

**Методические рекомендации к практическим занятиям
МДК.01.02. Прикладное программирование
ПРОФЕССИОНАЛЬНОГО МОДУЛЯ
Разработка программных модулей программного обеспечения для
компьютерных систем**

Введение

Методические рекомендации к практическим занятиям по МДК.01.02. Прикладное программирование профессионального модуля разработана на основе Федерального государственного образовательного стандарта по специальностям среднего профессионального образования (далее – СПО) 230115 Программирование в компьютерных системах (базовой и углубленной подготовки).

Для успешного освоения курса студенты должны использовать знания, полученные в дисциплинах «Основы алгоритмизации и программирование», «Базы данных», «Информатика», «Информационные технологии», «Архитектура ЭВМ», «Операционные системы и среды».

Дисциплина “Прикладное программирование” предоставляет студентам возможность получить базовые знания и навыки по разработке приложений для прикладных пакетов. В практической части курса закрепляются теоретические знания и формируются практические навыки программирования с использованием исходного языка изучаемого пакета прикладных программ.

В результате изучения практического курса студенты должны

иметь представление:

- о возможностях, предоставляемых пакетом прикладных программ;
- о потенциальных возможностях встроенного языка программирования;

знать:

- состав и структуру изучаемого пакета прикладных программ;
- функциональное назначение основных компонентов прикладного пакета;
- синтаксические конструкции встроенного языка программирования, операторы, используемые типы данных, возможности языка программирования;
- объектно-ориентированные возможности изучаемого пакета прикладных программ;
- способы организации взаимодействия с внешними приложениями;

уметь:

- использовать предоставляемые прикладным пакетом возможности для решения конкретных задач;
- использовать интегрированные средства отладки и профилирования приложений

В данном курсе рассматриваются основные возможности пакета: Microsoft Office. Этот пакет рассматривается как один из универсальных пакетов прикладных программ, ориентированных на решение задач, связанных с подготовкой и обработкой документации.

Темы практических занятий

Номер занятия	Тема	Объем, ч
1	Работа с программами пакета MS Office: Word.	2
2	Работа с программами пакета MS Office: Excel.	2
3	Работа с программами пакета MS Office: PowerPoint.	2
4	Создание макроса и средства его вызова	2
5	Интегрированная среда разработки VBA	2
6	Знакомство с основными элементами управления при создании приложения в VBA.	2
7	Ввод, редактирование и отладка первой программы в среде VBA	2
8	Разработка и отладка линейных программ с процедурами ввода-вывода и оператором присваивания	2
9	Программирование ветвлений на языке VBA с использованием условного оператора	2
10	Программирование ветвлений на языке VBA с использованием оператора выбора	2
11	Разработка программ, содержащих операторы цикла	2
12	Работа с элементами одномерного массива	2
13	Работа с элементами многомерного массива	2
14	Разработка программ, содержащих функции	2
15	Объектная модель Word. Доступ к объектам Word из VBA.	2
16	Объектная модель Word. Создание сложного документа Word с помощью VBA.	2
17	Объектная модель Excel. Доступ к объектам Excel из VBA.	2
18	Объектная модель Excel. Работа с макросами в Excel.	2
19	Объектная модель Excel. Работа с таблицами.	2
20	Межпрограммные коммуникации	2

Практическое занятие №1. Работа с программами пакета MS Office: Word.

Цель работы:

1. Формирование структуры документа. Создание оглавления.
2. Освоение приемов работы с использованием стилей.
3. Освоение операций поиска и замены фрагментов текста.
4. Настройка меню и панелей инструментов.
5. Освоение способов интеграции объектов: вставка в текстовый документ рисунков, созданных в другом приложении.

После изучения этой темы студент должен:

знать:

- как формировать структуру документа Word;
- как в структурированном документе создать автосодержание;
- как настраивать меню и панели инструментов Word;
- способы интеграции объектов в текстовый документ.

уметь:

- решать типовые офисные задачи по созданию и редактированию документов с использованием основных возможностей WinWord.
- вставлять изображения, формулы и диаграммы в документ Word

Задание

Создайте документ Word, содержащий в себе (помимо основного текста) формулы, диаграммы, изображения. Документ должен быть структурирован. Каждому новому разделу должны быть даны названия. Каждый раздел должен начинаться с новой страницы. На первой странице должно находиться содержание всего документа. Каждая страница должна иметь нумерацию.

Методические рекомендации

Запустите текстовый процессор Word.

Документ Word может состоять из нескольких разделов. Больше одного раздела создается в том случае, если разные части документа должны различаться параметрами раздела, к которым относятся:

- размер и ориентация листа бумаги;
- поля на листе бумаги;
- нумерация страниц;
- расположение и тип колонтитулов;
- нумерация строк и т.д.

Для того чтобы создать новый раздел, нужно поставить курсор в начало нового раздела и с помощью команды Вставка-Разрыв установить переключатели с характеристиками нового раздела.

Для того чтобы создать автосодержание документа, необходимо все заголовки, которые должны входить в оглавление, отформатировать стилем Заголовок (Заголовок 1, Заголовок 2 и т. д.). После этого установите курсор в самом начале документа и создайте оглавление документа, используя команды меню Вставка - Оглавление и указатели.

Для добавления новых команд в меню сделайте следующие действия:

- Сервис-Настройка и на вкладке Команды выберите нужную категорию и команду;
- откройте эту категорию и перетащите мышью выделенную команду в нужное место

Для создания формулы в документе Word необходимо: установить курсор в место вставки формулы, ввести команду меню Вставка-Объект-Microsoft Equation 3.0 откроется редактор формул.

Литература:

[1] стр. 255-339; [2] стр. 160-188; [3] стр. 285-309; [4] стр. 13-45; [6] стр. 200-256.

Практическое занятие №2. Работа с программами пакета MS Office: Excel.

Цель работы:

1. Освоение приемов работы в табличном процессоре Excel.
2. Получение навыков практической работы по созданию и редактированию электронных таблиц.
3. Ознакомление с использованием формул и функций, выполнение вычислений.
4. Ознакомление с приемами работы по созданию и редактированию диаграмм.
5. Освоение способов интеграции объектов, созданных в различных приложениях:
 - вставка электронной таблицы и диаграммы в текстовый документ;
 - редактирование вставленной таблицы средствами Excel.

После изучения этой темы студент должен:

знать:

- как создавать и редактировать электронные таблицы;
- как вводить формулы и выполнять вычисления в таблицах;
- способы создания и редактирования диаграмм;
- способы интеграции объектов электронной таблицы и диаграмм в текстовый документ.

уметь:

- решать типовые офисные задачи по созданию и редактированию электронных таблиц с использованием основных возможностей Excel.
- вставлять изображения, формулы и диаграммы в книгу Excel.
- вставлять и редактировать таблиц и диаграмм в текстовый документ.

Задание. Реализовать в Excel биологическую модель развития популяции «хищник–жертва».

1. Построить модель неограниченного роста, в которой численность популяции ежегодно увеличивается на определенный процент.
Математическая модель: $x_{n+1} = a \cdot x_n$, где a - коэффициент роста.
2. Модель ограниченного роста – учитывается эффект перенаселенности, связанный с нехваткой питания, болезнями, т.д., который замедляет увеличение численности, $x_{n+1} = (a - b \cdot x_n) \cdot x_n$, где b - коэффициент перенаселенности
3. Модель ограниченного роста с отловом. Учитывается, что на численность популяций промысловых животных оказывает влияние величина ежегодного отлова, $x_{n+1} = (a - b \cdot x_n) \cdot x_n - c$, где c - коэффициент отлова
4. В модели «хищник–жертва» количество жертв x_n и количество хищников y_n связаны между собой, $x_{n+1} = (a - b \cdot x_n) \cdot x_n - c - f \cdot x_n \cdot y_n$, где f - возможность гибели жертвы при встрече с хищниками, $y_{n+1} = d \cdot y_n + e \cdot x_n \cdot y_n$, где d - скорость

уменьшения популяции хищников, e - характеризует величину роста численности хищников за счет жертв.

Построить график изменения популяций с течением времени.

Подготовить отчет (в текстовом документе) о зависимости численности популяций от времени, а также от начальной численности популяций и коэффициентов.

Методические рекомендации

1. В ячейки B1 и B6 вносятся начальные значения численности популяций жертв и хищников.

В ячейки B2:B5 вносятся значения коэффициентов a, b, c, f , влияющие на изменение численности жертв. В ячейки B7 и B8 вносятся значения коэффициентов d, e , влияющих на изменение численности хищников.

В столбце D будем вычислять численность популяции в соответствии с моделью неограниченного роста, в столбце E – ограниченного роста, в столбце F- ограниченного роста с отловом, в столбцах G,H- «хищник-жертва».

2. В ячейки D1,E1,F1,G1 внести значения начальной численности популяций жертв, в ячейку H1-хищников.

В ячейку D2 внести рекуррентную формулу ограниченного роста $=B\$2*D1$. В

ячейку E2 внести рекуррентную формулу ограниченного роста $=(B\$2-B\$3*E1)*E1$

Аналогично, в ячейку F2 вносится рекуррентная формула ограниченного роста с отловом. В ячейку G2 вносится рекуррентная формула изменения количества жертв. А в ячейку H2 вносится рекуррентная формула изменения количества хищников

3. Скопировать внесенные формулы в ячейки столбцов
4. Начальные данные можно взять, к примеру, следующие:

$$x_1 = 0.9 \quad y_1 = 0.8$$

$$a = 1.7 \quad b = 0.008 \quad c = 0.007$$

$$f = 0.03 \quad e = 0.003 \quad d = 0.04$$

Литература:

[1] стр. 341-358; [2] стр. 108-218; [3] стр. 315-328; [4] стр. 123-158; [6] стр. 309-356.

Практическое занятие №3. Работа с программами пакета MS Office: PowerPoint

Цель работы:

1. Освоение приемов работы в редакторе презентаций PowerPoint.
2. Получение навыков практической работы по созданию и редактированию слайдов.
3. Ознакомление с возможностями по созданию анимации.
4. Освоение способов интеграции объектов: вставка в презентацию рисунков, таблиц и диаграмм, созданных в другом приложении.
5. Формирование навыков использования анимации при показе слайдов.
6. Формирование навыков по созданию меню в презентации

После изучения этой темы студент должен:

знать:

- что такое презентация и как ее создать средствами PowerPoint;

- что такое слайд.
- как создаются и редактируются слайды.
- каким образом можно настроить анимацию и звуковое сопровождение элементу слайда.
- как вставить таблицу и диаграмму в презентацию.
- что такое управляющие кнопки и как их создать.

уметь:

- решать задачи по созданию и редактированию электронных таблиц использованием основных возможностей PowerPoint.
- вставлять изображения, таблицы и диаграммы в презентацию.
- вставлять и редактировать кнопки управляющего меню

Задание

Создайте презентацию объемом от 15 слайдов на любую интересующую тему. На каждом слайде установите кнопки перехода от слайда к слайду, а также настройте презентацию на автоматическое воспроизведение слайдов с интервалом в 0,5 минуты. И сохраните ее в формате пригодном для публикации в Интернете.

Методические рекомендации

Наша работа будет состоять из 15 слайдов. Для каждого из этих объектов задаются: фон, текст, рисунки (изображения), кнопки перехода слайда.

Кнопки. На каждом слайде будет размещено 3 кнопки: переход к следующему слайду, переход к предыдущему слайду, выход из презентации. Для предания им действия в главном меню Показ слайдов – Управляющие кнопки. Выбираем нужную кнопку. В окне Настройка действия – По щелчку мыши – Перейти по гиперссылке выбираем: следующий слайд, предыдущий слайд и т.д.

Если презентация подготовлена как слайд-фильм, тогда для каждого объекта и слайда следует определить время его нахождения на экране. Это можно сделать путем указания нужного времени на панели Переход слайда или с помощью Проведение репетиции, запустив опцию Настройка времени.

Для публикации презентации в Интернет необходимо сохранить файл в определенном формате (с расширением *.html) в меню Файл-Сохранить как web-страницу.

Литература:

[4] стр. 45-56; [6] стр. 413-451.

Практическое занятие №4. Создание макроса и средства его вызова

Цели работы:

1. Ознакомление с понятие макрос и его назначение, возможностями и способами создания
2. Формирование навыков по автоматизированному созданию макросов с помощью макрорекордера
3. Формирование навыков по воспроизведению макроса, а также по просмотру и редактированию кода макроса.

После изучения этой темы студент должен:

знать:

- для чего применяются макросы;
- структуру и порядок создания макроса в среде MS Office;
- назначение и возможности макрорекордера;
уметь:
- использовать макрорекордер в офисных приложениях.

Задание

С помощью макроса отформатировать отдельные слова и фразы с помощью полужирного, подчеркнутого шрифта Comic Sans MS 14-го размера. Цвет этих слов или фраз должен быть красный. Назначить данный макрос кнопки на панели инструментов.

Методические рекомендации

Выделим первый участок текста, который нужно отформатировать.
Выберем команду меню Сервис- Макрос - Начать запись. Появится окно, представленное на рисунке 1.

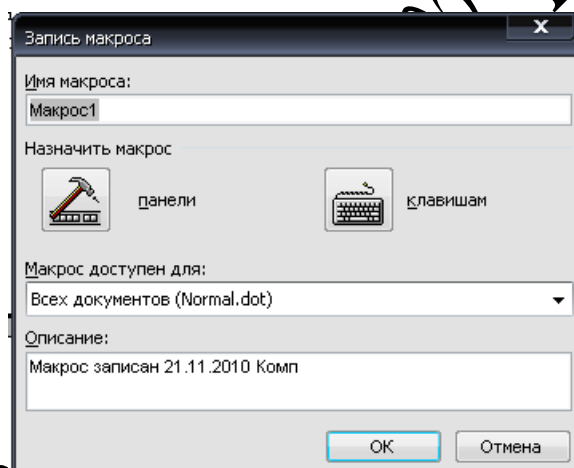


Рисунок 1. Вид окна "Запись макроса"

Рассмотрим поля этого диалога.

Имя макроса: в это поле нужно ввести имя макроса. Имена макросов должны начинаться с буквы, не должны содержать пробелов.

Назначить макрос панели: диалоговое окно назначения макроса настраиваемой панели.
Назначить макрос клавишам: диалоговое окно назначения макроса сочетанию клавиш на клавиатуре. Если назначается горячая клавиша для макроса, то впоследствии можно запускать этот макрос нажатием сочетания клавиш. Если макрос связывается командной кнопкой на панели инструментов, то - можно запускать данный макрос, щелкая кнопку на панели инструментов.

В поле Описание содержится описание макроса. Можно оставить его пустым.

Кнопка ОК начинает запись макроса.

Выполните следующие действия:

1. Из выпадающего списка "Шрифт" панели Форматирование выберите шрифт Comic Sans MS;
2. Из выпадающего списка "Размер шрифта" той же панели выберите размер 14;
3. Из выпадающего списка "Цвет шрифта" той же панели выберите цвет Красный;

4. Включите атрибут подчеркнутого шрифта, нажав на кнопку "Ц";
5. Включите атрибут жирного шрифта, нажав на кнопку "Ж".

Все необходимые действия произведены. Теперь запись макроса можно остановить. Для этого необходимо выбрать команду Сервис-Макрос-Остановить запись.

Назначение макроса кнопке панели инструментов

1. В меню Сервис выберите команду Настройка.
2. Откройте вкладку Команды.
3. В списке Категории щелкните Макросы.
4. Перетащите имя макроса из списка Команды в то место на панели инструментов, где требуется создать кнопку.
5. Щелкните Изменить выделенный объект и выберите соответствующие команды для изменения внешнего вида кнопки, таких как переименование, назначение изображения и так далее.

Литература:

[5] стр. 31-50; [7] стр. 15-41; [8] стр. 24-49.

Практическое занятие №5. Интегрированная среда разработки VBA

Цель работы:

1. Ознакомиться с назначением и возможностями интегрированной среды разработки
2. Ознакомиться с понятием проект VBA и его структурой.
3. Изучить оконные элементы VBA: редактор кода, редактор форм, инспектор объектов; окна отладки, наблюдения, вызовов.

После изучения этой темы студент должен:

знать:

- возможности, предоставляемые средой VBA;
- назначение основных элементов интерфейса VBA;
- структуру и типовой состав проекта на VBA;

уметь:

- загружать проект и управлять его свойствами;
- настраивать среду VBA;

Задание

Изучить интегрированную среду разработки VBA.

Методические рекомендации

Откройте окно Макросы нажатием соответствующей кнопки в меню Сервис, выберите команду редактор Visual Basic. Появится окно редактора:

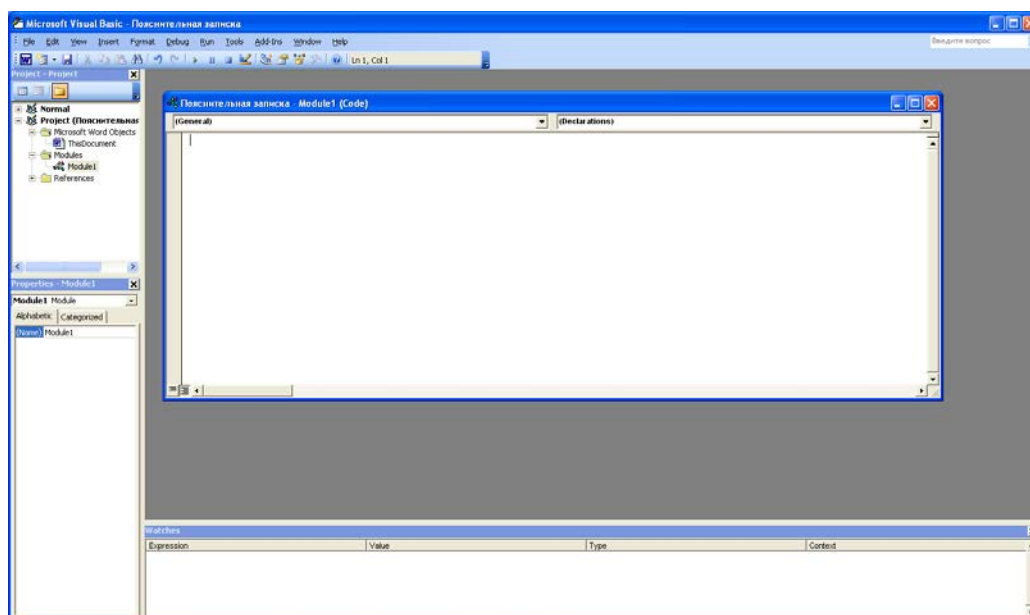


Рисунок 2. Окно редактора VBA

Окно редактора включает в себя несколько рабочих областей - окон, служащих для выполнения различных действий. В окне редактора присутствуют три рабочих области:

Code - это окно, в котором пишут тексты VBA-программ и редактируют макросы.

Project Explorer - оно отображает информацию о компонентах проекта, позволяет быстро перемещаться между компонентами.

Properties - отображает свойства выделенного объекта.

Помимо этого, редактор может содержать следующие элементы:

UserForm - служит для редактирования пользовательской формы в визуальном режиме.

Toolbox - содержит набор элементов управления (например - это кнопки, поля ввода и т.д.), которые можно добавлять на формы или в документы.

Object Browser - служит для просмотра информации об объектах, доступных в данном приложении.

Watch, Locals, Immediate - окна, средства которых используются при отладке приложений.

Рассмотрим главное меню этого окна.

File - Save — сохраняет файл.

File - Import File — позволяет импортировать внешний файл в редактор.

File - Export File — экспортирует данные из редактора во внешний файл.

File - Close and Return To Microsoft Word — закрывает VBA-редактор и возвращается в Microsoft Word (аналогичная команда есть и для MS Excel).

Edit — содержит команды для правки. Помимо стандартных команд отмены и возврата последнего действия (Undo, Redo), вырезания, копирования, вставки (Cut, Copy, Paste), поиска (Find), это меню содержит несколько особенных команд. В частности, это List Properties/Methods (Список Свойств/Методов) и другие.

View — содержит команды для отображения различных окон редактора. Названия команд соответствуют названиям окон.

Insert — служит для вставки в проект форм, модулей, процедур, файлов.

Format — служит для управления расположением элементов управления на формах.

Debug — содержит команды, позволяющие отлаживать программы.

Run — содержит команды для управления выполнением программ.

Tools — содержит средства для настройки свойств редактора, подключения дополнительных библиотек объектов.

Add-Ins — позволяет управлять дополнениями. По умолчанию это меню содержит лишь одну команду, запускающую менеджер дополнений.

Литература:

[5] стр. 31-50; [7] стр. 17-41.

Практическое занятие №6. Знакомство с основными элементами управления при создании приложения в VBA.

Цель работы:

1. Знакомство с формой и элементами управления VBA.
2. Формирование умений использования формы при создании приложений.
3. Формирование умений использования элементов управления при создании приложений.
4. Изучение основных свойств формы и элементов управления.

После изучения этой темы студент должен:

знать:

- Что такое форма;
- Какие элементы управления можно использовать в при создании приложения;
- Что такое объект;
- Для чего необходимо изменять свойства объекта.

уметь:

- Вставлять форму в проект своего приложения;
- Устанавливать на форму объекты;
- Изменять свойства объектов для повышения наглядности приложения.

Задание

Создайте проект. Добавьте к проекту форму и объекты: кнопка и надпись. Изучите их свойства.

Измените свойства этих трех элементов таким образом, чтобы получилось следующее:

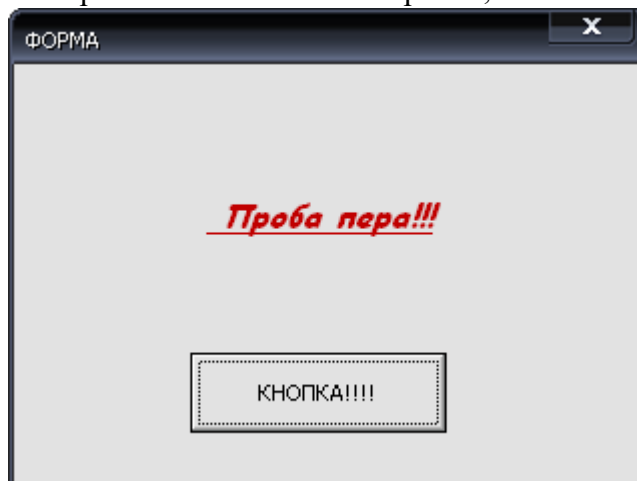


Рисунок 3. Первая форма

Методические рекомендации

В свой новый проект вставьте форму командой меню Insert - User Form:

Окно Toolbox содержит элементы управления, которые можно добавлять на форму. Элементы управления — это все то, что служит для управления работой программ. Например: кнопки, поля ввода и так далее.

Для того, чтобы добавить элемент управления на форму, нажмите в окне Toolbox кнопку с изображением этого элемента и выделите мышью ту область на форме, которую он должен занять.

Выделяя форму или кнопку, можно изучить основные свойства этих. Их можно увидеть в окне Properties.

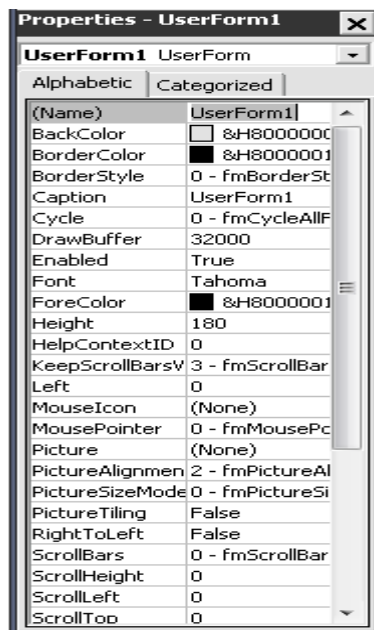


Рис. 4. Окно Properties

Важнейшими свойствами элементов управления являются:

Name — по имени к объекту можно обратиться из программного кода.

Caption — определяет надпись, которая видна на элементе управления.

Width, Height — определяют ширину и высоту объекта, которые измеряются в пикселях.

Top, Left — задают положение элемента на форме (относительно верхнего и левого угла формы соответственно).

Font — выбор гарнитуры, размера, начертания шрифта.

ForeColor — задает цвет шрифта элемента управления.

BackColor — позволяет задать цвет фона элемента управления.

Visible — видимость (значение свойства True) или невидимость (значение свойства False) объекта во время работы приложения.

Enable — активность (значение свойства True) или активность (значение свойства False) объекта во время работы приложения.

Для запуска приложения воспользуйтесь командой меню RUN- Run Sub/User Form.

Литература:

[5] стр. 31-50; [7] стр. 41-49; [8] стр. 221-258.

Практическое занятие №7. Ввод, редактирование и отладка первой программы в среде VBA

Цель работы:

1. Ознакомление со структурой программы в VBA.
2. Формирование умений по внесению изменений в код программы.
3. Знакомство с основными методами и событиями объекта.
4. Формирование умений применять события и методы объекта для создания приложений.

После изучения этой темы студент должен:

знать:

- какова структура программы VBA;
- что такое событие объекта;
- что такое методы объекта;
- как применять те или иные события объекта.

уметь:

- вносить изменения в код программы.
- правильно применять события или методы объекта для получения функционального приложения.
- запускать приложение на выполнение.

Задание

Измените текст предыдущей программы, таким образом, чтобы при нажатии на кнопку выходило сообщение «Это первая программа!». После этого кнопка становилась неактивной. А текст надписи на форме заменялся на: «Работа завершена»

Методические рекомендации

Откройте ваш предыдущий проект. Дважды щелкните мышью по кнопке. Откроется редактор кода. В нем будет автоматически создан каркас процедуры обработчика события Click для кнопки.

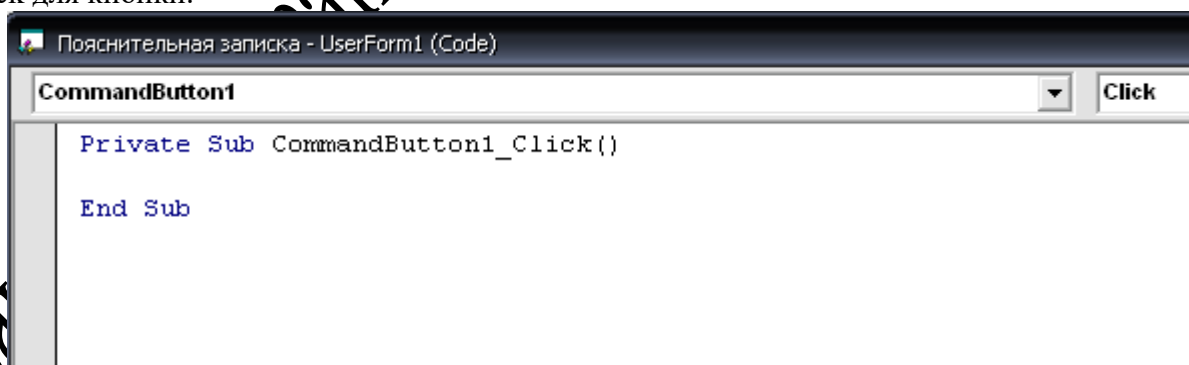


Рисунок 5. Пустая процедура в редакторе кода

Между этими двумя строчками и будем вносить необходимый код.

Программы на VBA управляются событиями. Обработчик — это подпрограмма, которая будет выполняться, если произойдет соответствующее ей событие. Событие Click возникает при нажатии на объект. То есть в результате нажатия на объект, срабатывает эта процедура и выполняется код программы, расположенный в ее теле.

Чтобы создать процедуру обработчика события для элемента управления или формы, выберите нужный элемент управления в левом поле, а в правом — название события.

Каждый элемент управления имеет уникальный набор событий, однако, в них не слишком сложно ориентироваться. На практике чаще всего используется лишь некоторые из них. Обычно события имеют осмысленные англоязычные названия.

Для того, чтобы при нажатии на кнопку вышло сообщение воспользуемся командой `msgbox`. Внесем это слово в редактор текста, при правильном вводе команды, редактор распознает ее и предложит контекстную подсказку.

В контекстной подсказке по `MsgBox` можно почерпнуть информацию о структуре вызова этой функции, о тех параметрах, которые ей можно передать. Параметры — это те данные, с которыми будет работать функция. Например, мы передаем `MsgBox` строку для вывода в окне сообщения. В контекстной подсказке по `MsgBox` эта строка называется `Prompt`. Она выделена жирным шрифтом — это значит, то, что будет сейчас введено, воспримется как `Prompt` — то есть как текст, который будет выведен в окне сообщения. Текст, который вы передаете функции, должен быть заключен в кавычки.

Если поставить после введенного текста запятую, в контекстной подсказке жирным выделится следующий параметр. Если вы хотите оставить этот параметр в значении по умолчанию, ставьте еще одну запятую, и сможете ввести значение для очередного параметра и т.д. Следующие параметры означают: какие кнопки будут видны у окна сообщений, заголовок этого окна, а также раздел справки, к которому стоит обратиться.

За активность кнопки отвечает параметр `Enable`. Поэтому следующей строчкой будет следующее: **`CommandButton1.Enabled = False`**

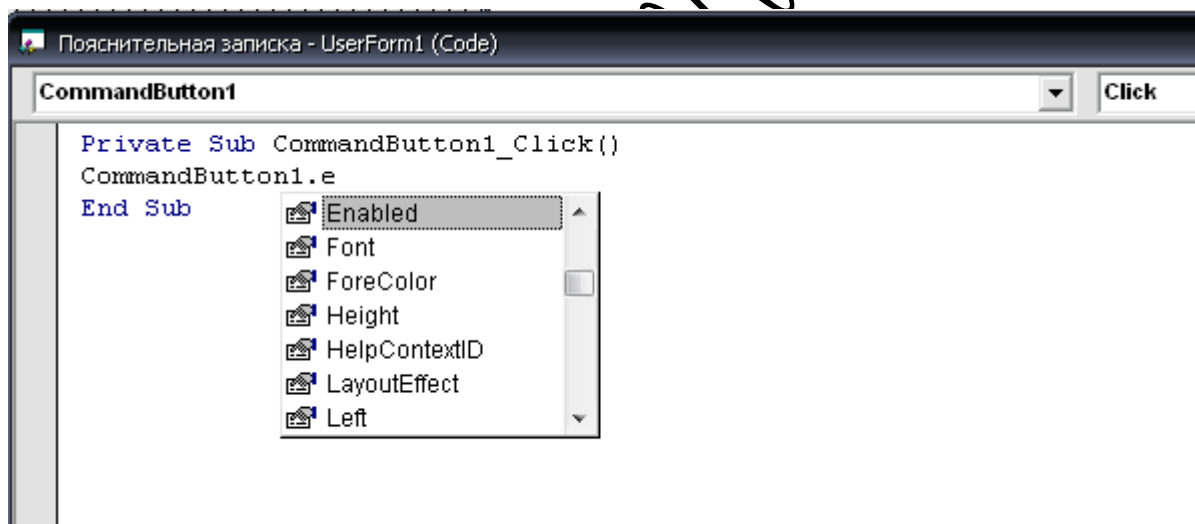


Рисунок 6. При наборе кода программы выходит контекстная подсказка

Текст надписи можно изменить, используя свойство `Caption` объекта `Label1`. то есть далее напишем следующее: **`Label1.Caption = "Работа завершена"`**

Литература:

[5] стр. 31-50; [7] стр. 41-49; [8] стр. 221-258.

Практическое занятие №8. Разработка и отладка линейных программ с процедурами ввода-вывода и оператором присваивания

Цель работы:

1. Ознакомить с синтаксисом языка: алфавит, идентификаторы, типы данных, управляющие структуры, процедуры и функции. Модули. Области видимости.

2. Сформировать навыки работы с переменными и типами данных.
3. Научить преобразовывать типы данных.
4. Ознакомить с процедурами ввода и вывода информации.

После изучения этой темы студент должен:

знать:

- основные возможности VBA;
- общий синтаксис языковых конструкций VBA;

уметь:

- писать программы на VBA, реализующие типовые линейные алгоритмы

Задание

Варианты задания должны соответствовать порядковому номеру студента в учебном журнале.

Составить программы для вычисления:

1. Периметра и площади прямоугольного треугольника по заданным длинам двух катетов a и b
2. Полусуммы кубов двух целых чисел c и d
3. Найти площадь равнобедренной трапеции с основаниями a и b и углом β при большем основании. Величину угла вводить в градусах
4. Среднего арифметического кубов двух целых чисел m и n
5. Известная длина окружности. Найти площадь круга, ограниченного этой окружностью
6. Найти площадь кольца с внутренним радиусом r и внешним радиусом R
7. Найти площадь треугольника по двум сторонам и углу между ними. Величину Угла вводить в градусах.
8. Найти катет прямоугольного треугольника, если известна гипотенуза и прилежащий к искомому катету угол.
9. Три сопротивления R_1 , R_2 , R_3 соединены параллельно. Найти их общее сопротивление
10. Найти дальность полёта тела, брошенного под углом к горизонту по формуле $S = V \cdot t \cdot \cos(a)$. Величину угла вводить в градусах

Методические рекомендации

Пример. Написать программу вычисления скорости (км/ч), с которой был преодолен маршрут, если известна длина маршрута (в км) и время (ч), затраченное на его прохождение

Решение. В редакторе Visual Basic создадим форму:

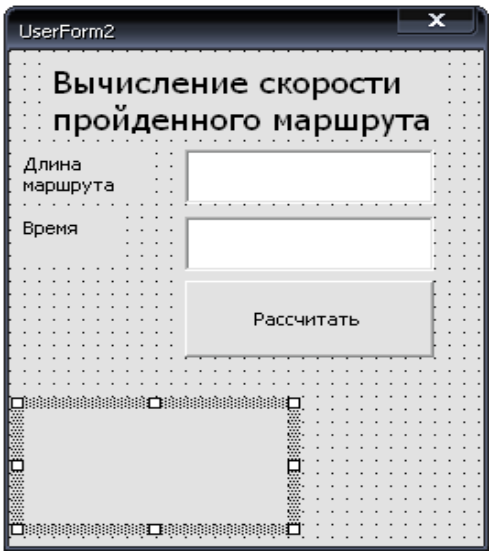


Рисунок 7. Форма

Основные свойства изменим следующим образом:

Таблица 1. Основные свойства компонентов

Компонент	Свойства	
	Caption	Visible
Label1	Вычисление скорости пройденного маршрута	True
Label2	Длина маршрута	True
Label3	Время	True
Label4		False
Edit1		True
Edit2		True
CommandButton1	Рассчитать	True

Остальные свойства измените по вашему усмотрению.
Дважды щелкнем по кнопке. В появившемся окне пропишем следующий код:

```
Private Sub CommandButton1_Click()  
    Dim s, v As Single 'объявляем переменные  
    's - пройденное расстояние  
    't - затраченное время  
    'v - скорость  
    'получаем значение расстояния, и присваиваем переменной s  
    s = TextBox1.Value  
    'получаем значение времени, и присваиваем переменной t  
    t = TextBox2.Value  
    'вычисляем значение скорости, и присваиваем переменной v  
    v = s / t  
    Label4.Visible = True 'делаем видимым компонент Label4  
    Label4.Caption = "Скорость прохождения маршрута " + Str(v) + " км/ч"  
    'выводим текст в компонент Label4  
End Sub
```

Пример работы программы представлен на рисунке Рисунок 8.

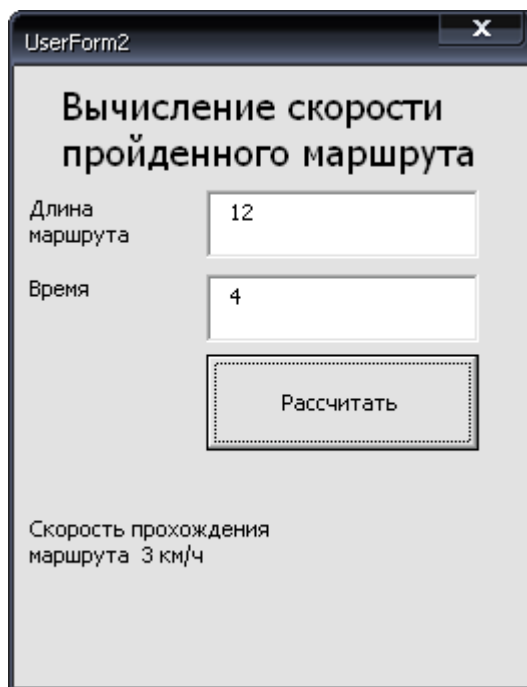


Рисунок 8. Пример программы

Литература:

[5] стр. 274-277; [7] стр. 83-105; [8] стр. 118-144.

Практическое занятие №9. Программирование ветвлений на языке VBA с использованием условного оператора

Цель работы:

1. Ознакомить с конструкцией оператора условного перехода.
2. Объяснить для чего нужны операторы условного перехода.
3. Научить применять условный оператор.

После изучения этой темы студент должен:

знать:

- основные возможности VBA;
- синтаксис конструкции оператора условного перехода VBA;

уметь:

- писать программы на VBA, реализующие алгоритмы ветвления.

Задание

Варианты задания должны соответствовать порядковому номеру студента в учебном журнале.

1. Составьте программу для упорядочения трех чисел a, b, c по возрастанию таким образом, чтобы имени a соответствовало наименьшее число, имени b - среднее, имени c – наибольшее
2. Составить программу, которая запрашивает пароль и выводит сообщение правильно ли он введен
3. Найти минимальное из трёх чисел
4. Даны действительные числа a, b, c. Вычислить корни квадратного уравнения

5. Составьте программу, в которой из трех введенных с клавиатуры чисел возводятся в квадрат положительные, а отрицательные остаются без изменения
6. Составьте программу нахождения произведения двух наибольших из трех введенных с клавиатуры чисел
7. Даны три различных числа, найти среднее из них. Средним назовем число, которое больше наименьшего из данных чисел, но меньше наибольшего
8. Вывести на экран номер четверти, которой принадлежит точка с координатами (x,y), при условии, что $(x > 0)$ и $(y > 0)$.
9. Найти наибольшее из трех данных чисел.
10. Даны целые числа a, b, c. Если $a \leq b \leq c$, то все числа заменить их квадратами, если $a > b > c$, то каждое число заменить наибольшим из них, в противном случае сменить знак каждого числа.

Методические рекомендации

Пример. Написать программу, которая запрашивает вес и рост, вычисляет оптимальное значение веса, сравнивая его с реальным и выводит соответствующее сообщение.

Решение. Как известно, для каждого человека существует оптимальное значение веса (кг), которое может быть вычислено по формуле: Рост (см) $\cdot 100$. Исходя из этого, напомним программу, проверяющую оптимален ли вес или нет.

В редакторе Visual Basic создадим форму:

Рисунок 9. Форма

Основные свойства изменим следующим образом:

Таблица 2. Свойства компонентов

Компонент	Свойства	
	Caption	Visible
Label1	Оптимален ли ваш вес	True
Label2	Введите ваш вес	True
Label3	Введите ваш рост	True
Label4		False
Edit1		True
Edit2		True

CommandButton1	Вычислить	True
----------------	-----------	------

Остальные свойства измените по вашему усмотрению.

Дважды щелкнем по кнопке. В появившемся окне пропишем следующий код:

```
Private Sub CommandButton1_Click()
    Dim w, h, opt, d As Single 'объявляем переменные
    'w - вес
    'h - рост
    'opt - оптимальный вес
    w = Val(TextBox1.Value) 'получаем значение веса, и присваиваем переменной w
    h = Val(TextBox2.Value) 'получаем значение роста, и присваиваем переменной h
    'вычисляем значение оптимального веса, и присваиваем переменной opt
    opt = h - 100
    Label4.Visible = True 'делаем видимым компонент Label4
    If w = opt Then
        Label4.Caption = "Ваш вес оптимален!!!" 'выводим текст в
компонент
    Else
        If w < opt Then
            Label4.Caption = "Надо поправится на " & Str(opt - w) & " кг"
'выводим текст в компонент Label4
        Else: Label4.Caption = "Надо похудеть на " & Str(w - opt) & " кг"
        End If
    End If
End Sub
```

Пример работы программы представлен на рисунке 10.

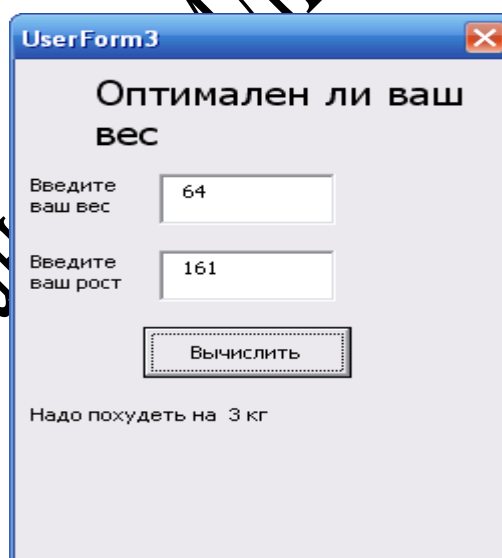


Рисунок 10. Пример работы программы

Литература:

[6] стр. 281-282; [7] стр. 152-164; [8] стр. 195-205.

Практическое занятие №10. Программирование ветвлений на языке VBA с использованием оператора выбора

Цель работы:

1. Ознакомить с конструкцией оператора выбора.

2. Объяснить для чего нужны операторы выбора.
3. Научить применять оператор выбора.

После изучения этой темы студент должен:

знать:

- основные возможности VBA;
- синтаксис конструкции оператора выбора VBA;

уметь:

- писать программы на VBA, реализующие алгоритмы ветвления с использованием оператора выбора.

Задание

Варианты задания должны соответствовать порядковому номеру студента в учебном журнале.

1. Дан номер месяца (1 — январь, 2 — февраль, ...). Вывести число дней в этом месяце для невисокосного года.
2. Дано целое число в диапазоне 0 – 9. Вывести строку — название соответствующей цифры на русском языке (0 — "ноль", 1 — "один", 2 — "два", ...).
3. Дано целое число в диапазоне 1 – 5. Вывести строку — словесное описание соответствующей оценки (1 — "плохо", 2 — "неудовлетворительно", 3 — "удовлетворительно", 4 — "хорошо", 5 — "отлично").
4. Арифметические действия над числами пронумерованы следующим образом: 1 — сложение, 2 — вычитание, 3 — умножение, 4 — деление. Дан номер действия и два числа А и В (В не равно нулю). Выполнить над числами указанное действие и вывести результат.
5. Единицы длины пронумерованы следующим образом: 1 — дециметр, 2 — километр, 3 — метр, 4 — миллиметр, 5 — сантиметр. Дан номер единицы длины и длина отрезка L в этих единицах (вещественное число). Вывести длину данного отрезка в метрах.
6. Единицы массы пронумерованы следующим образом: 1 — килограмм, 2 — миллиграмм, 3 — грамм, 4 — тонна, 5 — центнер. Дан номер единицы массы и масса тела M в этих единицах (вещественное число). Вывести массу данного тела в килограммах.
7. Робот может перемещаться в четырех направлениях ("С" — север, "З" — запад, "Ю" — юг, "В" — восток) и принимать три цифровые команды: 0 — продолжать движение, 1 — поворот налево, –1 — поворот направо. Дан символ С — исходное направление робота и число N — посланная ему команда. Вывести направление робота после выполнения полученной команды.
8. Локатор ориентирован на одну из сторон света ("С" — север, "З" — запад, "Ю" — юг, "В" — восток) и может принимать три цифровые команды: 1 — поворот налево, –1 — поворот направо, 2 — поворот на 180 градусов. Дан символ С — исходная ориентация локатора и числа N1 и N2 — две посланные ему команды. Вывести ориентацию локатора после выполнения данных команд.
9. Элементы окружности пронумерованы следующим образом: 1 — радиус (R), 2 — диаметр (D), 3 — длина (L), 4 — площадь круга (S). Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данной окружности (в том же порядке).
10. Написать программу, которая, используя дату сегодняшнего дня, вычисляет дату следующего дня

Методические рекомендации

Пример. Написать программу, которая запрашивает номер месяца и выводит соответствующее время года.

Решение. В редакторе Visual Basic создадим форму:

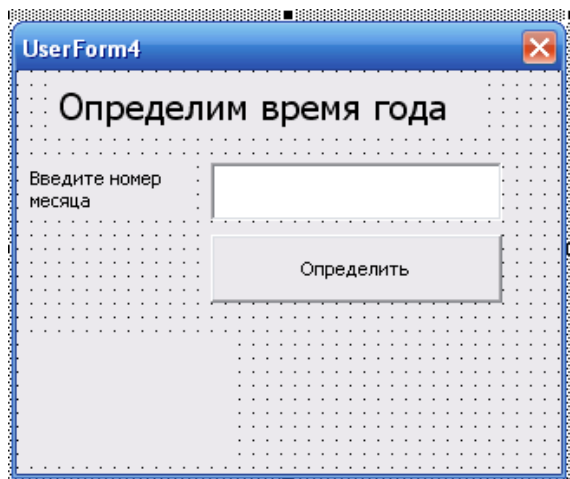


Рисунок 11. Форма

Основные свойства изменим следующим образом:

Таблица 3. Свойства компонентов

Компонент	Свойства	
	Caption	Visible
Label1	Определим время года	True
Label2	Введите номер месяца	True
Label3		False
Edit1		True
CommandButton1	Определить	True

Остальные свойства измените по вашему усмотрению.

Дважды щелкнем по кнопке. В появившемся окне пропишем следующий код:

```
Private Sub CommandButton1_Click()  
    Dim w As Integer 'объявляем переменные номер месяца  
    w = Val(TextBox1.Value) 'получаем значение номера месяца, и присваиваем  
    переменной w  
    Label3.Visible = True 'делаем видимым компонент Label3  
    Select Case w  
        Case 12, 1, 2 'если значение равно 12,1 или 2  
            Label3.Caption = "Зима" ' то выводим ЗИМА  
        Case 3 To 5 'если значение равно 3,4 или 5  
            Label3.Caption = "Весна" ' то выводим ВЕСНА  
        Case 6 To 8 'если значение равно 6,7 или 8  
            Label3.Caption = "Лето" ' то выводим ЛЕТО  
        Case 9 To 11 'если значение равно 9,10 или 11  
            Label3.Caption = "Осень" ' то выводим ОСЕНЬ  
        'если будет введено число не соответ ни одному из номеров месяца  
        Case Else  
            MsgBox ("Нет такого месяца") ' то выведем соотв сообщение
```

```
End Select  
End Sub
```

Пример работы программы представлен на рисунке 12.

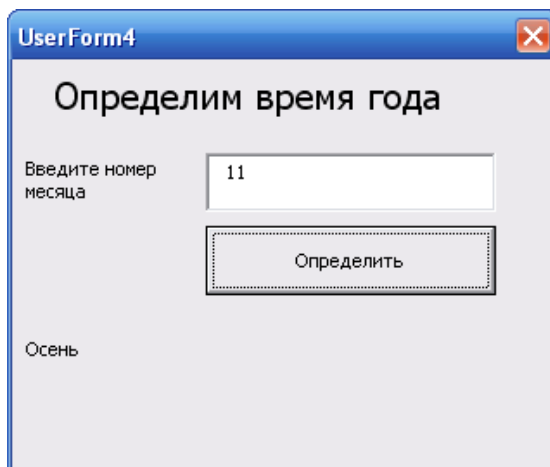


Рисунок 12. Пример работы программы

Литература:

[5] стр. 282; [7] стр. 164-173; [8] стр. 195-205.

Практическое занятие №11. Разработка программ, содержащих операторы цикла

Цель работы:

1. Ознакомить с конструкцией операторов цикла
2. Объяснить для чего нужны операторы цикла.
3. Научить применять операторы цикла.
4. Объяснить различие между различными видами оператора цикла

После изучения этой темы студент должен:

знать:

- основные возможности VBA;
- виды операторов цикла и их различие;
- синтаксис конструкций операторов цикла VBA;

уметь:

- писать программы на VBA, реализующие циклические алгоритмы с использованием операторов цикла.

Задание

Варианты задания должны соответствовать порядковому номеру студента в учебном журнале.

1. Введите два числа и найдите их произведение, используя только операцию сложения.
2. Введите два числа и найдите их частное от деления нацело (в переменной A), используя только операцию вычитания.
3. Вычислить сумму всех целых чисел от a до b
4. Вычислить произведение всех целых чисел от a до b

5. Дано вещественное число A (> 1). Вывести наименьшее из целых чисел N , для которых сумма $1 + 1/2 + \dots + 1/N$ будет больше A , и саму эту сумму.
6. Дано вещественное число A и целое число N (> 0). Вывести A в степени N : $A^N = A \cdot A \cdot \dots \cdot A$ (числа A перемножаются N раз).
7. Дано вещественное число A и целое число N (> 0). Вывести $1 + A + A^2 + A^3 + \dots + A^N$.
8. Дано вещественное число A и целое число N (> 0). Вывести $1 - A + A^2 - A^3 + \dots + (-1)^N A^N$.
9. Даны два целых числа A и B ($A < B$). Вывести все целые числа, расположенные между данными числами (включая сами эти числа), в порядке их возрастания, а также количество N этих чисел.
10. Даны два целых числа A и B ($A < B$). Вывести все целые числа, расположенные между данными числами (не включая сами эти числа), в порядке их убывания, а также количество N этих чисел

Методические рекомендации

Пример. Дано вещественное число A и целое число N (> 0). Вывести $1 + A + A^2 + A^3 + \dots + A^N$.

Решение. В редакторе Visual Basic создадим форму:

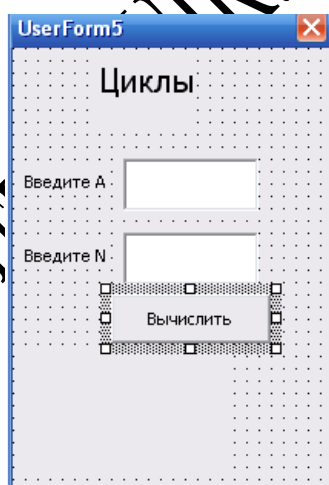


Рисунок 13. Форма

Основные свойства изменим следующим образом:

Таблица 4. Свойства компонентов

Компонент	Свойства	
	Caption	Visible
Label1	Циклы	True
Label2	Введите A	True
Label3	Введите N	True
Label4		False
Edit1		True

Edit2		True
CommandButton1	Вычислить	True

Остальные свойства измените по вашему усмотрению.

Дважды щелкнем по кнопке. В появившемся окне пропишем следующий код:
Данная программа реализуется с помощью цикла For-Next:

```
Private Sub CommandButton1_Click()  
    'объявляем переменные  
    Dim A, sum, p As Single  
    Dim N, i As Integer  
    'i - параметр цикла  
    'sum - хранит сумму  
    'p - хранит значение степеней A  
    A = Val(TextBox1.Value) 'получаем значение и присваиваем переменной A  
    N = Val(TextBox2.Value) 'получаем значение и присваиваем переменной N  
    Label4.Visible = True 'делаем видимым компонент Label4  
    sum = 1 'sum присваиваем начальное значение  
    p = 1 'p присваиваем начальное значение  
    For i = 1 To N  
        sum = sum + p  
        p = p * A  
    Next i  
    Label4.Caption = "Значение выражения " + Str(sum)  
End Sub
```

Изменим программу, таким образом, чтобы она реализовалась с помощью цикла While
Wend

```
Private Sub CommandButton1_Click()  
    'объявляем переменные  
    Dim A, sum, p As Single  
    Dim N, i As Integer  
    'i - параметр цикла  
    'sum - хранит сумму  
    'p - хранит значение степеней A  
    A = Val(TextBox1.Value) 'получаем значение и присваиваем переменной A  
    N = Val(TextBox2.Value) 'получаем значение и присваиваем переменной N  
    Label4.Visible = True 'делаем видимым компонент Label4  
    sum = 1 'sum присваиваем начальное значение  
    p = 1 'p присваиваем начальное значение  
    i = 1  
    While i <= N  
        sum = sum + p  
        p = p * A  
        i = i + 1  
    Wend  
    Label4.Caption = "Значение выражения " + Str(sum)  
End Sub
```

Пример работы программы представлен на рисунке 14.

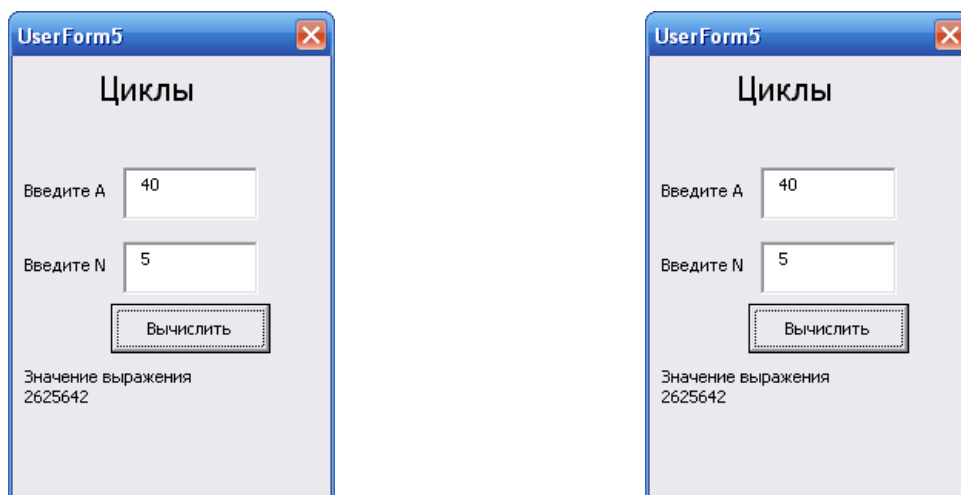


Рисунок 14. Пример работы программы

Литература:

[5] стр. 278-280; [7] стр. 181-203; [8] стр. 260-281.

Практическое занятие №12. Работа с элементами одномерного массива

Цель работы:

1. Объяснить что такое одномерный массив и для чего он нужен
2. Научить применять одномерные массивы.
3. Рассмотреть основные способы работы с одномерными массивами

После изучения этой темы студент должен:

знать:

- основные возможности VBA;
- для чего нужны одномерные массивы;
- как объявить и как обратиться к элементам массива;

уметь:

- писать программы на VBA, реализующие работу с одномерными массивами.

Задание

Варианты задания должны соответствовать порядковому номеру студента в учебном журнале.

1. Дан целочисленный массив размера 5. Вывести вначале все его четные а затем — нечетные
2. Дан целочисленный массив размера 5. Вывести вначале все его нечетные элементы, а затем — четные.
3. Заменить все положительные элементы целочисленного массива размера 5 на значение минимального.
4. Дан целочисленный массив размера 5. Вывести вначале его элементы с четными индексами, а затем — с нечетными.
5. Поменять местами минимальный и максимальный элементы целочисленного массива размера 5.
6. Дан целочисленный массив размера 5. Вывести его элементы в обратном порядке.

7. Дан целочисленный массив размера 5. Вывести вначале его элементы с нечетными индексами, а затем — с четными.
8. Заменить все отрицательные элементы целочисленного массива размера 5 на значение максимального.
9. Дан целочисленный массив размера 5. Преобразовать его, прибавив к четным числам первый элемент.
10. Дан целочисленный массив размера 5. Преобразовать его, прибавив к нечетным числам последний элемент.

Методические рекомендации

Пример. Дан массив размера 5. Найти максимальный элемент.

Решение. В редакторе Visual Basic создадим форму:

Рисунок 15. Форма

Основные свойства изменим следующим образом:

Таблица 5. Свойства компонентов

Компонент	Свойства	
	Caption	Visible
Label1	Поиск максимального элемента массива	True
Label2	a(1)=	True
Label3	a(2)=	True
...
Label7		False
Edit1		True
Edit2		True
...
CommandButton1	Вычислить	True

Остальные свойства измените по вашему усмотрению.

Дважды щелкнем по кнопке. В появившемся окне пропишем следующий код:

```
Private Sub CommandButton1_Click()  
    Dim a(1 To 5) As Integer 'объявляем массив
```

```
Dim i As Integer
Dim max As Integer ' максимальный элемент
Dim t As Integer 'номер максимального элемента
a(1) = Val(TextBox1.Value) 'получаем значение первого элемента массива и
присваиваем переменной a(1)
a(2) = Val(TextBox2.Value)
a(3) = Val(TextBox3.Value)
a(4) = Val(TextBox4.Value)
a(5) = Val(TextBox5.Value)
'предполагаем, что максимальным является первый элем массива
max = a(1)
'просмотрим весь массив, и если в нем есть элемент, который
'будет больше максимального, то переменной max присвоим это значение
'a также запомним номер, на котором найдем максим элемент
For i = 1 To 5
    If a(i) > max Then
        max = a(i)
        t = i
    End If
Next
Label7.Visible = True 'делаем видимым компонент Label7
Label7.Caption = "Максим элемент " + Str(max) + " находится на месте " +
Str(t)
End Sub
```

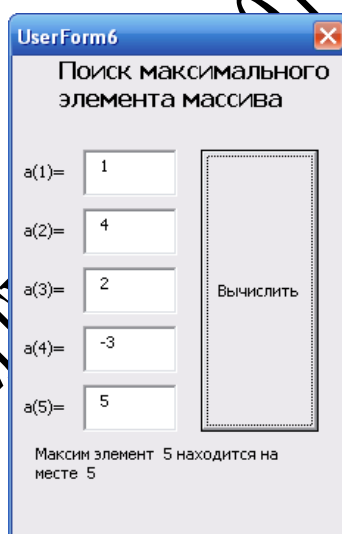


Рисунок 16. Пример работы программы

Литература:

[5] стр. 267-273; [7] стр. 71-83; [8] стр. 284-301.

Практическое занятие №13. Работа с элементами многомерного массива

Цель работы:

1. Объяснить что такое многомерный массив и для чего он нужен
2. Научить применять многомерные массивы.
3. Рассмотреть основные способы работы с многомерными массивами.

После изучения этой темы студент должен:

знать:

- основные возможности VBA;
- для чего нужны многомерные массивы;
- как объявить и как обратиться к элементам массива;
- что такое генератор случайных чисел и как его использовать при работе с массивами.

уметь:

- писать программы на VBA, реализующие работу с многомерными массивами.

Задание

Варианты задания должны соответствовать порядковому номеру студента в учебном журнале.

1. Дана квадратная матрица порядка М. Найти сумму элементов ее главной диагонали.
2. Дана квадратная матрица порядка М. Найти сумму элементов ее побочной диагонали.
3. Дана матрица размера 5 x 9. Найти суммы элементов всех ее четных строк.
4. Дана матрица размера 5 x 9. Найти суммы элементов всех ее нечетных строк.
5. Дана матрица размера 5 x 9. Найти суммы элементов всех ее четных столбцов.
6. Дана матрица размера 5 x 9. Найти суммы элементов всех ее нечетных столбцов.
7. Дана матрица размера 5 x 9. Найти минимальное значение в каждой строке.
8. Дана матрица размера 5 x 9. Найти минимальное значение в каждом столбце.
9. Дана матрица размера 5 x 9. Найти максимальное значение в каждой строке.
10. Дана матрица размера 5 x 9. Найти максимальное значение в каждом столбце.

Методические рекомендации

Пример. Дан двумерный массив размера 4 x 5. Определить сумму всех элементов третьей строки массива.

Решение. В редакторе Visual Basic создадим форму:

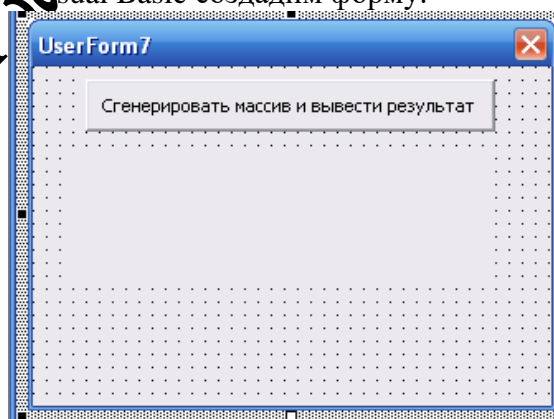


Рисунок 17. Форма

На форме содержаться следующие компоненты: CommandButton1 и Label1. Основные свойства изменим следующим образом:

Таблица 6. Свойства компонентов

Компонент	Свойства	
	Caption	Visible
Label7		False
CommandButton1	Сгенерировать массив и вывести результат	True

Остальные свойства измените по вашему усмотрению.

Дважды щелкнем по кнопке. В появившемся окне пропишем следующий код:

```
Private Sub CommandButton1_Click()
    Dim a(1 To 4, 1 To 5) As Integer 'объявляем массив
    Dim i, j As Integer
    Dim sum As Integer ' сумма элементов
    For i = 1 To 4
        For j = 1 To 5
            'для заполнения массива 4x5 случайными целыми числами от 0 до 10
            'можно написать такую строку
            'Rnd(k) генерирует случайные числа от 1 до k
            a(i, j) = Int(Rnd(1) * 10)
            Label1.Caption = Label1.Caption + " " + Str(a(i, j))
        Next j
        'для вывода следующей строки массива переведем строку
        Label1.Caption = Label1.Caption + Chr(13) + Chr(10)
    Next i
    Label1.Caption = Label1.Caption + Chr(13) + Chr(10)
    sum = 0
    For j = 1 To 5
        sum = sum + a(3, j)
    Next j
    Label1.Caption = Label1.Caption + "Сумма элементов 3-й строки равна " +
    Str(sum)
    Label1.Visible = True 'делаем видимым компонент Label1
End Sub
```

Пример работы программы представлен на рисунке 18.

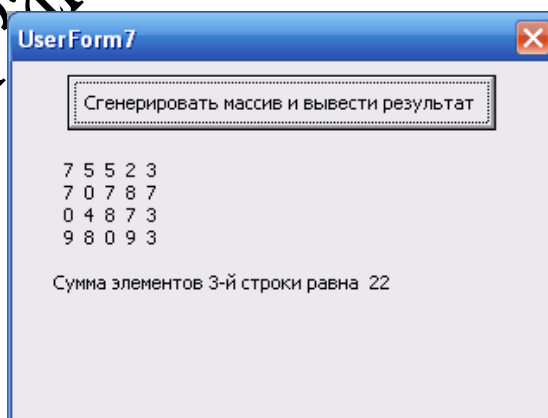


Рисунок 18. Пример работы программы

Литература:

[5] стр. 267-273; [7] стр. 71-83; [8] стр. 284-301.

Практическое занятие №14. Разработка программ, содержащих функции

Цель работы:

1. Объяснить что такое функция и для чего она нужна.
2. Научить создавать и применять функции
3. Научить задавать тип данных для функции

После изучения этой темы студент должен:

знать:

- Что такое функция;
- для чего нужна функция;
- как описать и как обратиться к функции;
- как установить тип данных для функции;

уметь:

- писать программы на VBA, реализующие работу с функциями.

Задание

Варианты задания должны соответствовать порядковому номеру студента в учебном журнале.

1. Составить программу нахождения наибольшего общего делителя нескольких чисел, используя функцию нахождения НОД двух чисел (алгоритм Евклида).
2. Даны действительные числа s, t . Получить:

$$F(t, -2s, 1.17) + F(2.2, t, s - t), \text{ где } F(x, b, c) = \frac{2a - b \sin c}{5 + |c|}$$

3. Даны действительные числа a, b . Получить $u = \min(a, b)$, $v = \min(ab, a+b^2)$, $\min(u + v, 3.14)$.
4. Описать функцию $\text{Min}(A, B)$ вещественного типа, находящую минимальное из двух вещественных чисел A и B . С помощью этой функции найти минимальные из пар чисел A и B , A и C и B , если даны числа A, B, C, D .
5. Описать функцию $\text{Max}(A, B)$ вещественного типа, находящую максимальное из двух вещественных чисел A и B . С помощью этой функции найти максимальные из пар чисел A и B , A и C , A и D , если даны числа A, B, C, D .
6. Даны два натуральных числа. Выяснить, в каком из них сумма цифр больше (определить функцию расчета суммы цифр натурального числа)
7. Определить значение $z = \max(a, 2b) * \max(2a - b, b)$.
8. Определить значение $z = \min(a, 3b) * \max(2a - b, 2b)$.
9. Определить значение $z = \text{sign}(x) + \text{sign}(y)$, где

$$\text{sign}(a) = \begin{cases} -1, & \text{if } a < 0 \\ 0, & \text{if } a = 0 \\ 1, & \text{if } a > 0 \end{cases}$$

10. Найти периметр треугольника, заданного координатами своих вершин (определить функцию для расчета длины отрезка по координатам его вершин)

Методические рекомендации

Пример. Вычислим значение сочетания $C_n^k = \frac{n!}{k! \cdot (n-k)!}$, где $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$

Решение. В редакторе Visual Basic создадим форму:

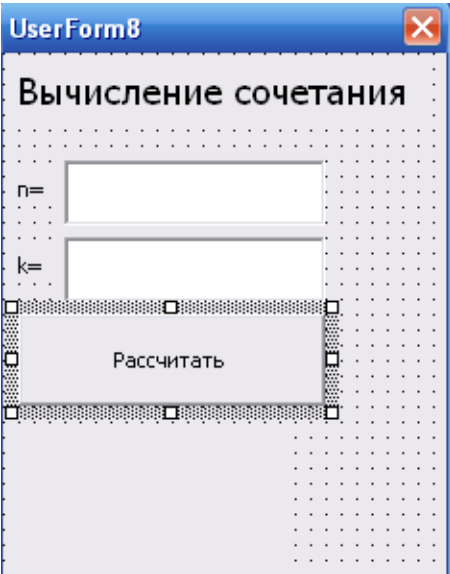


Рисунок 19. Форма

Основные свойства изменим следующим образом:

Таблица 7. Свойства компонентов

Компонент	Свойства	
	Caption	Visible
Label1	Вычисление сочетания	True
Label2	n=	True
Label3	k=	True
Label4		False
Edit1		True
Edit2		True
CommandButton1	Рассчитать	True

Остальные свойства измените по вашему усмотрению.
Дважды щелкнем по кнопке. В появившемся окне пропишем следующий код:

```
'описываем функцию
Function Fakt(m As Variant) As Variant
    Dim i As Integer
    Dim s As Long
    s = 1
    For i = 1 To m
        s = s * i
    Next
    Fakt = s
End Function

Private Sub CommandButton1_Click()
    Dim C, n As Variant 'объявляем переменные
    Dim k As Variant
    Dim p As Variant 'n!
```

```
Dim p1 As Variant 'k!  
Dim p2 As Variant '(n-k)!  
'получаем значение из поля TextBox1.Value, и присваиваем переменной n  
n = Val(TextBox1.Value)  
'получаем значение из поля TextBox2.Value, и присваиваем переменной k  
k = Val(TextBox2.Value)  
'вычислим n!  
p = Fakt(n)  
'вычислим k!  
p1 = Fakt(k)  
'вычислим (n-k)!  
p2 = Fakt(n - k)  
'вычислим сочетание  
C = p / (p1 * p2)  
Label4.Visible = True 'делаем видимым компонент Label4  
Label4.Caption = "Сочетание из " + Str(n) + " по " + Str(k) + " равно " +  
Str(C) 'выводим текст в компонент  
End Sub
```

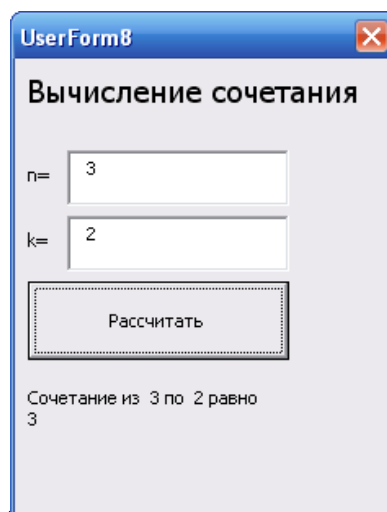


Рисунок 30. Пример работы программы

Литература:

[5] стр. 267-273; [8] стр. 144-190.

Практическое занятие №15. Объектная модель Word. Доступ к объектам Word из VBA.

Цель работы:

1. Научить использовать наборы для доступа к объектам Word;
2. Научить применять методы объекта Application
3. Научить применять свойства объекта Application
4. Научить применять события объекта Application

После изучения этой темы студент должен:

знать:

- способы доступа к объектам Word из VBA;
- основные методы, свойства и события объекта Application

уметь:

- программно обращаться к приложению Word.

Задание

Варианты задания должны соответствовать порядковому номеру студента в учебном журнале.

1. Имеется документ Word. Найти количество предложений, начинающихся с букв "А" или "а".
2. Имеется документ Word. В конце документа напечатать его пятое предложение.
3. Имеется документ Word. В конце документа напечатать его первое предложение.
4. Имеется документ Word. В конец шестого предложения записать новое предложение с текстом: «До свидания, люди!».
5. Имеется документ Word. В конец девятого предложения записать новое предложение с текстом: «Как прекрасен этот мир!».
6. Имеется документ Word. В выделенном предложении подсчитать количество символов.
7. Имеется документ Word. Найти самое длинное слово в выделенном предложении.
8. Имеется документ Word. После выделенного предложения вывести коды ее первого и последнего символа.
9. Имеется документ Word. Вывести выделенное предложение, содержащее те же символы, но расположенные в обратном порядке.
10. Имеется документ Word. Установить шрифт девятого предложения **Comic Sans MS**.

Методические рекомендации

Для программного обращения к документам Word необходимо знание основных объектных элементов.

Word (Microsoft Word) - чтобы вызывать объекты Microsoft Office Word из других приложений используется объект Word.

Application (Приложение) - объект, который включает в себя все остальные объекты и коллекции.

Documents (Документы) - коллекция, которая включает в себя объекты Document (Документ).

Для обработки отдельных составляющих документа применяются: Sections (разделы), Paragraphs (Абзацы), Sentences (Предложения), Words (Слова), Characters (Символы).

Например, чтобы перейти к 12 предложению можно записать `Sentences(12)`

Selection (Выделенная область) - представляет собой выделенную область документа или позицию, в которую будет осуществляться вставка очередного символа.

Range (Диапазон) - представляет собой диапазон документа, ограниченный начальным и конечным символом.

Помимо этого, для форматирования текста, можно применить объект Font (позволяет изменить свойства шрифта).

Вообще, для знакомства с примерами кодов, можно воспользоваться макрорекодером. То есть во время форматирования документа включите режим записи макроса. Затем проанализируйте полученный код.

Пример. Имеется документ Word. В выделенном тексте цвет всех заглавных букв изменить на красный, а строчных на зеленый.

```
Sub pr15()  
Dim fChar As Range  
Dim selText As Range  
Set selText = Selection.Range
```

```
If Selection.Type = wdSelectionIP Then
    MsgBox "Не выделен текст"
Else
    For Each fChar In selText.Words
        For i = 1 To fChar.Characters.Count
            If fChar.Characters.Item(i) Like "[A-ZА-Я]" Then
                fChar.Characters.Item(i).Font.Color = wdColorRed
            Else: fChar.Characters.Item(i).Font.Color = wdColorGreen
            End If
        Next i
    Next fChar
End If
End Sub
```

Литература:

[4] стр. 40-45; [5] стр. 167-174; [6] стр. 256-258; [8] стр. 513-530.

Практическое занятие №16. Объектная модель Word. Создание сложного документа Word с помощью VBA.

Цель работы:

1. Научить использовать наборы для доступа к объектам Word.
2. Научить получать данные из форм и применять эти данные при составлении документа.
3. Научить копировать материалы нескольких документов в один.

После изучения этой темы студент должен:

знать:

- способы доступа к объектам Word из VBA;

уметь:

- программно обращаться к приложению Word.
- работать сразу с несколькими документами Word.
- создавать пользовательские формы и использовать их для передачи данных в документ.

Задание

Предположим, что Вам необходимо ежедневно составлять договора купли-продажи по какому-либо конкретному товару. Все эти договора будут отличаться друг от друга только персональными данными, которые предоставляет покупатель. Предположим, что шаблон договора находится в документе Word (например, dogovor-prodaga.doc).

Ваша задача написать макрос, который:

1. запрашивает данные покупателя (Фамилия, Имя, Отчество, Адрес, паспортные данные, ИНН и т.д.)
2. создает новый документ Word
3. копирует текст договора (из документа dogovor-prodaga.doc)
4. Вставляет данные о покупателе, полученные из формы, в те места договора, где они необходимы.

Замечание. Перед началом работы подготовьте документ dogovor-prodaga.doc, в котором будет находиться шаблон договора купли-продажи.

Методические рекомендации

Первым делом необходимо создать форму, в которую будут вноситься персональные данные покупателя.

Для этого перейдем в редактор Visual Basic. И там создадим новую форму со всеми необходимыми полями, а также с кнопкой при нажатии на которую произойдет построение договора.



Рисунок 21. Форма приложения

При нажатии на кнопку, во-первых, должен создаваться новый документ. Для этого можно воспользоваться следующей конструкцией: Documents.Add.

При выполнении такого кода будет создан новый пустой документ. Для того, чтобы в дальнейшем можно было программно обратиться к данному документу нужно создать ссылку на него. Для этого можно предыдущую команду видоизменить на:

```
'объявляем переменную соответствующего типа
Dim obj_NewDoc As Word.Document
'присвоили этой переменной ссылку на новый объект
Set obj_NewDoc = Application.Documents.Add
```

Далее необходимо обратиться к документу с текстом договора (dogovor-prodaga.doc) и скопировать всё его содержимое.

Сначала программно откроем этот документ. Для этого воспользуемся командой

```
Dim obj_Dogovor As Document
Set obj_Dogovor = Documents.Open_
    FileName:="путь к файлу dogovor-prodaga.doc"
```

Затем выделим содержимое этого документа: obj_Dogovor.Content.Select

Копируем в буфер обмена: Selection.Copy

Активируем основной документ: obj_NewDoc.Activate

Вставляем в текущую позицию курсора материалы из буфера обмена: Selection.Paste

Закрываем документ с договором: obj_Dogovor.Close

ДОГОВОР КУПИ-ПРОДАЖИ № 1

г. Нефтекамск «...» 200 г.

ООО «Елка», именуемое в дальнейшем Продавец, в лице Генерального директора Иванова И. И. и **ФАМИЛИЯ ПОКУПАТЕЛЯ**, именуемое в дальнейшем Покупатель, с другой стороны, заключили настоящий Договор о нижеследующем:

1. Предмет договора

Рисунок 22. Фрагмент договора купли-продажи

На рисунке 22 представлен фрагмент типового договора купли-продажи.

Далее полученные данные из формы заносятся в договор. Например, Пусть в текстовом поле Fam, находящемся на форме содержится информация о фамилии покупателя. Исходя из визуального анализа документа, фамилию покупателя необходимо внести в 5 абзац вместо строки **ФАМИЛИЯ ПОКУПАТЕЛЯ**. То есть, можно выделить 5 абзац и заменить эту строку на содержимое текстового поля Fam. Сделать это можно с помощью следующего кода:

```
obj_NewDoc.Content.Select
Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = "ФАМИЛИЯ ПОКУПАТЕЛЯ"
    .Replacement.Text = Fam.Text
End With
Selection.Find.Execute
Selection.Find.Execute
With Selection
    If .Find.Forward = True Then
        .Collapse Direction:=wdCollapseStart
    Else
        .Collapse Direction:=wdCollapseEnd
    End If
    .Find.Execute Replace:=wdReplaceOne
    If .Find.Forward = True Then
        .Collapse Direction:=wdCollapseEnd
    Else
        .Collapse Direction:=wdCollapseStart
    End If
    .Find.Execute
End With
```

Аналогичным образом можно заполнить содержимое остальных полей.

Литература:

[4] стр. 40-45; [5] стр. 167-174; [6] стр. 256-258; [8] стр. 513-530.

Практическое занятие №17. Объектная модель Excel. Доступ к объектам Excel из VBA.

Цель работы:

1. Научить использовать наборы для доступа к объектам Excel;
2. Научить применять методы объектов Excel.
3. Научить применять свойства объектов Excel.
4. Научить находить используемые объекты.

После изучения этой темы студент должен:

знать:

- способы доступа к объектам Excel из VBA;
- как Visual Basic использует объекты.
- основные методы, свойства и события объектов Excel.

уметь:

- программно обращаться к книге Excel.

Задание

Написать макрос, который на листе Excel

1. изменяет цвет текста в ячейки M2
2. увеличивает значение ячейки M2 на 10
3. присваивает ячейки B6 значение $(M2+15)*2$
4. в ячейках E4:G10 записывает матрицу вида:
$$\begin{pmatrix} 1 & 2 & 3 \\ 10 & 20 & 30 \\ 100 & 200 & 300 \end{pmatrix}$$
5. в этой же книге создает новый лист, в который копируется содержимое первого листа.

Методические рекомендации

Для доступа к большинству объектов в Visual Basic используются наборы. Набор – это контейнер для всех объектов определенного класса. Например, набор Workbooks объединяет все текущие открытые рабочие книги, а Worksheets все листы в рабочей книге. Для доступа к ячейкам применяется объект Range.

Прделаем следующие действия.

Установим кнопку на лист. Для этого выбираем элементы управления

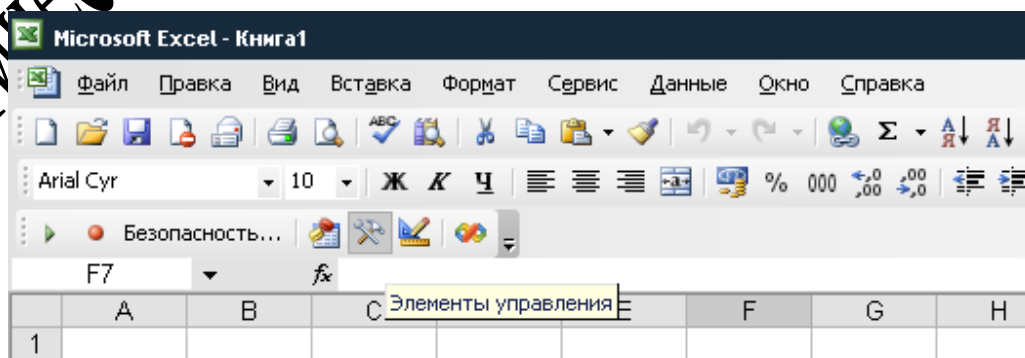


Рисунок 23. Выбираем элементы управления

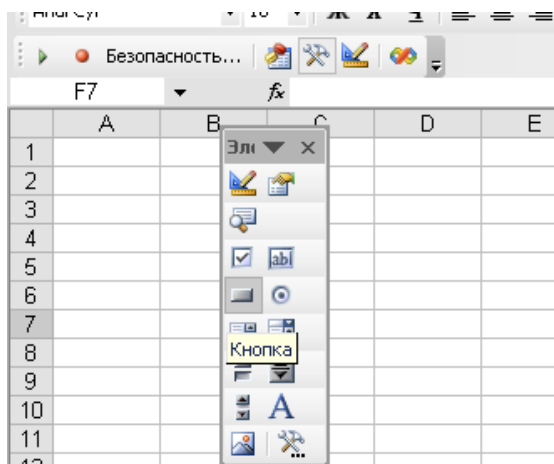


Рисунок 24. Выбираем кнопку

Рисуем компонент кнопки. Дважды щелкаем по ней. Появится процедура:

```
Private Sub CommandButton1_Click()  
  
End Sub
```

Вносим в нее строки

```
Dim n As Variant 'Раздел описания переменных  
Range("A1").Select 'Выделяем ячейку A1  
n = Range("A1").Value 'переменной n присваиваем значение ячейки A1  
ActiveCell.Offset(1, 0).Select 'спускаемся на одну ячейку вниз от активной ячейки  
ActiveCell.Offset(0, 12).Select 'переместимся на 12 ячеек вправо от активной ячейки  
ActiveCell.Value = n 'в активную ячейку вносим значение переменной n
```

В результате работы этого кода макроса копируется содержимое ячейки A1 в ячейку M2.

Все действия проводятся на активном листе активной книги. Чтобы явно обратиться к конкретному листу конкретной книги, то в код программы после описания переменных можно внести:

```
Workbooks("Книга1").Worksheets("Лист1").Activate
```

Этот код позволяет сделать активным Лист1 Книги1.

Литература:

[5] стр. 174-182; [6] стр. 361-362; [7] стр. 49-69; [8] стр. 396-441.

Практическое занятие №18. Объектная модель Excel. Работа с макросами в Excel.

Цель работы:

- Научить использовать наборы для доступа к объектам Excel;
- Научить применять методы объектов Excel.
- Научить применять свойства объектов Excel.
- Научить производить вычисления в Excel с использованием макросов.

После изучения этой темы студент должен:

знать:

- способы доступа к объектам Excel из VBA;
- как Visual Basic использует объекты.
- основные методы, свойства и события объектов Excel.

уметь:

- программно обращаться к книге Excel.
- определять конец данных в столбце.

Задание

Варианты задания должны соответствовать порядковому номеру студента в учебном журнале.

1. Имеется книга Excel. Вычислить сумму элементов, находящихся в первом столбце первого листа книги. Количество элементов не известно и может меняться по желанию пользователя. Результат вывести в первой пустой строке этого же столбца.
2. Имеется книга Excel. Вычислить количество элементов, находящихся в первом столбце первого листа книги. Количество элементов не известно и может меняться по желанию пользователя. Результат вывести в первой пустой строке этого же столбца.
3. Имеется книга Excel. Вычислить среднее арифметическое элементов, находящихся в первом столбце первого листа книги. Количество элементов не известно и может меняться по желанию пользователя. Результат вывести в первой пустой строке этого же столбца.
4. Имеется книга Excel. Вычислить произведение элементов, находящихся в первом столбце первого листа книги. Количество элементов не известно и может меняться по желанию пользователя. Результат вывести в первой пустой строке этого же столбца.
5. Имеется книга Excel. Вычислить среднее квадратическое отклонение элементов, находящихся в первом столбце первого листа книги по формуле

$$\sigma = \sqrt{\frac{\sum_{i=1}^n x_i^2}{n-1}}.$$

Количество элементов не известно и может меняться по желанию пользователя. Результат вывести в первой пустой строке этого же столбца.

6. Имеется книга Excel. Вычислить количество положительных элементов, находящихся в первом столбце первого листа книги. Количество элементов не известно и может меняться по желанию пользователя. Результат вывести в первой пустой строке этого же столбца.
7. Имеется книга Excel. Вычислить количество четных элементов, находящихся в первом столбце первого листа книги. Количество элементов не известно и может меняться по желанию пользователя. Результат вывести в первой пустой строке этого же столбца.
8. Имеется книга Excel. Вычислить количество элементов, кратных трем. Элементы находятся в первом столбце первого листа книги. Количество элементов не известно и может меняться по желанию пользователя. Результат вывести в первой пустой строке этого же столбца.
9. Имеется книга Excel. Найти максимальный элемент из элементов, находящихся в первом столбце первого листа книги. Количество элементов не известно и может меняться по желанию пользователя. Результат вывести в первой пустой строке этого же столбца.

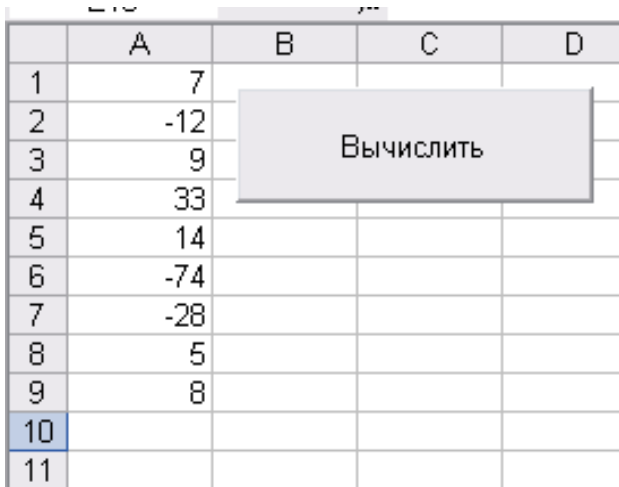
10. Имеется книга Excel. Вычислить количество нечетных элементов, находящихся в первом столбце первого листа книги. Количество элементов не известно и может меняться по желанию пользователя. Результат вывести в первой пустой строке этого же столбца.

Методические рекомендации

Пример. Имеется книга Excel. В первом столбце первого листа книги расположены некоторые числовые элементы. Заменить все отрицательные элементы из этого столбца на значение 0 и подсчитать количество сделанных замен. Количество значений в столбце не известно и может меняться по желанию пользователя. Результат вывести в первой пустой строке этого же столбца.

Решение.

Для начала откроем книгу Excel и создадим на ней кнопку. А также внесем в первый столбец числовые значения.

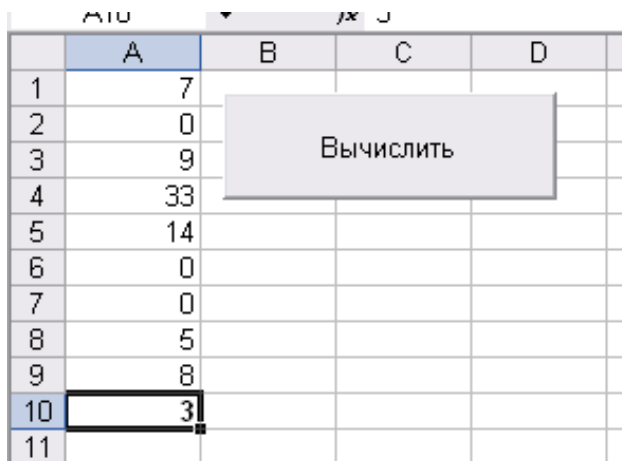


	A	B	C	D
1	7			
2	-12			
3	9			
4	33			
5	14			
6	-74			
7	-28			
8	5			
9	8			
10				
11				

Рисунок 25. Фрагмент таблицы с данными на листе Excel

Для кнопки пропишем следующий код:

```
Private Sub CommandButton1_Click()  
    'переменная для подсчета количества сделанных замен  
    Dim n As Integer  
    Workbook1.Worksheets("Лист1").Activate  
    Range("A1").Select  
    n = 0  
    'следующий цикл будет выполняться до тех пор, пока не будет найдена  
    'первая пустая ячейка. Условие можно прочитать так: «пока значение  
    'активной ячейки не является пустым значением»  
    While (IsEmpty(ActiveCell.Value) = False)  
        'следующее условие проверяет, является ли значение активной ячейки  
        'отрицательным. Если ответ «да», то значение этой ячейки заменится  
        'на 0, а к n добавляется единица.  
        If ActiveCell.Value < 0 Then  
            ActiveCell.Value = 0  
            n = n + 1  
        End If  
        ActiveCell.Offset(1, 0).Select  
    Wend  
    ActiveCell.Value = n  
    ActiveCell.Font.Bold = True  
End Sub
```

	A	B	C	D
1	7			
2	0			
3	9			
4	33			
5	14			
6	0			
7	0			
8	5			
9	8			
10	3			
11				

Рисунок 26. Пример работы программы

Литература:

[5] стр. 174-182; [6] стр. 361-362; [7] стр. 49-69; [8] стр. 396-441.

Практическое занятие №19. Объектная модель Excel. Работа с таблицами.

Цель работы:

1. Научить использовать наборы для доступа к объектам Excel.
2. Объяснить, как программно описывается создание и форматирование таблиц.
3. Объяснить, как программно описывается очистка листа.

После изучения этой темы студент должен:

знать:

- способы доступа к объектам Excel из VBA;
- программные способы создания и редактирования таблиц.

уметь:

- программно обращаться к приложению Excel.
- Создавать и редактировать таблицы.
- Производить программную очистку листа.

Задание

Предположим, существует некоторая фирма, выдающая деньги в долг под проценты на определенное количество месяцев. Возврат долга осуществляется равными частями. Необходимо составить таблицу возврата долга, по которой заемщик смог бы дату очередного платежа, сумму оплаты, а также остаток еще долга после этого платежа. Примерный вид таблицы представлен на рисунке 27.

Месяц	Ежемесячный платеж	Остаток
1	834,58р.	9 180,42р.
2	834,58р.	8 345,83р.
3	834,58р.	7 511,25р.
4	834,58р.	6 676,67р.
5	834,58р.	5 842,08р.
6	834,58р.	5 007,50р.
7	834,58р.	4 172,92р.
8	834,58р.	3 338,33р.
9	834,58р.	2 503,75р.
10	834,58р.	1 669,17р.
11	834,58р.	834,58р.
12	834,58р.	0,00р.
Итого	10 015,00р.	
Переплата	15,00р.	

Рисунок 27. Таблица погашения долга

В конце таблицы нужно вывести общую сумму долга (сумма долга плюс проценты), а также сумму переплаты.

Методические рекомендации

Для составления таблицы воспользуйтесь макрорекодером. Проанализируйте полученный код.

Для начала составьте таблицу, которая представлена на рисунке 28. А также разместите кнопку на листе. При нажатии на эту кнопку и будут происходить все вычисления и построение таблицы.

	A	B	C	D
1	Сумма долга	10 000,00р.		
2	Процентная ставка	15,00%		
3	Число месяцев	12		
4				
5				
6				
7				
8				
9				

Рисунок 28. Начальные данные в таблице

Первым делом при нажатии на кнопку у нас должна строиться таблица, в которой три столбца, а количество строк равно числу месяцев плюс один.

Для вычисления суммы ежемесячного платежа воспользуйтесь формулами:

$$\text{Общая сумма долга} = \text{сумма долга} + \frac{\text{сумма долга} \cdot \text{процент}}{100}$$

$$\text{ежемесячный платеж} = \frac{\text{общая сумма долга}}{\text{число месяцев}}$$

Формула для вычисления остатка:

$$\text{Остаток} = \text{общая сумма долга} - \text{выплаченная сумма}$$

Формула для вычисления суммы переплаты:

$$\text{Остаток} = \text{общая сумма долга} - \text{сумма долга}$$

Для очистки листа можно воспользоваться кодом

```
Range("A1:Z100").Select
Selection.ClearContents
```

Литература:

[5] стр. 174-182; [6] стр. 361-362; [7] стр. 49-69; [8] стр. 396-441.

Практическое занятие №20. Межпрограммные коммуникации

Цель работы:

Ознакомить с технологиями организации взаимодействия между приложениями.
Объяснить, как программно осуществить экспорт и импорт данных.
Осуществить обмен информацией между приложениями MS Office с помощью технологии OLE.

После изучения этой темы студент должен:

знать:

- технологии организации взаимодействия между приложениями;
- порядок использования в приложении MS Office внешних приложений.

уметь:

- вызывать внешние приложения из программы на VBA и обмениваться данными с ними.

Задание

Имеется книга Excel с некоторой таблицей. Необходимо переслать эти данные в новый документ Word.

Методические рекомендации

Межпрограммная коммуникация – это связь между двумя программами. Существует несколько стандартных методов организации межпрограммной коммуникации: DDE, OLE, COM и т.д.

Наша задача осуществить пересылку таблицы из книги Excel в новый документ Word.

Для начала создадим объект Word и дадим ссылку некоторой переменной на этот объект:

```
Set wa = CreateObject("Word.Application")
```

Теперь покажем созданный объект покажем на экране, добавив к объекту документ:

```
wa.Visible = True: Set wd = wa.Documents.Add
```

Далее будет необходимо реализовать выделение нужной таблицы. После чего выделенный диапазон нужно скопировать в буфер обмена.

Скопированные данные теперь можно будет вставить в новый документ Word. Для этого воспользуемся следующим кодом:

```
wd.Range.PasteExcelTable False, False, False  
Application.CutCopyMode = False
```

Литература:

[5] стр. 92-102; [7] стр. 454-477; [8] стр. 444-487.

Литература

1. Информатика. Серия «Учебники, учебные пособия»./Под ред. П.П.Беленького. – Ростов н/Д: Феникс, 2003.—448 с.
2. Информатика. Учебное пособие для среднего профессионального образования //Под ред. И. А. Черноскутовой – СПб.:Питер, 2005. – 272 с.
3. Информатика. Базовый курс. 2-е издание /Под ред. С. В. Симоновича. – СПб.:Питер, 2004. – 640 с.
4. Пакеты прикладных программ: Учеб. Пособие для сред. Проф. Образования /Э.Ф.Фуфаев, Л.И. Фуфаева. – М.: Издательский центр «Академия», 2004.- 352 с.
5. Хореев В.Д. Самоучитель программирования на VBA в Microsoft Office/.-К.: Юниор, 2001. – 320 с.
6. Романова Ю.Д. Информатика и информационные технологии: учебное пособие/ под ред. Ю.Д. Романовой.— 3-е изд., перераб. и доп. – М.:Эксмо, 2008. – 592 с.
7. Вильям Дж. Орвис Visual Basic For Application на примерах: Пер с. англ. – М.: БИНОМ. – 512 с.:
8. Кузьменко В.Г. Программирование на VBA 2002. – М.: ООО «Бином-Пресс», 2003 г. – 880 с.

Рецензия
на методические рекомендации к практическим занятиям
по дисциплине «Пакеты прикладных программ»
старшего преподавателя Башкирского экономико-юридического техникума
Стерляевой Лилии Венировны

Рецензируемые методические рекомендации предназначены для реализации в среднеспециальных образовательных учреждениях.

Методические рекомендации к практическим занятиям отвечает государственным требованиям к минимуму содержания и уровню подготовки по специальности 230105 «Программное обеспечение вычислительной техники и автоматизированных систем».

Целью создания методических рекомендаций является овладение необходимыми знаниями и практическими навыками по разработке приложений для прикладного пакета Microsoft Office.

В связи с поставленной целью задачей данной работы является ознакомление студентов с основными возможностями изучаемого пакета прикладных программ и его встроенного языка программирования.

Задачей также является обучить навыкам программирования в среде Visual Basic for Application для автоматизации работы с офисными документами.

Приведенный автором методический материал, а также список литературы поможет студентам довольно легко овладеть необходимыми знаниями и практическими навыками.

Преподаватель ФГОУ СПО «НМИК»



Ахметшина У.Д.