в ячейке** укажите **C3**, в поле **Значение** задайте 0, в поле **Изменяя значение ячейки** укажите **A1**.

4 Щелкните по кнопке **ОК** и посмотрите на результат подбора, отображаемый в диалоговом окне **Результат подбора параметра**.

5 Щелкните по кнопке **ОК**, чтобы сохранить полученные значения ячеек, участвовавших в операции.

Решение с использованием матричного уравнения

$X = A^{-1}B$ – матричная запись решения рассматриваемой системы.

A – основная матрица системы.

A^{-1} – обратная матрица A .

X – вектор-столбец независимых переменных.

B – вектор-столбец свободных членов.

Решение

Открываем мастер функций, используем функции обработки массивов, которые находятся в категории «математические».



Выполните следующее.

- 1 Исходные данные внесите на рабочий лист.
- 2 Для расчета обратной матрицы выделите диапазон ячеек, совпадающий по размеру с основной матрицей.
- 3 Активизируйте верхнюю левую ячейку диапазона создаваемой обратной матрицы и вызовите функцию **МОБР**.
- 4 В диалоговом окне введите диапазон, в котором размещена основная матрица системы. Для ввода выделите мышью диапазон в таблице исходных данных.
- 5 Нажмите сочетание клавиш **CTRL + SHIFT + ENTER**.
- 6 Для определения вектора столбца независимых переменных выделите диапазон.
- 7 Активизируйте верхнюю ячейку диапазона и вызовите функцию **МУМНЖ**.
- 8 В диалоговом окне укажите параметры перемножаемых массивов.
- 9 Нажмите сочетание клавиш **CTRL + SHIFT + ENTER**.

Метод Крамера

Метод Крамера $x_i = \Delta_i / \Delta$ (где Δ – определитель основной матрицы; Δ_i – определитель, полученный из основного подстановкой вместо i -го столбца столбца свободных членов).

Для вычисления определителя используется функция **МОПРЕД**, аргументом которой является диапазон размещения соответствующего массива.

Исходные данные для создания всех требуемых массивов и последующего вычисления определителей необходимо разместить на листе Excel.

Команда Поиск решения

Команда **Поиск решения** может быть применена для решения оптимизационных задач и нахождения решений систем линейных уравнений.

Использование данной команды для решения системы линейных уравнений рассмотрим на примере.

Пример – Решите систему уравнений

$$\begin{cases} x + y + z = 6; \\ 2x - y + z = 3; \\ x - 4y + 3z = 2. \end{cases}$$

При использовании команды **Поиск решения** на листе Excel необходимо предусмотреть диапазоны для размещения формул уравнений системы, правых частей уравнений, искомых значений неизвестных, а также поясняющих обозна-



чений. В приведенном далее варианте в ячейках A1...A3 обозначения переменных в соответствующих строках столбца В.

Выполните следующее.

1 В ячейки В1, В2 и В3 внесите начальные приближения переменных, например, 0.

2 В столбец D внесите левые части уравнений, ссылаясь на введенные значения.

3 В столбец E внесите правые (известные) части уравнений. В результате получится следующее (рисунок 2.6).

	А	В	С	D	E
1	x	0		=B1+B2+B3	6
2	y	0		=2*B1-B2+B3	3
3	z	0		=3*B3-4*B2+B1	2

Рисунок 2.6

4 Выполните команду **Поиск решения** (Сервис \Rightarrow Поиск решения в Excel 2003 или Данные \Rightarrow Поиск решения в более новых версиях).

5 Заполните окно **Поиск решения** (рисунок 2.7) следующим образом.

Одно из введенных уравнений принимается за целевую функцию, в рассматриваемой задаче используется первое уравнение. Адрес ячейки с этим уравнением указываем в качестве целевой функции, равной значению правой части данного уравнения.

Изменяя ячейки, указываем диапазон, отведенный под искомые значения неизвестных. В качестве ограничений указываем диапазон размещения остальных уравнений и диапазон правых частей уравнений.

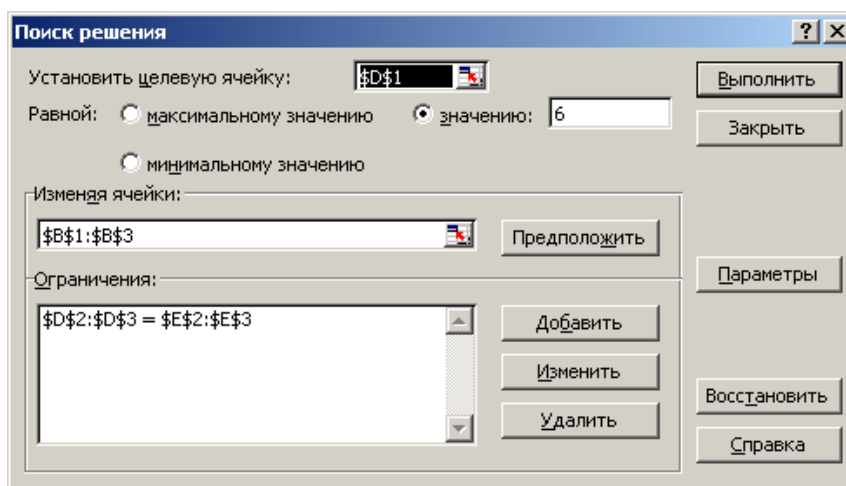


Рисунок 2.7

Нажав кнопку **Выполнить**, получим результаты (рисунок 2.8).



	A	B	C	D	E
1	x	1		6,000001	6
2	y	2		3	3
3	z	3,000001		2	2
4					

Рисунок 2.8

Контрольные вопросы

- 1 Назначение надстроек, используемых в программе Excel.
- 2 Алгоритм использования метода Подбор параметра.
- 3 Средства и методы, используемые в Excel для решения систем линейных уравнений.
- 4 Особенности использования функций массива.
- 5 Для решения каких задач используется надстройка Поиск решения?

2.5 Лабораторная работа № 8. Базы данных. Основные понятия

Цель: изучить основные понятия баз данных в приложении MS Excel.

Теоретические положения

Списки Excel как база данных.

Приложение Microsoft Excel обладает богатыми встроенными средствами для обработки и анализа данных. Аналогом простой базы данных в Excel служить список.

Список – это группа строк таблицы, содержащая связанные данные.

Отличительной особенностью списка является то, что каждый его столбец содержит однотипные данные, например, перечень фамилий, дату рождения и т. д.

Если провести аналогию между списком и табличной базой данных, то столбцы списка – поля базы данных, а его строки – записи. Считается, что первая строка списка является его заголовком и содержит названия столбцов списка. Заголовок должен иметь на листе электронных таблиц горизонтальную ориентацию. В списке не должно быть пустых строк и столбцов.

Сортировка данных.

Команда **Сортировка** позволяет переставить записи в другом порядке на основании значений одного или нескольких столбцов. Записи сортируются по возрастанию/убыванию или по выбранному пользователем порядку (например, по дням недели).

Чтобы отсортировать список, надо:

- 1) установить курсор в любую ячейку списка;
- 2) выполнить команду **Сортировка** на ленте **Данные** в группе **Сортировка и Фильтр**;



3) в диалоговом окне **Сортировка** выбрать поле, по которому будет происходить сортировка. Укажите тип сортировки и порядок (по возрастанию, убыванию, настраиваемый).

Автофильтр.

Отфильтровать список – значит показать только те записи, которые удовлетворяют заданному критерию.

Чтобы *установить или убрать автофильтр*, надо на ленте **Данные** в группе **Сортировка и фильтр** выбрать команду **Фильтр**. После этого нажать кнопку со стрелкой возле названия какого-либо поля, чтобы раскрыть список его элементов, и выбрать отображаемые значения или задать условие отбора. На экране появятся только те записи, которые отвечают заданному условию. В случае необходимости можно продолжить фильтрацию, нажимая кнопки со стрелками на других полях.

Показать все записи по всем полям, не убирая фильтр, – команда **Очистить**.

Для данных разного типа существуют дополнительные автофильтры, которые находятся в списке критериев **Текстовые фильтры**, **Числовые фильтры**, **Фильтры по дате** и т. д.

Если выделить какое-то числовое поле (например, *Возраст*), а в списке критериев выбрать **Числовые фильтры**, то появится список дополнительных фильтров, которые позволяют:

– задать критерий в виде неравенства – критерии равно, не равно, больше, больше или равно, меньше, меньше или равно, между;

– вывести первые N значений – критерий Первые 10: после выбора в списке Числовых фильтров команду Первые 10..., необходимо в появившемся окне указать число значений (N), а также способ вычисления: количество элементов списка, процент от количества элементов;

– определить условие по среднему значению в указанном столбце – критерии Выше среднего, Ниже среднего.

Расширенный фильтр.

Расширенный фильтр позволяет сформировать более сложные условия, в том числе состоящие более чем из двух условий.

Перед вызовом команды **Расширенный фильтр** необходимо сформировать критерии. Для удобства лучше формировать критерии на отдельном листе (можно дать ему имя, например, *Критерии*) и давать критериям имена *Kp1*, *Kp2* и т. д.

Основное правило: если критерии связаны между собой операцией **И**, то они должны располагаться в одной строке, а если операцией **ИЛИ** – то в разных.

После формирования критерия вызывают расширенный фильтр: на ленте **Данные** в группе **Сортировка и фильтр** команда **Дополнительно**.

Восстановить исходный список можно, выбрав на ленте **Данные** в группе **Сортировка и фильтр** команду **Очистить**.



Работа с базой данных

1 Ознакомьтесь с теоретическими положениями.

2 Получите файл с готовой базой данных у преподавателя или создайте собственную базу данных для автоматизации любой предметной области человеческой деятельности (учет сотрудников на предприятии, отдел кадров, туристическое агентство, центр недвижимости, гостиница, магазин и т. д.).

3 Используя построенную модель базы данных, выполните следующее.

3.1 Подведите итоги по одному или двум атрибутам.

3.2 Создайте и примените запросы на отбор данных с помощью автофильтра, используйте при этом простой автофильтр по значению и автофильтр с дополнительными критериями для данных разных типов (числовые, текстовые, дата / время), а также настраиваемый пользовательский автофильтр.

3.3 Создайте и примените запросы на поиск и отбор данных с помощью расширенного фильтра таким образом, чтобы созданные критерии содержали два-три условия, относящиеся как минимум к двум различным полям, и среди критериев были вычисляемые.

Контрольные вопросы

1 Для каких целей применяются электронные таблицы?

2 Для чего необходимы базы данных?

3 Что такое запись?

4 Что такое поле?

2.6 Лабораторная работа № 9. Базы данных. Методы обработки информации

Цель: освоить основные методы обработки информации в базе данных.

Задание 1. Структура базы данных

Создайте или получите у преподавателя файл **Excel**, состоящий из двух рабочих листов:

– лист **Сотрудники** (рисунок 2.9) – таблица с полями **Фамилия, Имя, Отчество, Табельный номер, Дата приема на работу, Код, Пол, Дата рождения** – содержит информацию для сотрудников некоторой условной фирмы;

– лист **Должность** (рисунок 2.10) – таблица с полями **Код, Должность, Коэффициент, Отдел**.



	A	B	C	D	E	F	G	H
1	Фамилия	Имя	Отчество	Табельный_номер	Дата_приема	Код	Пол	Дата_рождения
2	Абрамов	Илья	Борисович	865	03.09.2006	15	м	19.11.81
3	Алексеев	Тарас	Климович	804	13.09.2000	10	м	17.04.42
4	Алексеева	Таисия	Николаевна	856	24.10.2005	13	ж	18.08.68
5	Андреева	Елена	Артемовна	823	23.11.2002	8	ж	22.08.58
6	Антонова	Ольга	Андреевна	813	04.09.2001	15	ж	11.05.72
7	Арканов	Павел	Дмитриевич	814	29.09.2001	12	м	27.12.71
8	Арканов	Антон	Викторович	828	28.05.2003	13	м	08.03.78
9	Арканова	Наталия	Ивановна	807	07.12.2000	4	ж	20.01.83
10	Бажов	Петр	Викторович	818	22.05.2002	5	м	05.05.51

Рисунок 2.9 – Лист Сотрудники

	A	B	C	D
1	Код	Должность	Кoeffициент	Отдел
2	1	директор	12,00	АОП
3	2	зам.директора	9,00	АОП
4	3	зав.отделом	6,00	
5	4	администратор	5,00	МПО
6	5	менеджер по продажам	3,00	МПО
7	6	менеджер по рекламе	3,00	МПО
8	7	ст.экономист	6,00	ФЭО
9	8	экономист	5,00	ФЭО
10	9	бухгалтер	4,00	ФЭО
11	10	программист	6,00	ИВЦ
12	11	техник	4,00	ИВЦ
13	12	оператор	2,00	ИВЦ
14	13	инженер	4,00	ИТО
15	14	дизайнер	5,00	ИТО
16	15	секретарь	3,00	АДМ
17	16	курьер	2,00	МПО

Рисунок 2.10 – Фрагмент листа Должность

Задание 2. Редактирование базы данных

1 На листе **Должность** после столбца **Кoeffициент** добавьте пустой столбец с заголовком **Оклад**. Подсчитайте оклад, умножив коэффициент на базовую величину (24,5 р. с 01.01.2018).

2 На листе **Сотрудники** после столбца **Дата приема** добавьте пустой столбец с заголовком **Стаж**. Подсчитайте стаж работы сотрудника на данном предприятии.

3 На листе **Сотрудники** после столбца **Дата рождения** добавьте столбец **Возраст**, в котором подсчитайте примерный возраст каждого сотрудника (функции ОКРУГЛВНИЗ, ДОЛЯГОДА (базис 3), СЕГОДНЯ).

Задание 3. Форматирование списка

Оформите список согласно требованиям.

1 Осуществите подгонку ширины столбцов.

2 Отформатируйте заголовки полей по центру, используя полужирное начертание.

3 Отформатируйте по центру значения всех столбцов, кроме столбцов **Фамилия, Имя, Отчество, Должность**.

Задание 4. Сортировка списка

1 Отсортируйте список по полю Табельный_номер (на ленте Данные в группе Сортировка и фильтр команда Сортировка).

2 Отсортируйте список по полю Отдел так, чтобы отделы шли в порядке АОП, ФЭО, ИВЦ, ИТО, МПО. Для этого создайте свой список сортировки.

3 Отсортируйте список по двум ключам: по полу – сначала женщины, потом мужчины, а затем по стажу.

4 Отсортируйте список по ключам: по отделам, внутри отделов – сначала мужчины, потом женщины и, наконец, по фамилиям, именам и отчествам.

Контрольные вопросы

1 Что такое тип данных? Зачем необходимо указывать типы полей?

2 Данные каких типов могут быть записаны в ячейку?

3 Что такое сортировка? Как отсортировать список по двум и более ключам?

4 Какие виды фильтров вы знаете?

5 Что такое расширенный фильтр?

6 Как формировать критерии для расширенного фильтра?

2.7 Лабораторная работа № 10. Макросы

Цель: изучить способы создания и использования макросов.

Средство записи макросов **Excel** позволяет записывать последовательность действий пользователя, а затем преобразует их в код **VBA**.

Макрос – записанная последовательность команд и действий пользователя, сохраненная под уникальным именем, которую может выполнить **Excel**.

Рассмотрим процедуру записи макросов на примере простого макроса, который изменяет шрифт и цвет в выделенном диапазоне ячеек.

Выполните следующее.

1 Откройте новую рабочую книгу. Сохраните файл в рабочей папке с именем **Лаб_Макрос**.

2 В ячейку **A1** введите ваше имя, в **B1** – фамилию, в **C1** – название города, в котором живете, а в **D1** – название страны. Это данные, с которыми будете работать далее.

3 Перейдите в ячейку **A1**.

4 Выполните команду **Разработчик** ⇒ **Запись макроса**. Появится диалоговое окно **Запись макроса**. В этом окне можно ввести имя макроса и назначить сочетание клавиш для выполнения макроса.

5 Введите имя макроса, например **БольшойШрифт**, и нажмите клавишу



Enter для начала записи макроса. Заметьте, что в строке состояния окна **Excel** отобразилось слово **Запись**. Обычно также на экране появляется панель инструментов **Остановить запись**.

6 Выполните команду **Формат** ⇒ **Ячейки**. Появится диалоговое окно **Формат ячеек**. Перейдите на вкладку **Шрифт**.

7 Установите размер шрифта **16** пунктов, цвет – **красный**. Щелкните по кнопке **ОК**.

8 Щелкните по кнопке **Остановить запись** одноименной панели инструментов. Сеанс записи макроса закончен.

После записи макроса его можно выполнить.

Выполнение макросов

При выполнении макроса **Excel** повторяет те же действия, которые производились в процессе его создания.

Для выполнения макроса выполните следующее.

1 Выделите ячейку B1.

2 Выполните команду **Разработчик**⇒**Макрос**. Появится диалоговое окно **Макрос**. Диалоговое окно Макрос можно также отобразить с помощью комбинации клавиш <Alt+F8>.

3 Выделите макрос **БольшойШрифт** (если вы так назвали свой макрос) и щелкните по кнопке **Выполнить**. Шрифт в ячейке B1 станет красным и размером 16 пунктов.

4 Выделите ячейки C1 и D1 и снова выполните макрос **БольшойШрифт**. Заметьте, что в обеих ячейках шрифт стал красным и размером 16 пунктов, хотя при создании макроса была выделена только одна ячейка.

Назначение макросу клавишной команды

Для назначения макросу клавишной команды сделайте следующее.

1 Выберите вкладку **Разработчик**, нажмите кнопку **Макросы**.

2 В списке макросов выберите макрос, для которого назначается или меняется клавишная команда.

3 Нажмите сочетание клавиш, которое хотите присвоить макросу. Назначенное сочетание появится в поле **Новое сочетание клавиш**.

4 В поле **Сохранить изменения** выберите шаблон, для которого создавался макрос.

5 Нажмите кнопку **Назначить**, затем **Заккрыть**.

Задание

Создайте макросы, выполняющие:

- 1) построение заголовка и шапки таблицы (рисунок 2.11);
- 2) добавление в текст таблицы с заданным количеством столбцов и строк;
- 3) вычисление по формуле значений в ячейках таблицы;
- 4) вычисление итоговых значений в таблице;



5) вставку в текст заданного текстового фрагмента с указанными параметрами форматирования.

Создайте новый текстовый файл (например, Лаб_Макрос), сохраните его в рабочей папке.

Создайте макрос по построению заголовка и шапки таблицы следующего вида (см. рисунок 2.11).

№ п.п	Наименование	Количество	Цена	Сумма
-------	--------------	------------	------	-------

Рисунок 2.11

Контрольные вопросы

- 1 Назначение макросов.
- 2 Способы создания и редактирования макросов.
- 3 Выполнение макросов.

3 Основы языка VBA

3.1 Лабораторная работа № 11. Интерфейс редактора VBA. Программы линейной структуры

Цель: изучить основные элементы интерфейса редактора VBA и методику создания процедур линейной структуры.

Интерфейс редактора VBA

Visual Basic for Application (VBA) представляет собой консольное приложение, реализованное комплексом динамических и объектных библиотек **VBA**, функционирует только в составе приложений **MS Office**, таких как **Excel, Word, Access** и т. д.

Запуск **VBA** из среды **Excel**, как впрочем и других приложений **MS Office**, осуществляется посредством выполнения команды меню **Разработчик**⇒**Visual Basic**. Универсальный способ запуска VBA для любых версий MS Office – использование **[Alt + F11]**.

Интерфейс редактора **VBA** состоит из главной формы и дочерних форм. Главная форма редактора **VBA** включает в себя системное меню и панель управления, содержащую кнопки команд дублирующие основные команды системного меню. Важными дочерними окнами интерфейса являются окно проекта, окно редактора свойств объектов и окна редактора текста программы (редактора кода). Доступ к окну проекта и окну редактора свойств объектов осуществляется командами меню **Вид** ⇒ **Окно проекта**, **Вид**⇒**Окно свойств** соответственно. Доступ к окну редактора кода осуществляется командой меню **Вид**⇒**Программа** при условии выбора формы или модуля в окне проекта либо двойным щелчком мыши над выбранным объектом в окне проекта. Другими, менее значи-



мыми, окнами интерфейса являются окно отладки и окно локальных переменных, доступ к которым также осуществляется из меню **Вид**.

Создание приложений и отладка программ с алгоритмами линейной структуры.

Алгоритм линейной структуры – это алгоритм, в котором блоки выполняются последовательно друг за другом. Программа линейной структуры реализует линейный алгоритм. Для организации программы линейной структуры используют операторы присваивания, ввода исходных данных и вывода результатов обработки данных.

Чаще всего линейные алгоритмы используются для программирования вычислений по формулам, которые записываются в виде выражений.

Выражения состоят из констант, переменных, операций, функций и круглых скобок, определяющих последовательность выполнения действий. Значения выражений обычно присваиваются переменным.

С каждым видом выражений связаны определенные операции и встроенные (стандартные) функции языка **VBA**. Для числовых значений применяются арифметические операции и математические функции.

Программа (код программы) записывается в окне кода.

Окно кода используется при написании любой программы **VBA**, будь это код макроса, запуск которого осуществляется при нажатии кнопки в созданной пользователем форме, или подпрограмма.

Под строкой заголовка окна расположены два списка. В первом списке выводятся все объекты модуля, а во втором – список процедур, связанных с выбранным объектом.

Код программы вводится непосредственно в окно кода, так же как текст в любом текстовом редакторе.

Код программы может быть как связан с формой **UserForm** (программа пишется для соответствующей формы), так и не связан с ней. В последнем случае программу пишут в окне модуля.

Для того чтобы получить окно модуля, необходимо выполнить следующие действия: **Вставка** → **Модуль**.

Если программа пишется для обработки информации с использованием формы (**UserForm**) для получения окна формы необходимо произвести такие действия: **Вставка** → **UserForm**.

Если программа размещается в объекте форма, то ее код включает ряд подпрограмм. Для каждого события, возникающего в форме, необходимо написать процедуру (последовательность совместно выполняемых инструкций, имеющая имя) обработки.

Пример 1 – Решите задачу: найдите сумму $a + b = c$.

Порядок выполнения работы.

1 Откройте редактор **VBA**.



2 Выполните команду **Вставка** → **UserForm**.

3 Поместите на форму элементы, требуемые для решения задачи, с панели элементов и расположите их нужным образом (рисунок 3.1).

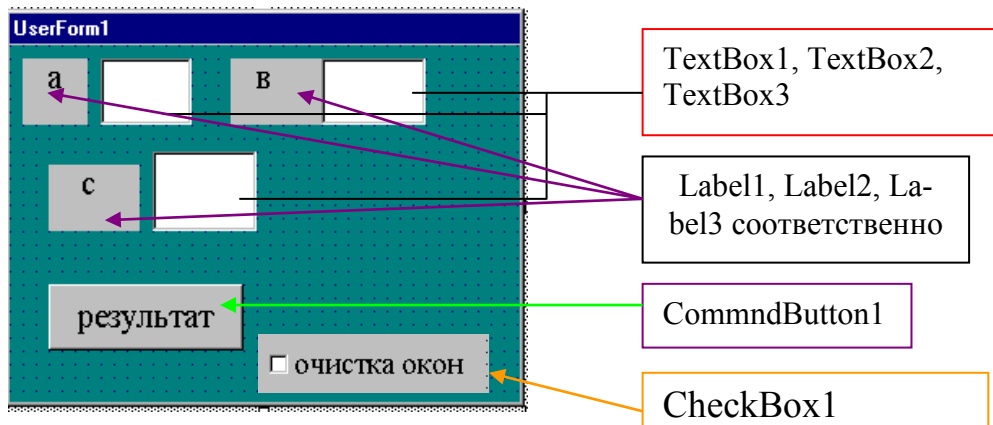


Рисунок 3.1

4 Напишите программный код. Для этого рекомендуется выполнить двойной щелчок по кнопке **результат** и перейти в окно ввода, где набрать текст процедуры обработки события `Click()` для кнопки `CommandButton1` и для флажка `CheckBox1`. Листинги процедур приведены далее.

```
Private Sub CheckBox1_Click()
    TextBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
    TextBox3.Visible = False
    TextBox1.SetFocus
    CheckBox1.Value = False
End Sub
```

```
Private Sub CommandButton1_Click()
    Dim a As Integer
    Dim b As Integer
    Dim c As Integer
    a = CInt(TextBox1.Text)
    b = CInt(TextBox2.Text)
    c = a + b
    MsgBox "результат смотри в TextBox3"
    TextBox3.Visible = True
    TextBox3.Text = c
End Sub
```

Задание

Вычислите выражение. Вариант задания согласуйте с преподавателем. При выполнении задания в качестве аналога используйте пример 1.

Контрольные вопросы

- 1 Назначение и использование алгоритмов линейной структуры.
- 2 Основные операторы, используемые для создания алгоритмов линейной структуры.
- 3 Понятие выражения и его компоненты.
- 4 Где записываются коды, средства, используемые для написания текстов программ?

3.2 Лабораторная работа № 12. Операторы ветвления. Объекты Excel

Цель: изучить алгоритмы и операторы, используемые в программах разветвляющейся структуры.

Оператор условного перехода

В программном коде, чтобы реализовать ветвление, применяется условный оператор **If Then**.

Условный оператор позволяет выбирать и выполнять действия в зависимости от истинности некоторого условия. Имеются два варианта синтаксиса. В первом случае он имеет вид:

IF <условие> **Then** [операторы 1] [**Else** [операторы 2]]

Во втором случае оператор расположен на нескольких строках:

```
IF <условие> Then
[операторы 1]
[ElseIf <условие > Then
[операторы 2 ]
[Else]
[Иначе Операторы 3]]
End If
```

Здесь условие обязательно в обоих вариантах. Оно может быть числовым или строковым выражением со значением **True** или **False**.

Далее рассматриваются примеры использования условного оператора и задания для самостоятельного выполнения.

Пример 2 – Определите положение точки с заданными координатами на плоскости.

Если точка $M(x_1, y_1)$ лежит:

– выше прямой $y=kx+b$, то выдается сообщение «Точка M лежит выше прямой»;



- ниже прямой, то выдается сообщение «Точка М лежит ниже прямой»;
- на прямой, то выдается сообщение «Точка М лежит на прямой».

Решение

Чтобы определить положение точки относительно прямой, нужно подставить координату точки x_1 в уравнение прямой и сравнить полученное значение Y с координатой точки y_1 . Если значение $y = y_1$, точка лежит на прямой; если полученное значение $y > y_1$, точка лежит ниже прямой; если $y < y_1$, точка лежит выше прямой.

Примерная структура процедуры, пронумерованные блоки выражений и операторов приведены далее. Для вывода сообщения использована функция MsgBox.

Private Sub Program1()

- 1 Блок описания переменных x_1, y_1, k, b, y .
- 2 Блок операторов считывания значений переменных из окон TextBox.
- 3 Оператор определения значения y ($y = k * x_1 + b$).
- 4 Условный оператор (текст приведен далее).

If y = y1 Then

MsgBox "Точка лежит на прямой"

Else If y > y1 Then

MsgBox "Точка лежит ниже прямой"

Else

MsgBox "Точка лежит выше прямой"

End If

End Sub

Задание

Осуществите ввод координат точки и установите ее положение относительно выделенной области (выбор варианта исходных данных по согласованию с преподавателем).

Программирование с использованием объектов EXCEL

Рассмотрим разработку приложения с использованием объектов EXCEL. Будем использовать объекты:

Worksheets() – для обозначения листа Excel;

Range() – для обозначения диапазона ячеек;

Свойство **.Value** – для обращения к значению ячейки.

Если мы хотим записать в ячейку значение F программным способом, нужно написать в программе так:

Worksheets().Range().Value = F, где в скобках указать имя или номер листа Excel и адрес ячейки.



Пример 3 – Введите X, вычислите F по формуле

$$F = \begin{cases} X/2, & \text{если } X > 0; \\ (X+1)/2, & \text{если } X < 0. \end{cases}$$

Решение

Введите исходные данные задачи на лист Excel. В таблице, приведенной на рисунке 3.1, указаны адреса ячеек и содержание вводимой информации.

Адрес ячейки	Вводимая информация
A1	<i>текст</i> Исходные данные
A2	<i>текст</i> X =
B2	<i>Значение</i> X
C1	<i>текст</i> Результат при x > 0
D1	<i>текст</i> Результат при x < 0

Рисунок 3.1

Откройте редактор VBA.

Выполните команду **Вставка** → **Модуль**.

В модуль вставьте процедуру **Sub LL()**, текст которой приведен далее.

Sub LL ()

X=Worksheets(1).Range("B2").Value

IF X>0 Then

F=X/2

Worksheets(1).Range("C2").Value = F

Else

F=(X+1)

Worksheets(1).Range("D2").Value = F

End If

End Sub

Контрольные вопросы

- 1 Варианты синтаксиса условного оператора If Then.
- 2 Способы открытия редактора VBA.
- 3 Приведите примеры объектов Excel.
- 4 Назначение и использование функций InputBox и MsgBox.



3.3 Лабораторная работа № 13. Операторы цикла

Цель: изучить основные понятия и использование в программах операторов цикла.

Операторы цикла (*For...Next*) и (*For Each...Next*)

Цикл **For...Next**

Оператор цикла **For** позволяет повторять группу операторов заданное число раз. Синтаксис:

For счётчик_цикла = начало **To** конец [**step шаг**]

Тело цикла

Next [счётчик_цикла]

Счётчик_цикла – это числовая переменная. В начале выполнения цикла она принимает значение, задаваемое числовым выражением начало.

Числовое выражение конец задает заключительное значение счётчика цикла. Числовое выражение шаг не обязательно и по умолчанию его значение равно 1.

Тело цикла – это последовательность операторов, которая будет выполнена заданное число раз. Если шаг положителен, цикл завершится, когда впервые выполнится условие

счетчик_цикла > конец

Если шаг цикла отрицателен, условие его завершения

счетчик_цикла < конец

Цикл **For Each...Next**.

Оператор **For Each...Next** позволяет выполнить отдельную итерацию для каждого элемента коллекции или массива. Для коллекции используется синтаксис:

For Each элемент **in** коллекция

...

Next

В примере 4 рассматриваемая конструкция цикла используется для вывода имен всех рабочих листов в активной рабочей книге.

Пример 4

```
Public Sub Sheets_work()
Dim ws As Worksheet
Dim s As String
For Each ws In ActiveWorkbook.Worksheets
s = s & ws.Name & Chr(10)
Next
MsgBox s
End Sub
```



Синтаксис конструкции **For Each...Next** для элементов массива:

For Each элемент **in** массив

...

Next

Пример 5 – Использование оператора цикла **For Each...Next** для вычисления суммы элементов массива.

Предполагается, что массив **MyArray** создан ранее и представленный фрагмент кода процедуры используется для решения поставленной задачи.

```
Dim var As Single, total As Single
```

```
total = 0
```

```
For Each var in MyArray
```

```
total = total + var
```

```
Next
```

Операторы цикла, в основном, применяются для обработки массивов.

Массив – это множество фиксированных значений, называемых **элементами**, с общим именем и типом данных. Вся эта группа значений интерпретируется и обрабатывается как одна переменная.

Для обращения к конкретному элементу массива используются **индексы**.

Когда необходимо хранить множество связанных однотипных значений, обычно используется массив. Операторы цикла – часто используемое средство для обработки массивов.

Массивы характеризуются размером (количество элементов) и размерностью – одномерные, двумерные, ... (максимальная размерность массива в **VBA** – 60).

Если количество элементов массива фиксировано, то такие массивы являются статическими. Массивы, размер которых устанавливается в процессе работы программы, – динамические.

Объявление массивов.

VBA не обязывает разработчика программы объявлять переменные до их использования, тип устанавливается **Variant** и переменной отводится максимальный из возможных размер памяти. Массив – это такая переменная, которая обязательно объявляется для того, чтобы указать его границы или специальным образом обозначить его как динамический.

Одномерные массивы

Описание массивов.

Dim<имя массива> (<начальное значение индекса>**To**<конечное значение индекса>) **As** <тип элементов массива>

Обращение к элементу массива осуществляется следующим образом: указывается имя массива, а затем в круглых скобках – номер элемента в массиве.

Ввод массивов.

Массивы можно вводить как с листа **Excel**, так и используя функцию **InputBox**.



Пример 6 – Приведены различные варианты ввода массивов А, В, С.

For i=1 To 15

A(i)=Worksheets(1).Range("A" & i).Value

B(i)=Worksheets(1).Cells(i,2)

C(i)=InputBox("Введите " & i & "ый элемент массива")

Next i

Двумерные массивы

Описание массивов.

Dim<имя массива>(<начальное значение индекса по строкам> **To** <конечное значение индекса по строкам >, < начальное значение индекса по столбцам> **To** <конечное значение индекса по столбцам>) **As**<тип элементов массива>

Обращение к элементу массива.

Обращение к элементу массива осуществляется следующим образом: указывается имя массива, а затем в круглых скобках через запятую –номер строки и номер столбца, где размещен элемент в массиве.

Задание 1. Одномерные массивы

1 Создайте одномерный массив из 10 элементов, которые являются произвольными положительными и отрицательными числами. Формирование массива обеспечьте, используя **InputBox** или считывая данные, предварительно записав их на лист **Excel**.

2 Предусмотрите вывод сформированного массива на лист **Excel**.

3 Разработайте программу обработки массива (вариант (таблица 3.1) по указанию преподавателя), предусмотрите в программе вывод результатов на лист **Excel**, используя **MsgBox**.

Таблица 3.1

Номер варианта	Задача
1	Найти количество положительных чисел
2	Найти количество отрицательных чисел
3	Найти сумму положительных чисел
4	Найти сумму отрицательных чисел
5	Заменить отрицательные элементы нулями
6	Найти количество элементов массива больших заданному значению
7	Найти произведение положительных элементов массива

Задание 2. Двумерные массивы

1 Создайте двумерный массив размерностью 5×5, состоящий из элементов, которые являются произвольными положительными и отрицательными числами. Его ввод обеспечьте, используя **InputBox** или считывая данные, предварительно записав их на лист **Excel**.



- 2 Предусмотрите вывод сформированного массива на лист Excel.
- 3 Напишите программу для решения задачи в соответствии с вариантом. Задачи приведены в таблице 3.2.

Таблица 3.2

Номер варианта	Задача
1	Переставить элементы последней строки массива и побочной диагонали
2	Определить значение элемента массива с наименьшим отклонением от среднего арифметического всех элементов массива
3	Суммировать все элементы матрицы, лежащие выше главной диагонали
4	Возвратить произведение всех элементов массива, у которых один или оба индекса равны трем
5	Переставить первый столбец массива с диагональю
6	Вычислить произведение максимального и минимального значений элементов главной диагонали массива
7	Возвратить сумму всех элементов вектора с нечетными индексами
8	Вычислить минимальное значение среди элементов массива
9	Суммировать все элементы массива без главной диагонали
10	Заменить максимальный элемент массива средним значением элементов
11	Вычислить сумму элементов двумерного массива, расположенных до минимального элемента (отсчет элементов производить по строкам)
12	Сформировать одномерный массив В, где В(i) равно сумме четных элементов и расположенных в i-й строке матрицы
13	Сформировать одномерный массив, содержащий значения минимальных элементов в столбцах матрицы

Объявление динамических массивов

Dim <имя массива> () [As <тип элементов массива>].

После определения количества элементов массива выполняется его переопределение:

ReDim <имя массива> <задается размерность массива >.

Контрольные вопросы

- 1 Варианты синтаксиса оператора For...Next.
- 2 Особенности использования оператора For Each...Next.
- 3 Понятия массива. Размер и размерность массива.
- 4 Описание статических и динамических массивов.
- 5 Способы формирования массивов.



3.4 Лабораторная работа № 14. Пользовательские процедуры и функции

Цель: освоить способы создания, редактирования и сохранения процедур.

Процедура – это набор описаний и инструкций, сгруппированных для вычисления. Существует три типа процедур: **Sub**, **Function**, **Property**.

Имя процедуры всегда определяется на уровне модуля. В процедурах должны содержаться все исполняемые программы. Вложенность процедуры в другие процедуры не допускается.

Для того чтобы вставить процедуру, необходимо вызвать редактор **VBA** и в меню редактора **Вставка** выбрать подменю **Модуль**. Затем снова выбрать **Вставка** и в открывшемся подменю выбрать **Процедура**. Появляется диалоговое окно **Вставка процедуры**.

В окне **Имя** пишется название процедуры. В рамке **Тип** выбирается необходимая процедура, а в рамке **Область определения** – необходимая область определения для вашей процедуры.

Инструкция **Sub** описывает имя, аргументы и текст программы, составляющий тело процедуры **Sub**.

Синтаксис:

```
[Private | Public] [Static] Sub имя [(списокАргументов)]
    инструкции
    [Exit Sub]
    [инструкции]
End Sub
```

Дополнительные сведения

Процедуры **Sub**, тип которых не указан явно с помощью слов **Public** или **Private**, являются общими по умолчанию. Все выполняемые команды должны содержаться в процедурах. Не допускается определение процедуры **Sub** внутри другой процедуры **Sub**, **Function** или **Property**.

Инструкция **Exit Sub** приводит к немедленному выходу из процедуры **Sub**. Выполнение программы продолжается с инструкции, следующей за инструкцией, содержащей вызов процедуры **Sub**.

Подобно процедурам **Function**, процедура **Sub** является самостоятельной процедурой, которая может получать аргументы, выполнять последовательность инструкций и изменять значения своих аргументов. Однако в отличие от процедуры **Function**, которая возвращает значение, процедура **Sub** не может применяться в выражении.

Если в процедуре используются общие переменные (**Public**) и они получают там новые значения, то при вызове этой процедуры другой процедурой **Sub**, в которой используются эти переменные, они будут содержать уже полученные значения. А вызвать одну процедуру другой можно следующим способом.

Процедура **Sub** вызывается в выражении по своему имени, за которым следует список аргументов в скобках.



Для вызова процедур **Sub** можно применять два способа.
В первом случае используется ключевое слово **Call**.

Call *ИмяПроц(СписокАрг)*

Если процедура не получает аргументов, скобки не ставятся.

Call *ИмяПроц*

Во втором случае ключевое слово **Call** не используется – указывается лишь имя процедуры, при наличии аргументов они перечисляются без скобок.

ИмяПроц

ИмяПроц СписокАрг

Переменные, используемые в процедурах **Sub**, разбиваются на две категории: явно описанные внутри процедуры и не описанные внутри процедуры.

Переменные, которые явно описаны в процедуре (с помощью ключевого слова **Dim** или эквивалентного ему **Private**), всегда являются локальными для этой процедуры.

Переменные, которые используются, но явно не описаны в процедуре, также являются локальными, если они явно не описаны на более высоком уровне.

Чтобы создать переменную с областью видимости на уровне модуля, она объявляется с помощью ключевого слова **Dim** внутри этого модуля за пределами всех процедур. Такой уровень видимости применяется в случаях, когда доступ к переменной необходим нескольким процедурам этого модуля.

Пример 7 – Вычислите площадь параллелепипеда со сторонами **a, b, c**. Для этого создайте две процедуры – **Plosh** и **Argum**. В процедуре **Plosh** обеспечьте ввод **a, b, c** и вывод результата **S**, а в **Argum** – расчет площади **S**. При этом процедура **Plosh** должна получить результат **S** путем вызова процедуры **Argum**.

Решение

Public a As Integer

‘Описываются общие переменные в модуле за пределами всех процедур.

Public b As Integer, c As Integer, S As Integer

Public Sub Plosh()

a = Range("a1").Value

b = Range("a2").Value

c = Range("a3").Value

Call Argum 'Вызов процедуры Argum

Range("a5").Value = S

End Sub

Public Sub Argum()



```
S = a * b * c
MsgBox S
End Sub
```

Задание

Создайте одномерный массив из 10 элементов. Для ввода элементов массива используйте **InputBox**. Отсортируйте массив по возрастанию, а так же найдите сумму всех его элементов. Результаты выведите на рабочий лист Excel.

Решение

Для выполнения задания создайте процедуры:

- ввода элементов массива и вывода исходного массива на рабочий лист в соответствующую строку таблицы;
- сортировки элементов массива (пример алгоритма сортировки массива размерностью n приведен далее);
- нахождения суммы элементов массива;
- вывода элементов отсортированного массива на рабочий лист в соответствующую строку таблицы;
- головной программы (процедуры) для запуска в нужной последовательности процедур для решения задачи.

Пример алгоритма сортировки массива

```
FOR i = 1 TO (n-1)
FOR j = i + 1 TO n
IF a(j) > a(i) THEN
b = a(i)
a(i) = a(j)
a(j) = b
END IF
NEXT j
NEXT i)
```

Пользовательские функции

Процедура **Function** представляет собой набор команд, с помощью которого можно решить определенную задачу. Отличие процедуры **Function** от процедуры **Sub** состоит в том, что процедуры данного типа обязательно возвращают значение. При создании процедуры **Function** можно описать тип данных, который возвращает функция.

Синтаксис описания функции выглядит следующим образом:

```
Function имя ([аргументы] ) [As тип]
Инструкции
имя=выражение
End Function
```



1 Ключевые слова **Function** и **End Function** используются в описании любой функции.

2 Параметр **имя** определяет имя функции.

3 Необязательные параметры аргументы позволяют передать в функцию требуемые значения.

4 Параметр **As тип** позволяет явно задать тип данных, который возвращает функция. Если он опущен, то по умолчанию возвращается значение типа Variant.

5 Внутри функции обычно имеется инструкция имя=выражение, которая используется для задания возвращаемого значения. Обратите внимание, что имя функции стоит слева от знака равенства, а параметр выражение определяет значение, которое должно быть совместимым с объявленным в параметре **As тип** типом данных для функции.

Процедуры **Function** возвращают значение, поэтому они обычно используются при выполнении вычислений, например, можно создать функцию, которая возвращает длину окружности заданного радиуса:

```
Function Length_C (Radius As Double) As Double
    Rem 'Вычисление длины окружности
    Const Pi = 3.14159
    Length_C = 2 * Pi * Radius
End Function
```

Функция Length_C возвращает значение типа Double.

Внутри тела функции определяется константа Pi и выполняется расчет длины.

Обратите внимание, что в вычисляемом выражении участвуют аргумент Radius и константа Pi, а результат расчета присваивается переменной Length_C, которая является именем функции.

Пример процедуры, в которой производится обращение к приведенной выше функции:

```
Public Sub Start()
    Dim r As Double, L As Double
    r = InputBox("Введите радиус")
    L = Length_C (r)
    MsgBox "Result L=" & l & " r=" & r
End Sub
```

Контрольные вопросы

1 Назовите основное отличие подпрограммы от функции.

2 Сколько аргументов может иметь процедура?

3 Какую область видимости по умолчанию имеет процедура?

4 Необходимо ли заключать значения аргументов в скобки при вызове процедуры?



3.5 Лабораторная работа № 15. Переключатели

Цель: изучить свойства элемента управления **Переключатель**, использовать его для решения задач.

Краткие теоретические сведения

Элемент управления **OptionButton** (переключатель) создается с помощью кнопки **Переключатель (OptionButton)**. Он позволяет выбрать один из нескольких взаимоисключающих параметров или действий. Переключатели обычно отображаются группами, обеспечивая возможность выбора альтернативного варианта.

В таблице 3.3 приведены наиболее часто используемые свойства элемента управления **OptionButton**.

Основными событиями переключателя являются события **Click** и **Change**.

Таблица 3.3

Свойство	Описание
Value	Возвращает True, если переключатель выбран, и False – в противном случае
Enabled	Допустимые значения: True (пользователь может выбрать переключатель) и False (в противном случае)
Visible	Допустимые значения: True (переключатель отображается во время выполнения программы) и False (в противном случае)
Caption	Надпись, отображаемая рядом с переключателем

Пример 8 – Разработайте программу выполнения одной из двух арифметических операций над двумя числами по выбору пользователя. Исполняемая операция устанавливается за счет выбора соответствующего переключателя.

На рисунке 3.2 представлена пользовательская форма с элементами управления OptionButton. Далее приведена программа, демонстрирующая использование OptionButton.

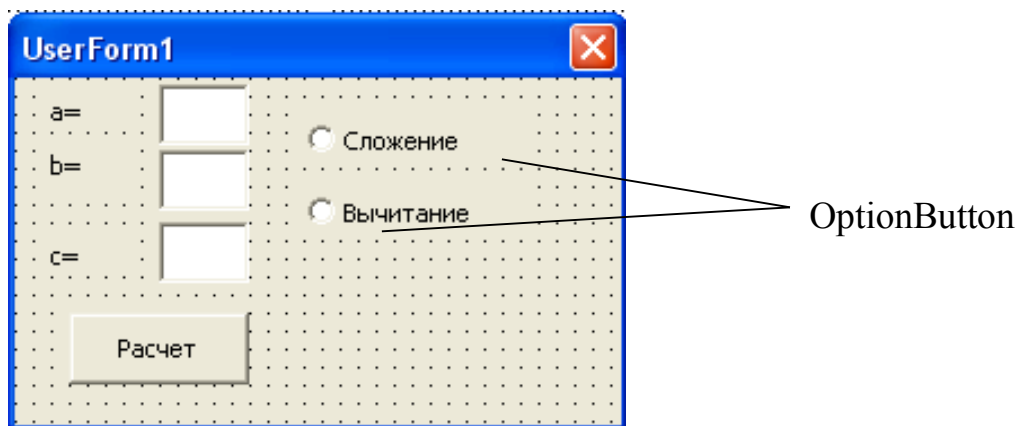


Рисунок 3.2 – Использование переключателей

Программа (процедура) для тестирования экранной формы UserForm1

```
Private Sub CommandButton1_Click()
Dim a As Integer, b As Integer, c As Integer
a = TextBox1.Value
b = TextBox2.Value
If OptionButton1.Value = True Then
c = a + b
End If
If OptionButton2.Value = True Then
c = a - b
End If
TextBox3.Value = c
End Sub
```

Задание

Создайте пользовательскую форму и напишите программу для решения следующей задачи с использованием *переключателей*.

Вариант задания по указанию преподавателя

1 Выбирается число от 1 до 4, определяющее пору года. Выведите название этой поры года.

2 Выбирается число от 1 до 7, определяющее день недели. Выведите название этого дня.

3 Выбирается число от 1 до 5. Дайте название этого числа.

4 Вводится нецелое число. Выведите либо его целую часть, либо дробную в зависимости от выбора пользователя.

5 Задано расстояние в метрах. Пересчитайте это расстояние в километрах, милях, футах или ярдах на выбор пользователя (1 миля = 1,609 километра, 1 метр = 1,094 ярда, 1 метр = 3,281 фута).

6 Дан объем в литрах. Пересчитайте этот объем в пинтах, галлонах, бушелях и квартах (английские меры объема жидких и сыпучих тел) на выбор пользователя (1 литр = 1,706 пинты, 1 литр = 0,220 галлона, 1 бушель = 36,35 литра, 1 кварта = 1,136 литра).

Контрольные вопросы

1 Назовите главное отличие между элементами управления CheckBox и OptionButton.

2 Назовите два способа создания группы переключателей OptionButton.



3.6 Лабораторная работа № 16. Списки

Цель: изучить свойства, события и методы элемента управления Список; использовать списки при решении задач.

Краткие теоретические сведения

Элемент управления **ListBox** (список) создается с помощью кнопки **Список** (ListBox). Применяется для хранения списка значений. Из списка пользователь может выбрать одно или несколько значений, которые в последующем будут использоваться в тексте программы. В таблице 3.4 приведены наиболее часто используемые свойства элемента управления **ListBox**.

Таблица 3.4

Свойство	Описание
ListIndex	Возвращает номер текущего элемента списка
ListCount	Возвращает число элементов списка
TopIndex	Возвращает элемент списка с наибольшим номером
ColumnCount	Устанавливает число столбцов в списке
Enabled	Допустимые значения: True (запрещен выбор значения из списка пользователем) и False (в противном случае)
Text	Возвращает выбранный в списке элемент
List	Возвращает элемент списка, стоящий на пересечении указанных строки и столбца. Синтаксис: List (row, column)
RowSource	Устанавливает диапазон, содержащий элементы списка
Clear	Удаляет все элементы из списка
RemoveItem	Удаляет из списка элементы с указанным номером. Синтаксис: Remove
AddItem	Добавляет элемент в список. Синтаксис: AddItem ([Item],[VarIndex])

Способы заполнения списка приведены в таблице 3.5.

Таблица 3.5

Способ заполнения	Алгоритм реализации способа
1	2
Поэлементно, если список состоит из одной колонки	<pre>With ListBox1 AddItem "Июнь" AddItem "Июль" AddItem "Август" End With</pre>
Массивом, если список состоит из одной колонки	<pre>With ListBox1 .List = Array("Июнь", "Июль", "Август") ListIndex = 1 End With</pre>



Окончание таблицы 3.5

1	2
Из диапазона A1 : B4, в который предварительно введены элементы списка. Результат выбора (индекс выбранной строки) выводится в ячейку C1	With ListBox1 ColumnCount = 2 RowSource = "A1:B4" ControlSource = "C1" End With
Поэлементно, если список состоит из нескольких колонок, например двух	With ListBox1 .ColumnCount = 2 .AddItem "Июнь" .List(0, 1) = "Сессия" .AddItem "Июль" .List(1, 1) = "Каникулы" .AddItem "Август" .List(2, 1) = "Каникулы" End With
Массивом, если список состоит из нескольких колонок, например двух	Dim A (2, 1) As String A(0, 0) = "Июнь" A(0, 1) = "Сессия" A(1, 0) = "Июль" A(1, 1) = "Каникулы" A(2, 0) = "Август" A(2, 1) = "Каникулы" With ListBox1 ColumnCount = 2

Задание

Разработайте программу, содержащую список. Вариант задания по указанию преподавателя.

1 Дан одномерный массив. Отсортируйте его. Выведите в один список – исходный массив, в другой – отсортированный.

2 Дан одномерный массив. Замените четные числа на 1, нечетные – на -0. Выведите в один список – исходный массив, в другой – преобразованный.

3 Дан одномерный массив. Выведите в один список – исходный массив, в другой – только элементы, кратные трем.

4 Вычислите $\frac{x^2}{2}, \frac{x^3}{3}, \dots, \frac{x^2}{10}$ для указанного значения x .

5 Выведите члены арифметической прогрессии. Значение первого члена, разность и количество членов задаются (формула n -го члена $a_n = a_1 + d(n - 1)$).

6 Выведите члены геометрической прогрессии. Значение первого члена, знаменатель и количество членов задаются ($b_n = b_1(q^n - 1)$).

Контрольные вопросы

1 Назовите свойства элемента управления **ListBox**, используемые для дополнения списка, удаления элементов и очистки списка.

2 Назовите два способа заполнения списка массивом.



3 Назовите способ списка заполнения поэлементно, если список состоит из нескольких колонок.

Список литературы

- 1 Информатика. Базовый курс : учебное пособие / Под ред. С В. Симоновича. – 3-е изд. – Санкт-Петербург : Питер, 2017. – 640 с. : ил.
- 2 **Кузин, А. В.** Базы данных : учебное пособие / А. И. Кузин, С. В. Левонисова. – 6-е изд., стер. – Москва : Академия, 2016. – 320 с.
- 3 **Советов, Б. Я.** Базы данных: теория и практика : учебник для бакалавров / Б. Я. Советов. – 2-е изд. – Москва : Юрайт, 2012. – 463 с.
- 4 **Царев, Р. Ю.** Информатика и программирование / Р. Ю. Царев. – Красноярск : Сибирский фед. ун-т, 2014. – 132 с.
- 5 **Гуриков, С. Р.** Информатика : учебник / С. Р. Гуриков. – Москва : ФОРУМ; ИНФРА-М, 2014. – 464 с.
- 6 **Каймин, В. А.** Информатика : учебник / В. Ф. Каймин. – Москва : ИНФРА-М, 2015. – 285 с.

