
**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)**

Кафедра ЭМИС

Вагнер Д.П.

Методические указания по проведению практических и
самостоятельных работ студентов по курсу
«Компьютерные технологии в управлении персоналом»

Томск 2018

Целью практических работ и индивидуальных заданий для самостоятельной работы по курсу «Компьютерные технологии в управлении персоналом» является закрепление теоретических знаний по курсу и овладение практическими навыками создания программ с использованием технологии программирования.

СОДЕРЖАНИЕ

Введение	4
ПРАКТИЧЕСКАЯ РАБОТА №1. Технология программирования. Структура программ.....	5
ПРАКТИЧЕСКАЯ РАБОТА №2. Технология программирования. Оператор условного перехода.....	12
ПРАКТИЧЕСКАЯ РАБОТА №3. Технология программирования. Создание элементов интерфейса программ.....	18
ПРАКТИЧЕСКАЯ РАБОТА №4. Технология программирования. Работа с массивами.....	20
Методические указания по самостоятельной работе.....	22
Список рекомендованной литературы	26

Введение

Цель практических работ — овладение практическими навыками создания программ с использованием технологии программирования, использования инструментов создания интерфейса для программ.

Методические указания к практическим работам по курсу «Компьютерные технологии в управлении персоналом» содержат только те понятия и определения, которые необходимы для их успешного выполнения.

ПРАКТИЧЕСКАЯ РАБОТА №1. Технология программирования.

Структура программ.

Цель работы: получение навыков по созданию программ на основе использования операторов ввода и вывода данных.

Темы для предварительного изучения. Структура программ. Операторы ввода-вывода информации.

Теоретические сведения

Будем осуществлять написание собственного кода на языке VBA в среде редактора Visual Basic либо его аналога Basic из пакета OpenOffice .

Структурным элементом программы, написанной на языке VBA, является модуль – совокупность объявлений и процедур, объединенных в единое целое.

Окно редактирования кода служит в качестве редактора для ввода и изменения кода внутри модуля. Каждый модуль состоит из *области описания* и одной или нескольких *процедур*. Процедура представляет собой последовательность *операторов*, которые часто называют *программными кодами*. Входящие в модуль процедуры объединены общей *областью описания*. В ней описываются данные и объекты, которые являются общепринятыми для процедур модуля.

При создании и редактировании кода удобно пользоваться командой **Insert (Вставить) / Procedure (Процедуру)**. В появившемся окне необходимо выбрать функцию или подпрограмму и задать ее имя. После этого будут автоматически сформированы операторы начала и конца процедуры и можно переходить непосредственно к набору операторов процедуры.

Для набора следующей процедуры в том же модуле необходимо повторить команду **Insert (Вставить) / Procedure (Процедуру)**. Если нужно

создать новый модуль повторяется команда **Insert (Вставить) / Module (Модуль)**.

Проверка правописания осуществляется на этапе компиляции командой **Debug (Отладка) / Compile (компилировать)**. Для запуска программы требуется выполнить команду **Run (Запуск)**.

В появившемся окне следует выбрать имя нужной программы и щелкнуть по кнопке Run (Выполнить).

Структура программного кода

На данном этапе мы будем создавать программы в рамках следующей синтаксической конструкции:

```
Sub <Имя> ([<Список аргументов>])  
[Dim <Имя> [As <Тип>]]  
[Const <Имя> [As <Тип>] = <Выражение>]  
[Инструкции]  
End Sub
```

В этой записи:

<Имя> – имя процедуры, удовлетворяющее стандартным правилам именования;

<Список аргументов> – список переменных, представляющий аргументы, которые передаются в процедуру при ее вызове. Имена переменных разделяются запятой.

Dim <Имя> [As <Тип>] – блок описания переменных;

Const <Имя> [As <Тип>] = <Выражение> – блок описания констант;

Инструкции – любой набор любых команд VBA;

Sub, End Sub – служебные операторы начала и окончания программы(процедуры).

Встроенные диалоговые окна

В программном коде на Basic ввод и вывод организуется с помощью встроенных диалоговых окон: окон ввода и окон сообщений. Окно сообщений (процедура MsgBox) выводит простейшие сообщения для

пользователя, а окно ввода (Функция `InputBox`) обеспечивает ввод информации.

Функция **`InputBox`** выводит на экран диалоговое окно, содержащее сообщение и поле ввода, устанавливает режим ожидания ввода текста пользователем или нажатия кнопки. Затем возвращает значение типа *String*, содержащее текст, введенный в поле. Синтаксис:

`InputBox (prompt, [, title] [, default] [, xpos] [, ypos] [, helpfile, context])`

Аргументы:

`prompt` – строковое выражение, отображаемое как сообщение в диалоговом окне.

`title` – строковое выражение, отображаемое в строке заголовка диалогового окна. Если этот аргумент опущен, в строку помещается имя приложения.

`default` – строковое выражение, отображаемое в поле ввода как используемое по умолчанию, если пользователь не введет другую строку. Если этот аргумент опущен, поле ввода изображается пустым.

`xpos, ypos` – числовое выражение, задающее расстояние по горизонтали и вертикали между верхней границей диалогового окна и верхним краем экрана. Если этот аргумент опущен, диалоговое окно помещается по вертикали примерно на одну треть высоты экрана.

`helpfile` – строковое выражение, определяющее имя файла справки, содержащего справочные сведения о данном диалоговом окне. Если этот аргумент указан, необходимо также наличие аргумента `context`.

`context` – числовое выражение, определяющее номер соответствующего раздела справочной системы. Если этот аргумент указан, необходимо также наличие аргумента `helpfile`.

Процедура **`MsgBox`** выводит на экран диалоговое окно, содержащее сообщение, устанавливает режим ожидания нажатия кнопки пользователем. Затем возвращает значение типа *Integer*, указывающее, какая кнопка была нажата.

Значение	Нажатая кнопка
1	ОК
2	Отмена
3	Стоп
4	Повтор
5	Пропустить
6	Да
7	Нет

Синтаксис:

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

Аргументы:



prompt – строковое выражение, отображаемое как сообщение в диалоговом окне.

button – числовое выражение, представляющее сумму значений, которые указывают число и тип отображаемых кнопок, тип используемого значка, основную кнопку. Значение этого аргумента по умолчанию равняется 0.

Значения аргумента, определяющие отображаемые кнопки:

Константа	Значение	Отображаемые кнопки
VbOKOnly	0	ОК
VbOKCancel	1	ОК, Отмена
VbAbortRetryIgnore	2	Стоп, Повтор, Пропустить
VbYesNoCancel	3	Да, Нет, Отмена
VbYesNo	4	Да, Нет
VbRetryCancel	5	Повтор, Отмена

Значения аргумента, определяющие отображаемые значки

Константа	Значение	Значок сообщения
VbCritical	16	
VbQuestion	32	
VbExclamation	48	
VbInformation	64	

Демонстрационные примеры

Приведем пример использования встроенных диалоговых окон.

```
Sub Тест()
```

```
Dim msg As String, style As Integer, title As String, x
```

```
Dim default
```

```
msg = "Мои первые шаги в VB!"
```

```
title = "Сообщение с одной кнопкой"
```

```
MsgBox msg, , title
```

```
title = "Сообщение с двумя кнопками"
```

```
style = vbYesNo
```

```
MsgBox msg, style, title
```

```
title = "Сообщение с двумя кнопками и значком"
```

```
style = vbYesNo + vbCritical
```

```
MsgBox msg, style, title
```

```
msg = "Введите число, кратное 16, не больше 64!"
```

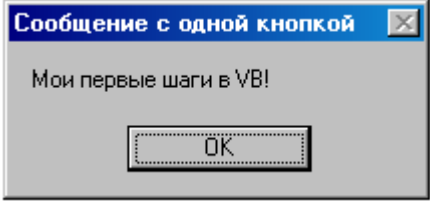
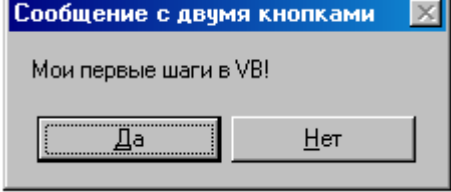
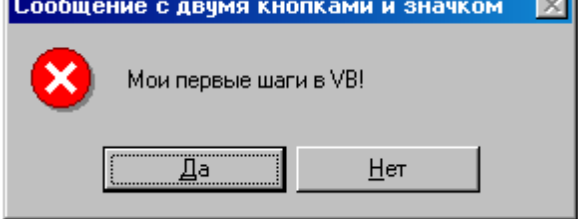
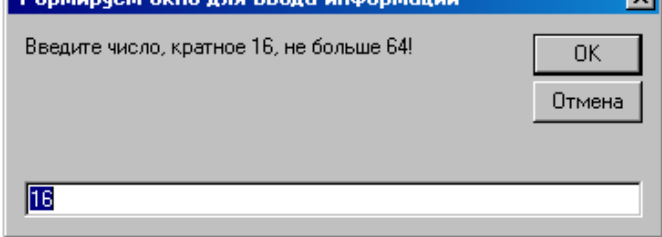
```
title = "Формируем окно для ввода информации "
```

```
default = 16
```

```
x = InputBox (msg, title, default)
```

```
End Sub
```

В результате выполнения приведенной программы на экране последовательно будут появляться окна. В таблице отражены окна, соответствующие определенным участкам кода. Отметим, что значок, который будет изображен в последнем окне, определяется значением переменной *x*, вводимым с клавиатуры при появлении окна “Формируем окно для ввода информации”. В нашем случае было введено значение, равное 48.

<pre>msg = "Мои первые шаги в VB!" title = "Сообщение с одной кнопкой" MsgBox msg, , title</pre>	
<pre>title = "Сообщение с двумя кнопками" style = vbYesNo MsgBox msg, style, title</pre>	
<pre>title = "Сообщение с двумя кнопками и значком" style = vbYesNo + vbCritical MsgBox msg, style, title</pre>	
<pre>msg = "Введите число, кратное 16, не больше 64!" title = "Формируем окно для ввода информации" default = 16 x = InputBox (msg, title, default)</pre>	

Приведенная программа может быть переписана следующим образом:

```
Sub Тест1()
```

```
Dim x as Byte
```

```
MsgBox "Мои первые шаги в VB!", , "Сообщение с одной кнопкой"  
MsgBox "Мои первые шаги в VB!", vbYesNo, "Сообщение с двумя  
кнопками"  
MsgBox "Мои первые шаги в VB!", vbYesNo + vbCritical, "Сообщение с  
двумя кнопками и значком"  
x = InputBox ("Введите число, кратное 16, не больше 64!", "Формируем  
окно для ввода информации ", 16)  
MsgBox " Мои первые шаги в VB!", vbYesNo + x, " Мое сообщение"  
End Sub
```

Постановка задачи

Задание 1. Организуйте ввод с клавиатуры двух чисел. Окна ввода должны иметь различные заголовки, появляться в разных частях экрана, содержать разные сообщения и неравные значения в поле ввода по умолчанию.

Требуется найти сумму, произведение, частное и остаток от деления первого числа на второе. Результаты операций вывести на экран. Все окна сообщений должны иметь различное количество кнопок и разные значки.

Задание 2. Напишите программу, осуществляющую простейший диалог с пользователем. Вначале появляется окно ввода, значением по умолчанию в котором является строка, содержащая ваше имя. После обработки полученных данных, на экране должно появиться окно сообщения с текстом приветствия, содержащего обращение по имени, введенному ранее.

Задание 3. Напишите простейший тест для пользователя состоящий из двух вопросов разных типов, в конце выведите на экран итоговое сообщение с количеством правильных ответов.

ПРАКТИЧЕСКАЯ РАБОТА №2. Технология программирования.

Оператор условного перехода.

Цель работы: получение навыков по созданию алгоритмов и программ на основе использования операторов условного перехода.

Темы для предварительного изучения. Операторы условного перехода. Алгоритм работы операторов условного перехода.

Теоретические сведения

В практических задачах часто бывает необходимо обеспечить выбор одного из альтернативных путей работы алгоритма в зависимости от результатов проверки условия. В этом случае говорят о базовой алгоритмической конструкции – ветвлении. *Оператор ветвления(условного перехода)* позволяет проверить некое условие и в зависимости от результатов проверки выполнить то или иное действие, т.е. он является средством ветвления вычислительного процесса.

Синтаксис оператора:

```
If <условие> Then  
    <инструкции_1>  
    [Else  
        <инструкции_2>  
End If]
```

Здесь **If**, **Then**, **Else**, **End If** – зарезервированные слова VBA (если, то, иначе, конец ветвления);

<условие> – произвольное выражение логического типа;

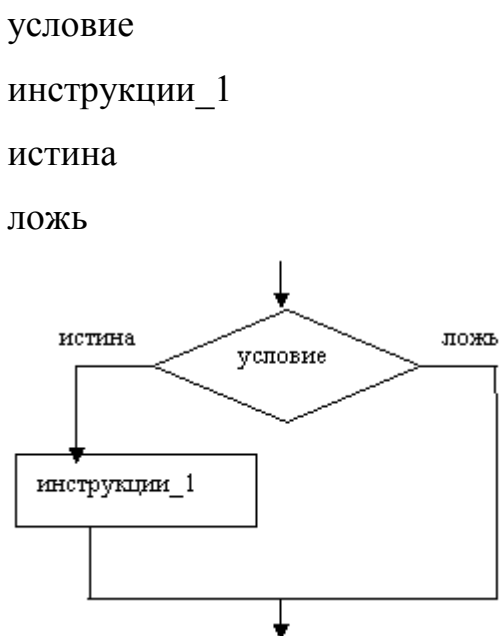
< инструкции_1>, < инструкции_2> – любые операторы языка VBA.

Условный оператор работает по следующему алгоритму. Вначале вычисляется значение условного выражения <условие>. Если результат есть **True** (истина), то выполняются <инструкции_1>, а <инструкции_2>

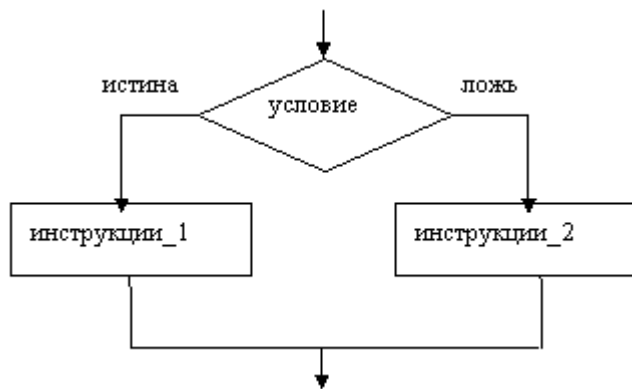
пропускаются. Если результат есть **False** (ложь), наоборот, <инструкции_1> пропускаются, а выполняются <инструкции_2>.

Часть условного оператора, начинающаяся со слова Else, может быть опущена, в этом случае имеет место *неполная форма условного оператора*. При этом, если условное выражение имеет значение True, выполняются инструкции, стоящие за Then; в противном случае весь оператор пропускается.

Оператор условного перехода в виде блок схемы:



условие
инструкции_1
ложь
истина
инструкции_2



Допустимо так же использование вложенных ветвлений. В этом случае синтаксис таков:

```

If <условие_1> Then
    <инструкции_1>
    Elseif <условие_2> Then
    <инструкции_2>
    .....
    [Else
    <инструкции_n>
    End If]
  
```

Пример реализации программы

Задача. Даны значения двух величин. Выбрать большее из них.

Обозначим данные величины через А и В, результат – С.

Условием, по которому разветвляется алгоритм, является отношение неравенства – логическое выражение. *Логические выражения* описывают некоторые условия, которые могут выполняться или не выполняться. Логическое выражение может принимать только два значения True (истина) или False (ложь).

Если логическое условие справедливо, то результатом будет логическая величина True и выполнение алгоритма продолжится по стрелке

“истина”]; в противном случае логическое выражение примет значение False и выполнение алгоритма пойдет по ветви “ложь”.

Соответствующая программа на VBA имеет следующий вид.

```
Sub Больше_из_двух ()  
Dim A As Integer, B As Integer, C As Integer  
A= Val (InputBox (“Введите число”, ”Ввод числа”))  
B= Val (InputBox (“Введите число”, ”Ввод числа”))  
If A>=B Then  
    C=A  
Else  
    C=B  
End If  
MsgBox “Наибольшее число ” & C , ”Вывод результата”  
End Sub
```

Постановка задачи

Задание 1. Создать программу, решающую следующую задачу согласно выданному преподавателем варианту.

Варианты задач для лабораторной работы

1. Даны две точки $A(x_1, y_1)$ и $B(x_2, y_2)$. Составить алгоритм, определяющий, которая из точек находится ближе к началу координат.
2. Даны три действительные числа. Возвести в квадрат те из них, значения которых неотрицательны, и в четвертую степень — отрицательные.
3. Даны два угла треугольника (в градусах). Определить, существует ли такой треугольник. Если да, то будет ли он прямоугольным.
4. Определить, является ли треугольник со сторонами a, b, c равносторонним.

5. Дано трехзначное число N . Проверить, будет ли сумма его цифр четным числом.
6. Определить, является ли треугольник со сторонами a , b , c равносторонним.
7. Определить, является ли целое число N четным двузначным числом.
8. Определить, является ли треугольник со сторонами a , b , c прямоугольным.
9. Определить, является ли треугольник со сторонами a , b , c равнобедренным .
10. Подсчитать количество отрицательных чисел среди чисел a , b , c .
11. Подсчитать количество положительных чисел среди чисел a , b , c .
12. Написать программу нахождения суммы большего и меньшего из 3 чисел.
13. Определить, является ли целое число N нечетным трехзначным числом.
14. Определить правильность даты, введенной с клавиатуры (число — от 1 до 31, месяц — от 1 до 12). Если введены некорректные данные, то сообщить об этом.

Задание 2. Создать программу, решающую следующую математическую задачу, на основе введенного пользователем значения переменной x согласно выданному преподавателем варианту.

Варианты задач для лабораторной работы

$$1. \quad F(x) = \begin{cases} x^2 - 3x + 9, & x \leq 3 \\ \frac{1}{x^3 + 6}, & x > 3 \end{cases}$$

$$3. \quad F(x) = \begin{cases} 9, & x \leq -3 \\ \frac{1}{x^2 + 1}, & x > -3 \end{cases}$$

$$5. \quad F(x) = \begin{cases} -3x + 9, & x \leq 7 \\ \frac{1}{x - 7}, & x > 7 \end{cases}$$

$$7. \quad F(x) = \begin{cases} x^2, & 0 \leq x \leq 1 \\ 4, & x > 1 \text{ или } x < 0 \end{cases}$$

$$9. \quad F(x) = \begin{cases} x^2 - x, & x > 1 \\ x^2 - \sin x, & x \leq 1 \end{cases}$$

$$11. \quad F(x) = \begin{cases} \sin x, & x \geq 9 \\ \frac{x^2}{x - 9}, & x < 9 \end{cases}$$

$$13. \quad F(x) = \begin{cases} 3x - 9, & x \leq 7 \\ \frac{1}{x^2 - 4}, & x > 7 \end{cases}$$

$$2. \quad F(x) = \begin{cases} -x^2 + 3x + 9, & x \geq 3 \\ \frac{x}{x^3 - 6}, & x < 3 \end{cases}$$

$$4. \quad F(x) = \begin{cases} 0, & x \leq 1 \\ \frac{1}{x + 6}, & x > 1 \end{cases}$$

$$6. \quad F(x) = \begin{cases} 0, & x \leq 1 \\ \frac{1}{x + 6}, & x > 1 \end{cases}$$

$$8. \quad F(x) = \begin{cases} x^2 + 4x + 5, & x \leq 2 \\ \frac{1}{x^2 + 4x + 5}, & x > 2 \end{cases}$$

$$10. \quad F(x) = \begin{cases} \sin x, & x < 0 \\ \cos x, & x \geq 0 \end{cases}$$

$$12. \quad F(x) = \begin{cases} \frac{1}{5 - x^2}, & x < -5 \text{ или } x > 5 \\ \cos x, & -5 \leq x \leq 5 \end{cases}$$

$$14. \quad F(x) = \begin{cases} x^2 - 3x + 9, & x \leq 3 \\ \frac{1}{x^3 + 6}, & x > 3 \end{cases}$$

ПРАКТИЧЕСКАЯ РАБОТА №3. Технология программирования.

Создание элементов интерфейса программ.

Цель работы: получение навыков по созданию элементов интерфейса для программ с помощью форм.

Темы для предварительного изучения. Интерфейс пользователя, создание форм и элементов управления.

Постановка задачи

Задание 1. Создайте форму «Калькулятор» с возможностью ввода двух чисел с клавиатуры. Пример формы для вычисления суммы и разности двух чисел приведен на рисунке 11.1

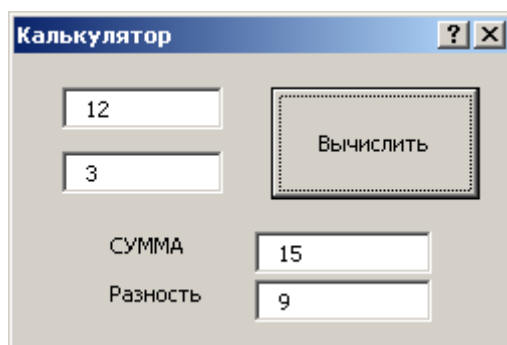


Рисунок 11.1 Форма 1

Также создайте поля сумма, разность, произведение, частное. Результаты операций должны выводиться на экран при нажатии кнопки «Вычислить».

Задание 2. Измените форму «Калькулятор», чтобы она приняла вид, приведенный на рисунке 11.2:

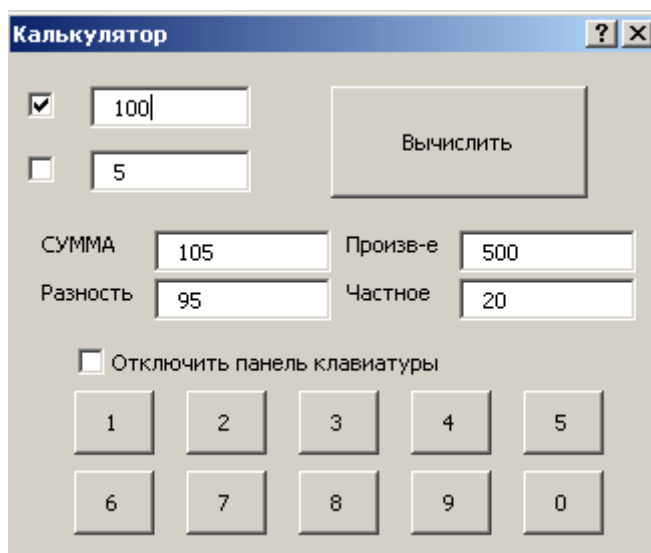


Рисунок 11.2 Форма 2

Возле каждого из двух полей ввода чисел необходимо создать 2 поля CheckBox, с условием, что в произвольный момент времени галочкой может быть отмечено либо одно поле, либо ни одного (рисунок 11.3).

Также необходимо реализовать функционал панели клавиатуры таким образом, чтобы при нажатии на соответствующую цифру – она появлялась бы в одном из двух полей, отмеченных галочкой. При нажатии на пункт «Отключить панель клавиатуры» вся кнопки панели должны стать недоступными для обработки.

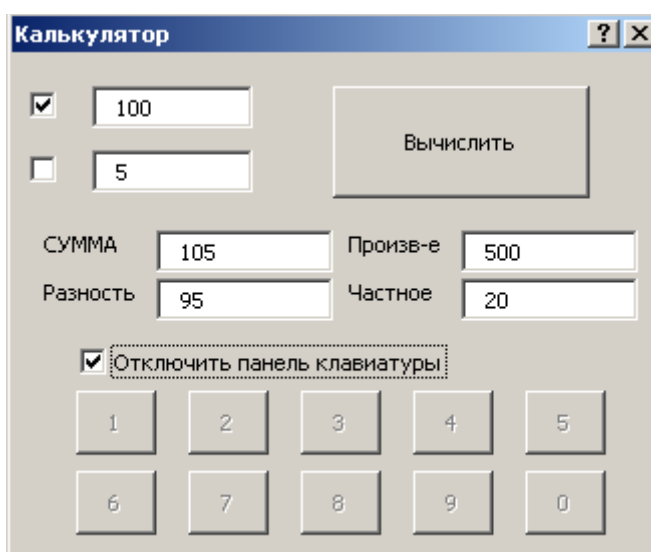


Рисунок 11.3 Форма 3

ПРАКТИЧЕСКАЯ РАБОТА №4. Технология программирования.

Работа с массивами.

Цель работы: получение практических навыков по созданию и управлению массивами.

Темы для предварительного изучения. типы данных, массивы, сортировка данных.

Теоретические сведения

Массив – это совокупность элементов, которые имеют одно и то же имя и тип. Элементы массива различаются по индексу. Упорядоченность данных в массиве позволяет обращаться к любому элементу массива по его индексу, а однотипность данных позволяет использовать циклическую обработку все элементов. Как правило, ввод, обработка и вывод массива осуществляется поэлементно, с использованием цикла с параметром. Простейший способ ввода – ввод массива с клавиатуры. Над массивом нельзя выполнять арифметические действия (вычитать, складывать и т.д.) Все действия выполняются поэлементно.

Постановка задачи

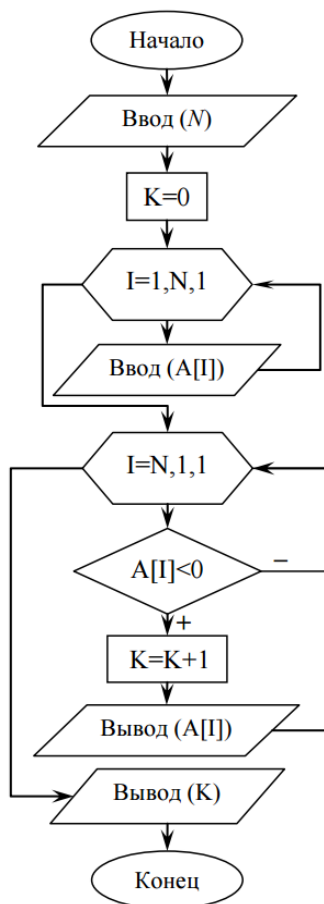
Задание 12.1 Написать программу для работы с массивом согласно представленному ниже алгоритму на языке Basic.

Порядок выполнения работы

Дан целочисленный массив размера N. Вывести все содержащиеся в данном массиве отрицательные числа в порядке убывания их индексов и их количество.

Необходимо ввести весь массив, для этого создадим цикл, в котором будет только ввод данных. Для вывода результата и подсчета отрицательных элементов организуем цикл с параметром и с шагом – 1. Выведем подсчитанное количество нужных элементов

1. Начало;
 2. Ввод (N);
 3. $K=0$;
 4. Для $I = 1$ до N выполнить
Ввод ($A[I]$);
 5. Для $I = N$ до 1 выполнить
Если $A [I]<0$ то нц
Вывод ($A[I]$);
 $K=K+1$;
кц
 6. Вывод (K);
- Конец.



Методические указания по самостоятельной работе

1. Слияние документов
2. Использование макросов в офисных приложениях.
3. Использование специальных функций в электронных таблицах
4. Планирование и управление проектами с использованием MS Project
5. Использование методологии RAD при программировании

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Тема 1. Слияние документов

Слияние документов - это объединение основного документа, содержащего постоянную часть информации, и источника данных, содержащих переменную часть. Примером слияния документов может быть персонализация писем. Текст делового письма постоянный, например, сообщение участникам математической олимпиады. Это основной документ. Такое письмо нужно выслать участникам олимпиады. Переменным является Фамилия И.О. участника, его адрес, набранные баллы. Данные об участниках представляют собой источник данных (список). Слияние проходит по следующей схеме.

Источник данных
(список)

Фамилия	Имя	Отчество	Индекс	Адрес	Сумма_баллов
Петров	Иван	Сергеевич	220015	г. Минск ул. Я. Мавра д.23 кв.12	25
Сергеев	Петр	Иванович	220088	г. Минск ул. Ленина, д.34 кв. 112	30

Основной документ

Поле слияния

<<Индекс>>
<<Адрес>>

Уважаемый << Фамилия >> << Имя >> << Отчество >>!
Сообщаем, что Вы, участвуя в математической олимпиаде,
набрали <<Сумма_баллов>> баллов.

Оргкомитет

Результат слияния

220015
г. Минск ул. Я. Мавра д.23 кв.12

Уважаемый Петров Иван Сергеевич!
Сообщаем, что Вы, участвуя в математической олимпиаде,
набрали 25 баллов.

Оргкомитет

220080
г. Минск ул. Ленина, д.34 кв. 112

Уважаемый Сергеев Петр Иванович!
Сообщаем, что Вы, участвуя в математической олимпиаде,
набрали 30 баллов.

Оргкомитет

В результате слияния основного документа и источника данных (списка) для каждого участника из списка готовится письмо. В итоге получается несколько писем одинакового содержания.

Слияние документов выполняется в диалоговом окне «Слияние», вызываемом командой **Сервис/Письма и рассылки/Слияние** в офисных приложениях.

Вопросы для рассмотрения

1. Источники данных
2. Переменные в документах
3. Самостоятельная работа с документами при слиянии

Тема 2. Использование макросов в офисных приложениях

Вопросы для рассмотрения

1. Особенности работы функции «Запись макроса».
2. Изменение записанных макросов.
3. Ручное создание новых макросов

Тема 3. Использование специальных функций в электронных таблицах

Она позволяет выполнять логические сравнения значений и ожидаемых результатов. Самая простая **функция ЕСЛИ** означает следующее: **ЕСЛИ** (это истинно, то сделать это, в противном случае сделать что-то еще).

Вопросы для рассмотрения

1. Синтаксис специальных функций.
2. Функция ЕСЛИ

Тема 4. Планирование и управление проектами с использованием MS Project. Вопросы для рассмотрения

1. Этапы планирования проекта в MS Project
2. Основные бизнес-процессы.
3. Планирование ресурсной базы проекта.

Тема 5. Использование методологии RAD при программировании

Вопросы для рассмотрения

1. Особенности методологии RAD
2. Отличия ЖЦ программ при использовании RAD
3. Примеры использования методологии.

ФОРМА КОНТРОЛЯ

1. Сообщения в форме докладов, индивидуальный опрос.
2. Индивидуальные отчеты, опрос по принципу коллоквиумов.

Список рекомендованной литературы

1. Миньков С.Л. Excel: лабораторный практикум. — Томск: ТУСУР, 2004. — 145 с.
2. Карпова Т.С. Базы данных: модели, разработка, реализация : учебное пособие / Т. С. Карпова. - СПб. : Питер, 2002. - 303[1] с. : ил.
3. Ковалев С.П. Формальный подход к разработке программных систем: Уч.пособие / НГУ. Новосибирск, 2004. 180с.
4. Боровской И.Г., Колесникова С.И., Матолыгин А.А. Специализированная подготовка разработчиков бизнес-приложений. Уч.пособие. Томск: ТУСУР, 2007. – 267с.
5. Ковалев С.П. Формальный подход к разработке программных систем: Уч.пособие / НГУ. Новосибирск, 2004. 180с.