Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кузбасский государственный технический университет
имени Т. Ф. Горбачева»

Кафедра прикладных информационных технологий

Составитель
Л. С. Таганов

ИНФОРМАТИКА

Методические указания к контрольной работе № 2 для студентов заочной формы обучения

Рекомендовано учебно-методической комиссией
специальности 21.05.04 «Горное дело» в качестве электронного издания
для самостоятельной работы

Кемерово 2016

Рецензенты:

Соколов И. А. – доцент кафедры прикладных информационных технологий, кандидат технических наук

Удовицкий В. И. – председатель учебно-методической комиссии специальности 21.05.04 «Горное дело», профессор, доктор технических наук

Таганов Леонид Степанович
Колокольникова Алла Ивановна

Информатика: методические указания к контрольной работе № 2 [Электронный ресурс]: для студентов специальности 21.05.04 «Горное дело» заочной формы обучения / сост.: Л. С. Таганов; КузГТУ. – Электрон, дан. – Кемерово, 2016. – Систем, требования: Pentium IV; ОЗУ 8 Мб; Windows XP; мышь. – Загл, с экрана.

Контрольная работа предназначена для оценки знаний, полученных в результате самостоятельного изучения материала курса, включает варианты заданий, требования к оформлению, список литературы.

Содержание

1. Общее содержание работы

Контрольная работа по дисциплине «Информатика» предполагает выполнение **5 заданий** в соответствии с выбранным вариантом. Процедура выбора варианта описывается в каждом задании. Все задания должны быть оформлены в виде одного файла.

Первое задание нацелено на проверку знаний логических основ работы ЭВМ (понятие систем счисления, умение выполнять операции по переводу из одной системы счисления в другую, производить арифметические операции в различных системах счисления).

Второе задание нацелено на формирование навыков построения блок – схем алгоритмов линейной, разветвляющейся и циклической структуры.

Три последующих задания предполагают написание текстов программ на алгоритмическом языке VBA по темам:

- Программирование линейных вычислительных процессов;
- Программирование разветвляющихся вычислительных процессов;
- Программирование циклических вычислительных процессов.
- Контрольная работа должна содержать:
- Титульный лист, на котором не забываем указывать номер варианта: «Вариант №.»,
- Содержание (автособираемое содержание), состоящее из 5 пунктов, соответствующих 5 заданиям контрольной работы,
- Ход выполнения пяти заданий контрольной работы.

Описание хода выполнения задания начинается с указания номера задания: **Задание №** и текста задания.

З**адание № 2** предполагает построение блок-схемы алгоритма решения задачи. В случае описания хода выполнения **заданий № 3, 4, 5** необходимо представить блок-схему алгоритма, текст программы (листинг или скриншот), исходные данные и полученные результаты. Блок-схема алгоритма решения задачи строится для заданий 2, 3, 4, 5. Текст программы, исходные данные и результат расчёта приводятся для заданий 3, 4, 5. Обязательно наличие листинга или скриншота текста программы, исходных данных и полученных результатов.

Если в задании не указаны исходные данные, студент вправе выбрать данные по своему усмотрению. Блок-схема алгоритма решения задачи должна быть оформлена с соблюдением ГОСТ 19_701-90, в среде текстового процессора MS WORD.

2. Правила оформления контрольной работы

2.1. Общие требования

a. Работа должна быть выполнена в электронном виде. Отчет по работе отправить по электронной системе обучения КузГТУ или в крайнем случае на электронную почту ведущего преподавателя.
b. При наборе и форматировании текста в среде текстового процессора следует соблюдать следующие требования:
   - Шрифт Times New Roman, размер 14pt;
   - Выравнивание – по ширине;
   - Межстрочный интервал 1;
   - Автоматический перенос слов;
   - Размеры полей: левое – 25 мм, правое – 10 мм, верхнее – 20 мм, нижнее – 20 мм;
   - Отступ первой сроки должен быть одинаковым по всему тексту и равен 1,25 см.
c. Разрешается использовать компьютерные возможности акцентирования внимания на определенных терминах, формулах, определениях применяя разные шрифты.

2.2. Нумерация страниц

a. Страницы работы следует нумеровать арабскими цифрами, соблюдая сквозную нумерацию по всему тексту, включая приложения. Номер страницы проставляют в центре нижней части листа.
b. Титульный лист включают в общую нумерацию страниц работы. Номер страницы на титульном листе не проставляют.
c. Иллюстрации и таблицы, расположенные на отдельных листах, включают в общую нумерацию страниц документа.
d. Иллюстрации и таблицы на листе формата А3 учитывают, как одну страницу.

2.3. Титульный лист

Титульный лист является первой страницей работы и предшествует основному тексту (образец в приложении 2).

2.4. Содержание

a. Содержание включает введение, наименование всех разделов, подразделов, пунктов (если они имеют наименование), заключение, библиографический список, приложения с указанием номеров страниц, с которых начинаются эти элементы документа. Наименование приложений в разделе «Содержание» не указывают, а помещают после библиографического списка на отдельном листе.

b. Пункты: «Введение», «Заключение», «Библиографический список», «Приложения», по желанию автора, могут быть пропущены в контрольной работе.
c. Обязательным является наличие основной части, содержащей 3-4 раздела.
d. Слово «Содержание» записывают в виде заголовка, симметрично тексту (по центру), прописными буквами.
e. Наименования, включенные в содержание, записывают строчными буквами, начиная с прописной буквы.
f. Заголовки «Введение», «Заключение», «Библиографический список», «Приложения» не нумеруются и вводятся на уровне номера буквы (цифры) наименования раздела.
g. Между наименованием раздела (главы) и номером страницы можно использовать заполнитель, например, в виде точек.
h. Формирование содержания следует осуществлять средствами текстового процессора (автособираемое оглавление – Вкладка **Ссылки**, группа **Оглавление**, кнопка **Оглавление**).

3. Задание № 1

Ответить на контрольные вопросы

1. Понятие алгоритма.
2. Свойства алгоритма и их сущность.
3. Способы записи алгоритма.
4. Основные элементы блок-схемы.
5. Виды алгоритмов.
6. Отличительные особенности алгоритмов с предусловием и постусловием.
7. Правила составления сложных (вложенных) циклов.
8. Инкапсуляция данных вместе с кодом, предназначенным для их обработки – это?
9. Объект, содержащий несколько других объектов того же типа – это?
10. Проект, на основе которого будет создан объект – это?
11. Действия, выполняемые над объектом – это?
12. Атрибут объекта, определяющий его характеристики – это?
13. Действия, распознаваемые объектом – это?
14. Основные парадигмы объектно-ориентированного программирования на языке VBA.
15. Основные конструкции языка VBA.

Вариант задания №2, 3, 4, 5 выбирается по первой букве фамилии студента согласно ниже представленной таблице. В ячейках таблицы размещены два символа. Первый символ – буква (указывает на фамилию студента), второй – 44 цифра (указывает на номер варианта).

А 1	Б 2	В 3	Г 4	Д 5	Е 6	Ё 7	Ж 8	З 9	И 10

Й 1	К 2	Л 3	М 4	Н 5	О 6	П 7	Р 8	С 9	Т 10
У 1	Ф 2	Х 3	Ц 4	Ч 5	Ш 6	Щ 7	Ы 8	Э 9	Ю 10
Я 1									

4. Задание № 2

Построение блок-схемы алгоритма расчёта

Примечание: Для построения блок-схемы можно использовать: в MS Word меню: “Вставка – Фигуры”; MS Excel меню: “Вставка – Иллюстрации – Фигуры”.

Вариант № 1. Построить блок-схему алгоритма нахождения квадрата минимального значения среди трёх чисел x, y, z.

Вариант № 2. Построить блок-схему алгоритма нахождения наибольшего из четырёх неравных чисел a, b, c, d.

Вариант № 3. Построить блок-схему алгоритма вычисления величины y

$y = x^2 + 1$ *для* $x = 1,2,.....,50$

Вариант № 4. Построить блок-схему алгоритма вычисления величины V, при заданных V_0, a, t $V = V_0 + at^2/2$

Вариант № 5. Построить блок-схему алгоритма вычисления величины s, при заданных b_i

$$s = \sum_{i=1}^{n} b_i$$

Вариант № 6. Построить блок-схему алгоритма вычисления величины z, при заданных a, h, k.

$$z = \frac{1}{a} + \frac{1}{a+h} + \frac{1}{a+2*h} + \cdots\cdots + \frac{1}{a+k*h}$$

Вариант № 7. Построить блок-схему алгоритма расчёта величины y, при заданном x. Вычисление продолжать до тех пор, пока очередное слагаемое не станет меньше заданного числа e по абсолютной величине.

$$y = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots\cdots$$

Вариант № 8. Дан одномерный массив D, содержащий 15 элементов. Построить блок-схему алгоритма вычисления суммы элементов массива, начиная со второго, выбирая их через два.

Вариант № 9. Дана последовательность из 20 целых чисел. Построить блок-схему алгоритма вычисления суммы и количества всех положительных и всех отрицательных чисел.

Вариант № 10. Даны две последовательности чисел A_i и B_i. Построить блок-схему алгоритма перемножения элементов этих последовательностей по правилу: $C_i = A_i * B_i$.

5. Задание № 3

Составить программу вычисления значения функции согласно предложенному варианту

Вариант № 1. $y = arctg\,|(a-b)/2| + \sqrt{|a-b|}$ при а=0,266; в=0,0805

Вариант № 2. $y = \sin^2\alpha \cdot (a^2 + \ln x) + (\sin\alpha)$ при а=0,266; х=0,0831; $\alpha = 42^0$

Вариант № 3. $m = |(k-n)/k \cdot n| \cdot \sqrt{\sin(2\alpha/3)}$ при n=0.508; k=0.756; a = 45^0

Вариант № 4. $y = \sqrt{d + \sin x/|a^2 - b^2|}$ при d=1.3139; х=23; а=1.325; b=0.0823

Вариант № 5. $y = (2\sin(\pi/6 \cdot n) + n^2)/a + \sqrt{a+n}$ при а=0.08; n=0.50874

Вариант № 6. $y = \sin\alpha/(a-b) + \sqrt{(a+b)/b}$ при $\alpha = 32^0 15'$; а=0.875; b=0.5275

Вариант № 7. $z = (tg\alpha + (a+b))/b + |a+b|^3$ при $\alpha = 45^0$; а=1.753; b=2.856

Вариант № 8. $f = \sqrt{\ln|(a-b)| \cdot (a-b)^2}$ при а=0.243; b=2.713

Вариант № 9. $z = (\sin\alpha \cdot e)/a \cdot \sqrt{|e^2 - a|}$ при а=0.266; $\alpha = 42^0 30'$

Вариант № 10. $y = d + (|a^2 - b^2| / \sin\pi/c)^2$ при а= -1.325; b=0.00823; c=21.03; d=1.3139

6. Задание № 4

Составить программу расчета значения функции согласно предложенному варианту

Вариант № 1. $y=\begin{cases} x^2-a, & при\ x\rangle a \\ a^2-x, & при\ x\langle a \\ x/a, & при\ x=a \end{cases}$

Вариант № 2. $f=\begin{cases} y^3-0.3, & при\ y\langle 0 \\ y^3+1, & при\ y\rangle 1 \\ 0, & при\ 0\le y\le 1 \end{cases}$

Вариант № 3. $Z=\begin{cases} a^3\cdot t+b\cdot t, & при\ t\langle a \\ a^4\cdot t-b\cdot t, & при\ t\rangle a \\ a^2\cdot t+b, & при\ t=a \end{cases}$

Вариант № 4. $Z=\begin{cases} 5^x-2\cdot x, & при\ -1\le x\le 0 \\ 5^x+2^x, & при\ 0\le x\le 2 \\ 0, & в\ \ др.случ. \end{cases}$

Вариант № 5. $N=\begin{cases} 17000-0.486\cdot R, & при\ R-120\langle 0 \\ 0, & при\ R-120=0 \\ 180, & при\ R-120\rangle 0 \end{cases}$

Вариант № 6. $Y=\begin{cases} 1, & при\ x\rangle 12 \\ 2+x, & при\ 0\le x\le 12 \\ -2, & в\ \ др.случ. \end{cases}$

Вариант № 7. $f=\begin{cases} x^2, & при\ y\rangle 0\ и\ \ t\rangle 0 \\ x^3, & при\ y\rangle 0\ и\ \ t\langle 0 \\ 0, & в\ \ др.\ случ. \end{cases}$

Вариант № 8. $y=\begin{cases} x^2-a, & при\ x\rangle a \\ a^2-x, & при\ x\langle a \\ x/a, & в\ \ др.случ. \end{cases}$

Вариант № 9. $f=\begin{cases}0, & при\ x\langle -1\\ a+b & при\ -1\le x\le 1\\ 1, & при\ x\rangle 1\end{cases}$

Вариант № 10. $Z=\begin{cases}0, & при\ x\langle -1\\ (a+b)/5, & при\ -1\le x\le 1\\ 1+a/b, & при\ x\rangle 1\end{cases}$

7. 8Задание № 5

Составить программу решения задачи согласно предложенному варианту

Вариант № 1. Напишите программу расчета стоимости порций сыра весом 50, 100, 150, …,1000 г. Цена 1 кг – 250 руб.

Вариант № 2. Напишите программу перевода единицы длины из дюймов в сантиметры. (1дюйм=2.54 см) для значений от 1 до 10 дюймов с шагом 1.

Вариант № 3. Напишите программу вычисления суммы квадратов первых 7 натуральных чисел.

Вариант № 4. В 2010 г. урожай ячменя составил 20 ц с га. В среднем каждые 2 года за счет применения новой технологии урожай увеличивается на 5 %. Определить через сколько лет урожайность достигнет 25 ц с га.

Вариант № 5. Имеется массив чисел x_1, x_2, …, x_n. Требуется составить программу вычисления среднего арифметического этих чисел по правилу: $S = (x_1+x_2+\ldots+x_n)/n$

Вариант № 6. Составьте программу вычисления величины z при заданных a, h, k.

$$z=\frac{1}{a}+\frac{1}{a+h}+\frac{1}{a+2*h}+\cdots\cdots+\frac{1}{a+k*h}$$

Вариант № 7. Напишите программу перевода температуры из градусов по шкале Цельсия (°C) в градусы шкалы Фаренгейта (F) для значений от 15°C до 30°C с шагом 1°C. Перевод осуществляется по формуле $F = 1{,}8\cdot°C+32$

Вариант № 8. Напишите программу соответствия между весом в фунтах и в кг для значений от 1 до 10 фунтов с шагом 1 фунт (1 русский фунт равен 409,5 г).

Вариант № 9. Напишите программу вычисления величины y по следующей формуле $y = x^2 + 1$ *для* $x = 1,2.....50$

Вариант № 10. Начав тренировки, спортсмен в первый день пробежал 10 км. Каждый следующий день он увеличивал дневную норму пробега на 10 % от нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за 7 дней?

Литература

1. Таганов, Л. С. Информатика [Электронный ресурс]: учеб, пособие для студентов техн. Специальностей и направлений / Л. С. Таганов, А. Г. Пимонов; ГОУ ВПО «Кузбас. гос. техн. ун-т имени Т. Ф. Горбачева». – Кемерово, 2010. – 330 с. – Режим доступа: http://library.kuzstu.ru/meto.php?n=90457&.type=utchposob:common
2. Колокольникова, А. И. Основы информатики [Электронный ресурс]: учеб, пособие для самостоятельной подготовки студентов / А. И. Колокольникова, Л. С. Таганов; ГОУ ВПО «Кузбас. гос. техн. ун-т имени Т. Ф. Горбачева». – Кемерово, 2015. – 308 с. – Режим доступа: http://library.kuzstu.ru/meto.php?n=91267&type=utchposob:common
3. Таганов, Л. С. Информатика: Мультимедийные материалы к курсу лекций [Электронный ресурс]: учебное пособие для студентов специальности 130400.65 «Горное дело» / Л. С. Таганов, А. И. Колокольникова; ГОУ ВПО «Кузбас. гос. техн. ун-т имени Т. Ф. Горбачева». – Кемерово, 2012. – Режим доступа: http://library.kuzstu.ru/meto.php?n=91110&type=utchposob:common

Приложение 1

1. Основы алгоритмизации

Алгоритм – понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение цели.

Любой алгоритм должен обладать следующими **свойствами:**

• **определенностью –** за конечное число шагов либо должен быть получен результат, либо доказано его отсутствие;

• **результативностью** – обязательным получением некоторого результата (числа, таблицы, текста, звука, изображения и т. д.) или сигнала о том, что данный алгоритм неприменим для решения поставленной задачи;

• **массовостью** – возможностью получения результата при различных исходных данных для некоторого класса сходных задач;

• **формальностью** – отвлечение от содержания поставленной задачи и строгое выполнение некоторого правила, инструкции;

• **дискретностью** — возможностью разбиения алгоритма на отдельные элементарные действия.

Существуют следующие **формы представления алгоритма**:

• Словесная (вербальная) на неформальном языке (псевдокод).

• На языках программирования (программа).

• Графическая (блок-схема).

Словесная форма представления алгоритма имеет ряд недостатков. Для достаточно сложных алгоритмов описание становится слишком громоздким и не наглядным. Эта форма представления обычно используется лишь на начальных стадиях разработки алгоритма.

Алгоритм, записанный на **языке программирования**, называется *программой***.**

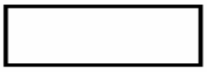
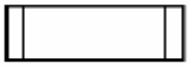
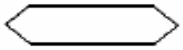
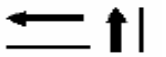
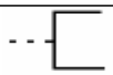
Графическая форма представления алгоритмов является более наглядной и строгой. Алгоритм изображается в виде последовательности связанных между собой блоков, каждый из которых соответствует выполнению одного или нескольких операторов. Такое графическое представление называется **блок-схемой алгоритма.**

Условные графические обозначения символов, используемых для составления блок-схемы алгоритма, стандартизированы.

Некоторые, часто используемые обозначения, приведены в табл. 1.

Таблица 1

Условные графические обозначения символов

Название блока	**Обозначение**	**Название блока**	**Обозна-чение**
Начало или конец алгоритма		Решение	
Процесс (действие или серия действий)		Предопределенный процесс (вспомогательный алгоритм)	
Ввод/вывод данных		Модификация (заголовок цикла)	
Линии потока		Комментарии	

Отдельные блоки алгоритмов соединяются между собой линиями потоков, которые проводятся параллельно внешней рамке чертежа. Направления линий потока сверху вниз и слева направо принимаются за основные и, если линии потоков не имеют изломов, стрелками не обозначаются. Обратные направления линий потока помечаются стрелкой.

«Процесс» (этап вычисления) изображается прямоугольником, внутри которого записывается набор действий. Ромбом изображается «решение», внутри которого осуществляется проверка условия. Ввод исходных данных и вывод результатов изображаются параллелограммами, внутри которых пишутся слова «ввод» или «вывод» и перечисляются переменные, подлежащие вводу или выводу.

Представление алгоритма в виде блок-схемы является промежуточным, так как алгоритм в таком виде не может быть непосредственно выполнен ЭВМ, но помогает пользователю при создании (написании) программы для ПК.

Использование блок-схем дает возможность:

- наглядно отобразить базовые конструкции алгоритма;
- сосредоточить внимание на структуре алгоритма, а не на синтаксисе языка;
- анализировать логическую структуру алгоритма;
- преобразовывать алгоритм методом укрупнения (сведения к единому блоку) или детализации – разбиения на ряд блоков;
- использовать принцип блочности при коллективном решении сложной задачи;
- осуществить быструю проверку разработанного алгоритма (на уровне идеи);
- разобрать большее число учебных задач.

Составление блок-схемы алгоритма является важным и в большинстве случаев необходимым этапом решения сложной и большой задачи на ЭВМ, значительно облегчающим процесс составления программ.

Базовые структуры (модели) программирования

Выделяют три основные структуры алгоритмов:

1. Линейная.
2. Разветвляющаяся (альтернатива «если–то–иначе» или «если–то»).
3. Циклическая (повторение).

Линейная структура – является основной. Она означает, что действия выполняются друг за другом. Прямоугольник, показанный на рисунке, может представлять, как одну единственную команду, так и множество операторов, необходимых для выполнения сложной обработки данных. Команды записываются с помощью операции присваивания.

Присваивание переменной какого-либо значения или присваивание одной переменной значения другой переменной является наиболее часто выполняемым действием в программе, написанной на любом языке программирования. В языке программирования присваивание является операцией и обозначается как «:=» или «=». Это означает, что при выполнении этой операции происходит не только присваивание значения определенной переменной, но и возвращается некоторое значение.

Разветвляющаяся структура (ветвление) – это структура, обеспечивающая альтернативный выбор в зависимости от заданного условия. Выполняется проверка условия, а затем выбирается один из путей, где ***P*** – это условие, в зависимости от истинности (Да) или ложности (Нет) которого управление передается по одной из двух ветвей.

Может оказаться, что для одного из результатов проверки условия ничего предпринимать не надо. В этом случае можно применять только один обрабатывающий блок (рис. 3).

Эта структура называется также ЕСЛИ – ТО – ИНАЧЕ. Каждый из путей (ТО или ИНАЧЕ) ведет к общей точке слияния, так что выполнение программы продолжается независимо от того, какой путь был выбран.

Циклическая структура (или повторение) предусматривает повторное выполнение некоторого набора действий. Последовательность действий, которые повторяются в цикле, называют **телом цикла.**

Циклические алгоритмы подразделяют на алгоритмы с предусловием, постусловием и алгоритмы с конечным числом повторов. В алгоритмах с предусловием сначала выполняется проверка условия окончания цикла и затем, в зависимости от результата проверки, выполняется (или не выполняется) так называемое тело цикла.

Пример № 1. Определить площадь трапеции по введенным значениям оснований (a и b) и высоты (h).

Запись решения задачи на алгоритмическом языке:

Алгоритм трапеция
Вещественные a, b, h, s
Начало
Ввод a, b, h
*s= ((a + b)/2)*h*
Вывод s

Конец

Запись алгоритма в виде блок-схемы (рис. 1):

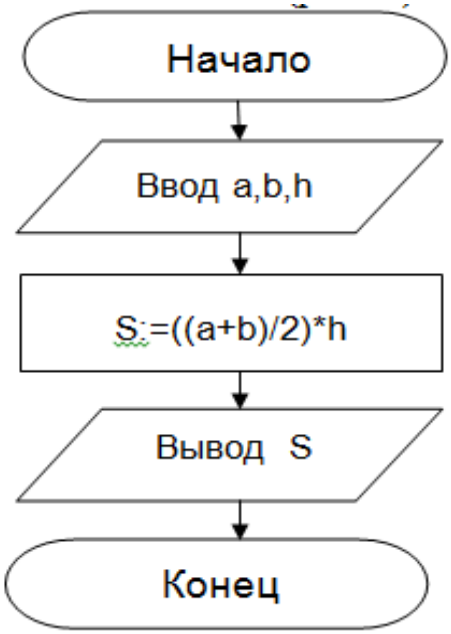


Рис. 1. Блок-схема линейного алгоритма

Пример № 2. Определить среднее арифметическое двух чисел, если a положительное, иначе частное (a/b).

Запись решения задачи на алгоритмическом языке:

Алгоритм числа
Вещественные a, b, c
Начало
Ввод a, b
Если a> 0
тогда c = (a + b) /2
иначе c=a/b
Конец если
Вывод c
Конец

Запись алгоритма в виде блок-схемы (рис. 2):

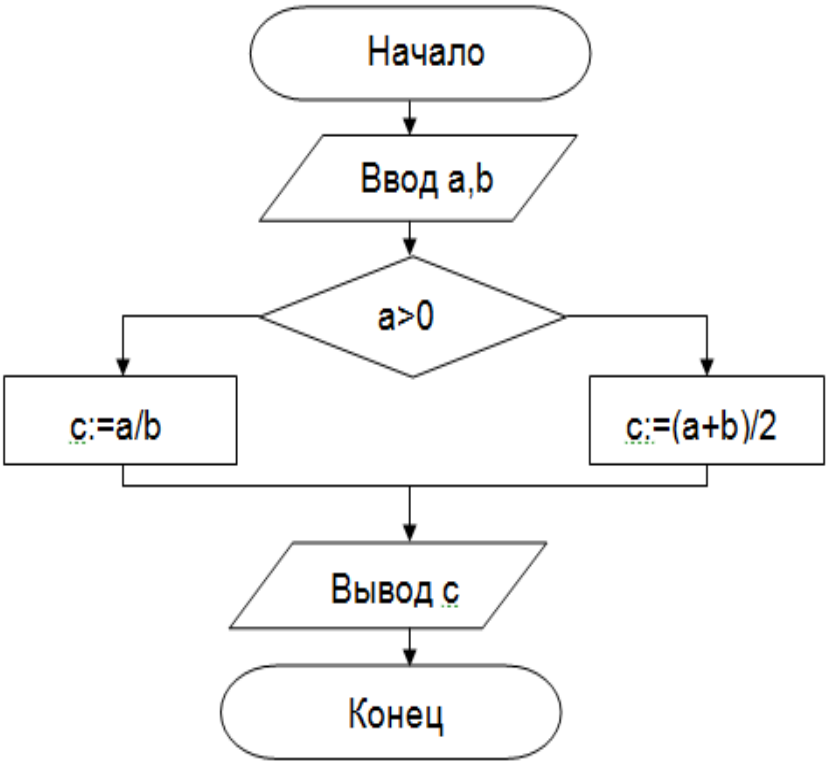


Рис. 2. Блок-схема алгоритма с ветвлением

В алгоритме с предусловием сначала проверяется условие окончания цикла а затем выполняется тело цикла. Решение задачи нахождения суммы первых десяти целых чисел в данном случае будет выглядеть следующим образом:

Пример № 3. Составить алгоритм нахождения суммы целых чисел в диапазоне от 1 до 10.

Запись решения задачи на алгоритмическом языке:

```
Алгоритм сумма
   Целые a, s
Начало
   S=0
   a=1
   Начало цикла
      Пока a <=10
           S=S + a
           a=a+1
   Конец цикла
   Вывод S
Конец
```

Запись алгоритма в виде блок-схемы (рис. 3):

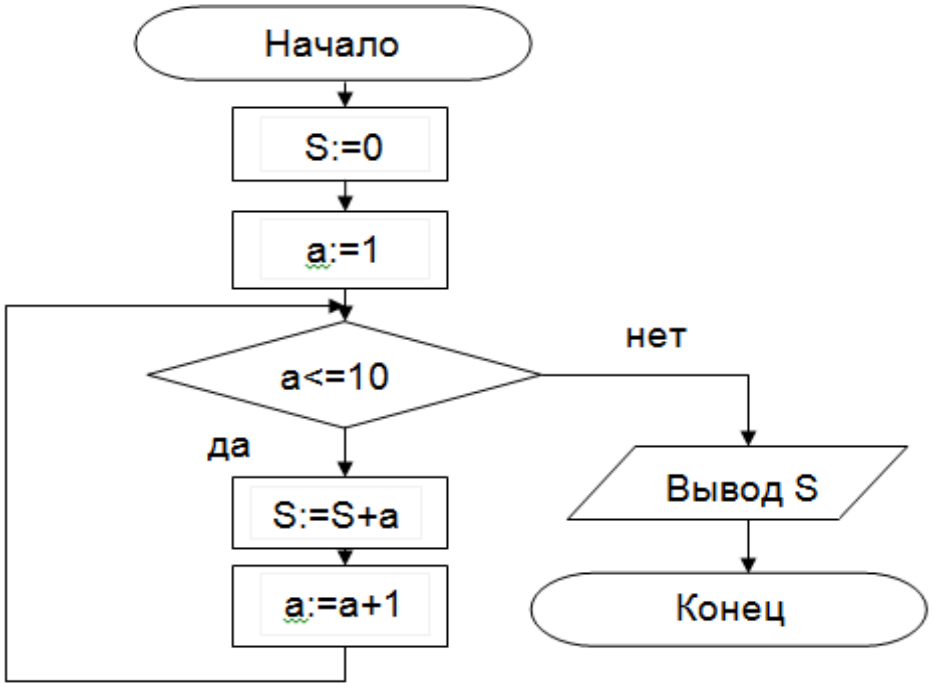


Рис. 3. Циклический алгоритм с предусловием

В алгоритме с постусловием сначала выполняется тело цикла, а затем проверяется условие окончания цикла. Решение задачи нахождения суммы первых десяти целых чисел в данном случае будет выглядеть следующим образом:

Алгоритм сумма
Целые a, s
Начало
S=0
A=1
Начало цикла
  S=S + a
  A=a+1
  Пока a <=10
Конец цикла
  Вывод S
Конец

Запись алгоритма в виде блок-схемы (рис. 4):

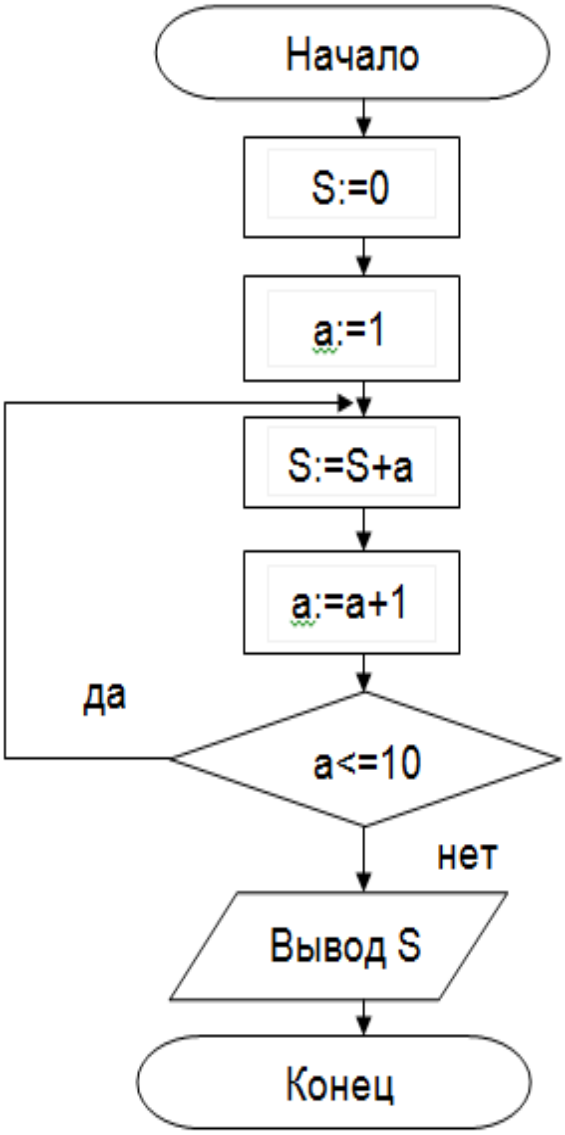


Рис. 4. Циклический алгоритм с постусловием

2. Основные понятия языка программирования VBA

Электронная таблица Excel содержит встроенную систему программирования на VBA. Для того, чтобы активизировать VBA можно воспользоваться одним из следующих путей. Во-первых, самый простой путь – находясь в редакторе Microsoft Excel нажать комбинацию клавиш «Alt» – «F11». Во-вторых, щелкнув правой кнопкой мыши на ярлыке любого рабочего листа из всплывающего меню можно выбрать пункт «Исходный текст». В любом случае открывается окно, которое имеет вид, показанный на рис.1.

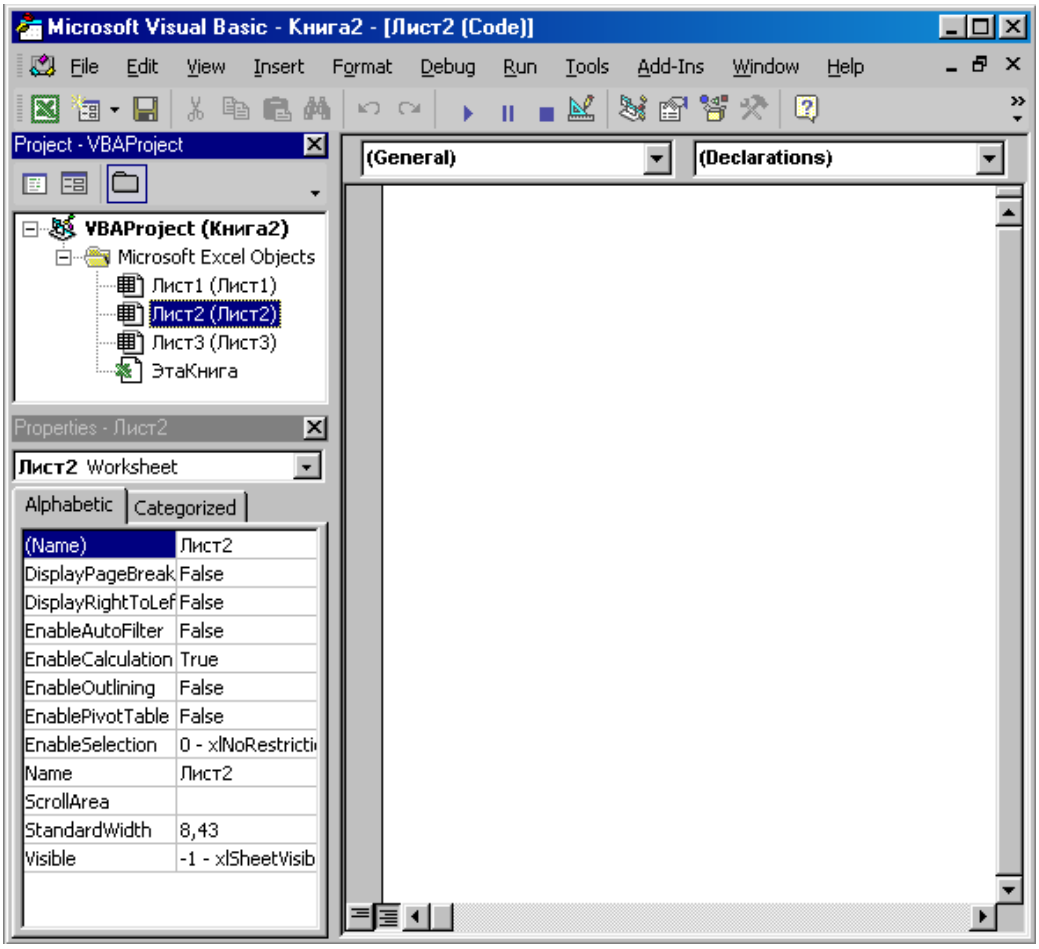


Рис. 1. Окно проекта

Это окно, в свою очередь, состоит из нескольких окон.

Окно проекта, расположенное в левом верхнем углу экрана, представляет иерархическую структуру рабочих листов, модулей и форм данной программы (проекта).

Окно свойств, расположенное ниже окна проекта, содержит перечисление свойств текущего объекта. Окно свойств состоит из двух частей – списка объектов и рабочей части – собственно свойств выбранного объекта.

В рабочей части перечислены имена и значения всех свойств объекта. Две закладки в верхней части позволяют переупорядочивать имена свойств по алфавиту или по категориям. Пользователь, создавая объект, может установить для него необходимые значения свойств.

Окно редактора исходного текста программы занимает большую часть в правой части экрана. Это окно тесно связано с окном проекта. При выборе какого – либо объекта в окне проекта, в окне редактора исходного текста автоматически будет появляться программный код, связанный с этим объектом (так называемый ***модуль объекта***). Написание программ в окне редактора существенно упрощается за счет его интеллектуальных возможностей. Так, например, редактор сам контролирует синтаксическую правильность вводимых операторов и выдает сообщения об ошибках, дописывает за программиста те участки программного кода, которые однозначно должны фигурировать в данном контексте и так далее. Кроме того, редактор автоматически распознает и выделяет разными цветами синтаксические конструкции VBA. Редактор кода

обладает еще одной очень полезной особенностью. Если текстовый курсор расположить на каком – либо служебном слове VBA, имени процедуры, свойства или метода и нажать клавишу «F1», то на экране появится окно со справочной информацией об этом объекте, слове или методе. Обычно в справке приводится и пример правильного использования данного программного кода. Один только недостаток – справка по VBA не переведена на русский язык. Но при даже незначительных знаниях английского – всегда можно оперативно получить помощь.

3. Основные синтаксические конструкции VBA

К основним конструкціям язика VBA относятся переменные, константы и служебные слова. Из этих основных, базовых для любого алгоритмического языка конструкций, строятся в свою очередь более сложные конструкции – операторы, процедуры, функции и, в конечном итоге, программы.

Переменные. Переменной называется область оперативной памяти, предназначенная для хранения какого-либо значения. Содержимое этой области памяти (значение переменной) может меняться во время выполнения программы. Каждая переменная имеет собственное имя, которое назначается программистом и формируется по следующим правилам:

- первым символом имени обязательно должна быть буква;
- имя может содержать только буквы, цифры и знак подчеркивания («_»);
- длина имени не должна превышать 255 символов;
- в одной и той же подпрограмме (процедуре или функции) не могут быть объявлены две переменные с одним и тем же именем.

Каждая переменная при создании должна получить определенный тип. Тип переменной характеризует длину, способ представления и диапазон изменения тех значений, которые могут храниться в переменной. В языке VBA возможны типы переменных, приведенные в таблице 1.

Таблица 1. Типы переменных языка VBA

Тип переменной	Способ	Размер	Диапазон значений в языке

	описания	**(Байт)**	**VBA**
Байт	Byte	1	Целые числа от 0 до 255
Логический тип	Boolean	2	Только значения True (истина) или False (ложь)
Целое число	Integer	2	От -32768 до +32767
Длинное целое	Long	4	От -2 147 483 648 до +2 147 483 647
Число с плавающей запятой одинарной точности	Single	4	От $-1.401298 \cdot 10^{-45}$ до $+3.042823 \cdot 10^{38}$
Число с плавающей запятой двойной точности	Double	8	От $-4.94065645841247 \cdot 10^{-324}$ до $+3.042823 \cdot 10^{38}$
Дата и время	Date	8	От 1 января 100 г. до 1 января 9999 г.
Любой объект	Object	4	Хранит указатель на любой объект, в том числе и на объекты Microsoft Excel
Строка символов	String	Длина строки	От 0 до 65000 символов в строке
Произвольный тип данных	Variant		Может хранить любое значение из описанных выше

Переменные могут быть объединены в так называемые массивы.

Массив – это группа однотипных переменных, которые названы одним общим именем. Массивы можно считать одной из разновидностей обычных (простых) переменных. Отличие состоит в том, что в массивах можно хранить не одно, а несколько значений. Доступ к заданному элементу массива осуществляется с помощью номера этого значения в массиве – так называемого индекса. Синтаксис объявления массива отличается от синтаксиса объявления переменной тем, что здесь требуется указать также размерность массива и границы изменения индексов.

Константы. Для повышения наглядности при написании программ вместо использования какого-либо постоянного значения часто используют *константы*. Применение констант облегчает восприятие текста программы, а так же значительно упрощает отладку программы.

В VBA существуют константы двух типов:

- В*строенные константы.* Список этих констант можно просмотреть в окне просмотра объектов. Их количество определяется используемым приложеним. Например, в Microsoft Excel к таким константам относятся *True, False, Null* и еще множество других, описывающих свойства объектов. Например, xlMaximized,

xlMinimized и другие выражения, упомянутые в п. 1.2.1, являются именно константами Excel. Об этом, к стати, говорит и префикс «xl» перед каждой такой константой.

- *Пользовательские константы* объявляются с помощью служебного слова *Const*. В момент объявления пользовательским константам присваиваются удобные для восприятия имена и значения. Например, часто используемое в программе число π можно описать в виде константы Const Pi = 3.141592, а затем идентификатор Pi многократно использовать в программе. Наглядность текста такой программы существенно повышается.

Операторы являются основными конструкциями языка. Именно с помощью операторов реализуется алгоритм работы программы. В современных языках существует огромное количество самых разнообразных операторов. Ниже кратко описаны основные операторы языка VBA.

Арифметические операторы предназначены для выполнения основных арифметических операций над операндами, в качестве которых могут выступать как переменные, так и числа, и константы. Знаками этих операторов являются общепринятые математические знаки „**+**”, „**-**”, „*****”, „**/**”. Например, оператор “**+**” выполняет операцию сложения двух чисел или выражений, являющихся операндами. Большинство арифметических операторов VBA требуют наличия двух операндов. Оператор присваивания, задаваемый необязательным служебным словом Let, используется в том случае, когда требуется присвоить значение выражения переменной или свойству.
Ее синтаксис имеет вид: [Let] <Имя Переменной> = <выражение>
Элементы синтаксиса инструкции Let содержат элементы, представленные в таблице 2.

Таблица 2. Элементы синтаксиса инструкции Let

Элемент	Описание
Let	Необязательный элемент. Ключевое слово Let обычно опускают
<Имя Переменной>	Обязательный элемент, представляет собой имя переменной или свойство, удовлетворяющее правилам именования переменных
<Выражение>	Обязательный элемент, определяющий значение, присваиваемое переменной или свойству

Значение выражения может быть присвоено только в том случае, если типы переменной и выражения совместимы. Например, нельзя присвоить числовой переменной значение выражения, которое является строкой.

В Visual Basic for Applications имеются стандартные окна для ввода данных в программу и отображения результатов, т.е. вывода данных. Окно

сообщений (*MsgBox*) и окно ввода (*InputBox*) позволяют организовать простейший диалог с пользователем.

Синтаксис окна ввода: *Переменная* = *InputBox*(“текст”, “заголовок окна”)

Этот оператор выводит на экран диалоговое окно, которое содержит текст – сообщение и поле для ввода данных.

Синтаксис окна сообщений MsgBox:

MsgBox(сообщение, кнопки+ значки, заголовок окна)

Параметр *сообщение* содержит текст, который выводится в окне диалога.

Параметр *кнопки* указывает число и тип отображаемых кнопок в окне сообщения**.**

Меню редактора VBA

После входа в редактор Visual Basic, а также при изменении макроса в меню вы автоматически окажетесь в редакторе. Состав и назначение основных пунктов меню редактора:

- **File** – команды сохранения изменений в проекте и вывода на экран и печать исходного кода макросов.
- **Edit** – команды управления исходным кодом в окне Code, а также объектами в формах.
- **View** – команды, позволяющие выводить или убирать с экрана различные окна самого редактора VBA.
- **Insert** – команды вставки в проект различных объектов: процедур, модулей, форм, классов и пр.
- **Format** – команды, используемые при создании пользовательских диалоговых окон. Они позволяют выравнивать объекты в форме по отношению друг к другу, настраивать размеры и внешний вид элементов управления и другие операции.
- **Debug** – команды тестирования и отладки кода. Позволяют запускать код с любой точки, отслеживать ход выполнения по шагам, видеть значения, прерывать программу в нужном месте.
- **Run** – команды запуска программного кода на выполнение, прерывания, возобновление работы, а также возврата прерванной программы в исходное состояние.
- **Tools** – команды, позволяющие выбрать макрос для выполнения или получения доступа к внешним библиотекам макросов. Доступ к диалоговому окну Option (параметры) редактора и окну свойств VBA.
- **Add-Ins** – одна команда Add-Ins Manager для вывода диалогового окна, в котором можно загружать, выгружать, регистрировать или определять поведение программ – дополнений (надстроек).

Основными парадигмами объектно-ориентированного программирования на языке VBA являются объект, свойство, метод, событие, класс и семейство объектов.

Объект – это инкапсуляция данных вместе с кодом, предназначенным для их обработки.

Семейство – объект, содержащий несколько других объектов того же типа:
Worksheets (“Лист 1”) – рабочий лист с именем Лист1,
Worksheets (1) – первый лист рабочей книги.
Класс – это проект, на основе которого будет создан объект, т.е. класс определяет имя объекта, его свойства и действия, над ним выполняемые. А каждый объект, свою очередь, является экземпляром класса.
Методы – это действия, выполняемые над объектом.
Объект. Метод – синтаксис метода
Пример.
Application. Quit – закрыть объект Application.
Worksheets (“Лист1”). Chartobjects.Delete – удалит все диаграммы с листа “Лист1”.
Свойства – это атрибут объекта, определяющий его характеристики: размер, цвет, положение на экране или состояние (доступность, видимость).
Для изменения характеристик меняют его свойства:
Объект. Свойство=Значение свойства
Пример.
Worksheets.Visible = False
Есть свойства, возвращающие объект:
ActiveCell возвращает активную ячейку активного листа активной рабочей книги.
ActiveWindow – активное окно.
Свойства: ActiveCell, ActiveWindow. ActiveCell и Application.
ActiveCell возвращают одну и ту же активную ячейку.
События – это действия, распознаваемые объектом.
Суть программирования на VBA и заключается в том, чтобы на событие получить отклик.

Рекомендации по минимизации количества ошибок при составлении кода программы на языке программирования VBA

Любой программист должен стремиться составить надежную программу. Надежная программа – это программа, которую легко понять и она может работать без особых проблем.

Составление надежной программы – результат больших усилий и опыта, что потребует много времени и упражнений. Для облегчения такой работы направлены некоторые рекомендации:

- Обязательно объявлять переменные явно, для этого в начале каждого программного модуля включать инструкцию *Option Explicit* (все переменные объявлять явно). Это простое действие, возможно, является самым важным в написании надежной программы, потому что предотвращает все досадные ошибки и дефекты, которые происходят из-за неправильно написанных названий переменных и их, как правило, так тяжело выявить;

- Не использовать без крайней необходимости тип *Variant*, так как он иногда является причиной некоторых трудноуловимых ошибок в программе;
- Не использовать без крайней необходимости тип *Object*, **так** как он может содержать ссылку на любой тип объектов, а эта гибкость может быть источником проблем. При всякой возможности объектные переменные объявлять, как конкретный тип, на который они будут ссылаться;
- Все объявления переменных помещать в начало процедуры (модуля) и располагать их по одному объявлению на строку;
- Не писать в одной строке несколько операторов, даже если возможно поместить в одной строке несколько операторов через двоеточие, потому что такую программу будет затруднительно читать;
- Проверять корректность данных, получаемых процедурами в качестве аргументов и введенных пользователями. Прежде чем использовать данные в программе их необходимо проверять, для этого целесообразно использовать, например, функции проверки типов (IsNumeric, IsArray и другие).

Следуя этим рекомендациям можно гарантированно сократить время разработки и отладки программы.

Возможные типы ошибок

Все возникающие в программе ошибки можно разделить на три типа:

- Ошибки выполнения (*runtime errors*), возникающие в процессе выполнения программы. Если не предусмотрен механизм их перехвата, то происходит аварийная остановка программы с выдачей стандартного описания такой ошибки;
- Логические ошибки (*bugs*), приводящие к неправильной работе программы, выражающиеся в получении неправильных результатов. Ошибки такого типа не приводят к аварийной остановке программы. Устранение таких ошибок возможно лишь исправлением алгоритма программы, который должен разрабатываться программистом на одном из первых этапов решения поставленной задачи;
- Синтаксические ошибки (*syntax errors*), возникающие в процессе набора текста программы при нарушении правил синтаксиса языка VBA. Каждая строка текста программы в процессе ее набора анализируется редактором.
- Рекомендуется набирать текст программы в нижнем регистре, тогда после перехода на следующую строку редактора введенный текст изменится в соответствии с синтаксисом языка VBA (некоторые буквы отобразятся в верхнем регистре). Если этого не произошло, необходимо проверить набранную строку программы.

Приложение 2

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования
“Кузбасский государственный технический университет имени Т. Ф. Горбачева”

Кафедра прикладных информационных технологий

Отчет

по контрольной работе на тему:
“Алгоритмизация и программирование ”

Выполнил студент группы
(наименование группы)

(Фамилия инициалы)

Кемерово (год выполнения)