**КОНТРОЛЬНАЯ РАБОТА №2.**

Разработка блок-схем программ и их реализация на языке VBA.

**Специфические требования**

***Требования к содержанию отчета:***

Отчет должен состоять из следующих пунктов: 

* титульный лист.
* задание к контрольной работе.
* блок-схемы решение задач и его реализацию на языке VBA (листинг программ).

**Теоретическая часть**

**1.Понятие алгоритма**

Для составления программы, предназначенной для решения на ЭВМ какой-либо задачи, требуется составление алгоритма ее решения.   
  
***Алгоритм — это точное предписание, которое определяет процесс, ведущий от исходных данных к требуемому конечному результату.*** Алгоритмами, например, являются правила сложения, умножения, решения алгебраических уравнений, умножения матриц и т.п. Слово алгоритм происходит от algoritmi, являющегося латинской транслитерацией арабского имени хорезмийского математика IX века аль-Хорезми. Благодаря латинскому переводу трактата аль-Хорезми европейцы в XII веке познакомились с позиционной системой счисления, и в средневековой Европе алгоритмом называлась десятичная позиционная система счисления и правила счета в ней.

Применительно к ЭВМ алгоритм определяет вычислительный процесс, начинающийся с обработки некоторой совокупности возможных исходных данных и направленный на получение определенных этими исходными данными результатов. Термин ***вычислительный процесс***распространяется и на обработку других видов информации, например, символьной, графической или звуковой.

Если вычислительный процесс заканчивается получением результатов, то говорят, что соответствующий алгоритм применим к рассматриваемой совокупности исходных данных. В противном случае говорят, что алгоритм неприменим к совокупности исходных данных. Любой применимый алгоритм обладает следующими ***основными свойствами****:*

• результативностью;

• определенностью;

• массовостью;

• дискретностью;

• конечностью.

***Результативность***означает возможность получения результата после выполнения конечного количества операций.   
  
***Определенность*** состоит в совпадении получаемых результатов независимо от пользователя и применяемых технических средств.   
  
***Массовость*** заключается в возможности применения алгоритма к целому классу однотипных задач, различающихся конкретными значениями исходных данных.   
  
***Дискретность*** означает разбиение алгоритма на конечную последовательность действий или шагов при его выполнении.   
  
***Конечность*** означает то, что алгоритм должен выполняться за конечное время.

Для задания алгоритма необходимо описать следующие его элементы:

• набор объектов, составляющих совокупность возможных исходных данных, промежуточных и конечных результатов;

• правило начала;

• правило непосредственной переработки информации (описание последовательности действий);

• правило окончания;

• правило извлечения результатов.

Алгоритм всегда рассчитан на конкретного исполнителя. В нашем случае таким исполнителем является ЭВМ. Для обеспечения возможности реализации на ЭВМ алгоритм должен быть описан на языке, понятном компьютеру, то есть на языке программирования.

Таким образом, можно дать следующее определение программы.   
  
***Программа для ЭВМ***представляет собой описание алгоритма и данных на некотором языке программирования, предназначенное для последующего автоматического выполнения.

**2.Способы описания алгоритмов**

К основным способам описания алгоритмов можно отнести следующие:

• словесно-формульный;

• структурный или блок-схемный;

• с помощью граф-схем;

• с помощью сетей Петри.

Перед составлением программ чаще всего используются словесно-формульный и блок-схемный способы. Иногда перед составлением программ на низкоуровневых языках программирования типа языка Ассемблера алгоритм программы записывают, пользуясь конструкциями некоторого высокоуровнего языка программирования. Удобно использовать программное описание алгоритмов функционирования сложных программных систем. Так, для описания принципов функционирования ОС использовался Алголоподобный высокоуровневый язык программирования.

При ***словесно-формульном способе***алгоритм записывается в виде текста с формулами по пунктам, определяющим последовательность действий.

Пусть, например, необходимо найти значение следующего выражения:

у = 2а – (х+6).

Словесно-формульным способом алгоритм решения этой задачи может быть записан в следующем виде:

1. Ввести значения *а*и *х.*

2. Сложить х и 6.

3. Умножить *a* на 2.

4. Вычесть из *2а*сумму (х+6).

5. Вывести *у*как результат вычисления выражения.

При ***блок-схемном***описании алгоритм изображается геометрическими фигурами (блоками), связанными по управлению линиями (направлениями потока) со стрелками. В блоках записывается последовательность действий.

Данный способ по сравнению с другими способами записи алгоритма имеет ряд преимуществ. Он наиболее нагляден: каждая операция вычислительного процесса изображается отдельной геометрической фигурой. Кроме того, графическое изображение алгоритма наглядно показывает разветвления путей решения задачи в зависимости от различных условий, повторение отдельных этапов вычислительного процесса и Другие детали.

Оформление программ должно соответствовать определенным требованиям. В настоящее время действует единая система программной документации (ЕСПД), которая устанавливает правила разработки, оформления программ и программной документации. В ЕСПД определены и правила оформления блок-схем алгоритмов (ГОСТ 10.002-80 ЕСПД, ГОСТ 10.003-80 ЕСПД, ГОСТ 19.701-90).

Операции обработки данных и носители информации изображаются на схеме соответствующими ***блоками.***Большая часть блоков по построению условно вписана в прямоугольник со сторонами *а и b.*Минимальное значение ***а*= 10 мм**, увеличение *а*производится на число, кратное **5 мм**. Размер **b=1,5a**. Для от дельных блоков допускается соотношение между *а и b,*равное 1:2. В пределах одной схемы рекомендуется изображать блоки одинаковых размеров. Все блоки нумеруются. Виды и назначение основных блоков приведены в табл. 2.1

Линии, соединяющие блоки и указывающие последовательность связей между ними, должны проводится параллельно линиям рамки. Стрелка в конце линии может не ставиться, если линия направлена слева направо или сверху вниз. В блок может входить несколько линий, то есть блок может являться преемником любого числа блоков. Из блока (кроме логического) может выходить только одна линия. Логический блок может иметь в качестве продолжения один из двух блоков, и из него выходят две линии. Если на схеме имеет место слияние линий, то место пересечения выделяется точкой. В случае, когда одна линия подходит к другой и слияние их явно выражено, точку можно не ставить.

***Схему алгоритма***следует выполнять как единое целое, однако в случае необходимости допускается обрывать линии, соединяющие блоки.

Если при обрыве линии продолжение схемы находится на этом же листе, то на одном и другом конце линии изображается специальный *символ****соединитель****—*окружность диаметром 0,5 *а.*Внутри парных окружностей указывается один и тот же идентификатор. В качестве идентификатора, как правило, используется порядковый номер блока, к которому направлена соединительная линия.

Если схема занимает более одного листа, то в случае разрыва линии вместо окружности используется ***межстраничный соединитель****.*Внутри каждого, соединителя указывается адрес — откуда и куда направлена соединительная линия. Адрес записывается в две строки: в первой указывается номер листа, во второй — порядковый номер блока.

Блок-схема должна содержать все разветвления, циклы и обращения к подпрограммам, содержащиеся в программе.

*Таблица 2.*   
  
**Условные обозначения блоков схем алгоритмов**

|  |  |  |
| --- | --- | --- |
| **Наименование** | **Обозначение** | **Функции** |
| Процесс | http://mognovse.ru/mogno/961/960040/960040_html_53df9960.png | Выполнение операции или группы операций, в результате которых изменяется значение, форма представления или расположение данных. |
| Ввод-вывод | http://mognovse.ru/mogno/961/960040/960040_html_m2eff66fd.png | Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод). |
| Решение | http://mognovse.ru/mogno/961/960040/960040_html_m69c6932c.png   http://mognovse.ru/mogno/961/960040/960040_html_m7da8ac8c.png | А) Выбор направления выполнения алгоритма в зависимости от некоторых переменных условий.  Б) Выбор одного из N направлений выполнения алгоритма, в зависимости от некоторых условий. |
| Модификация | http://mognovse.ru/mogno/961/960040/960040_html_6eb112d0.png | Организация циклических конструкций |
| Предопределенный процесс | http://mognovse.ru/mogno/961/960040/960040_html_32884f37.png | Использование ранее созданных и отдельно написанных программ (подпрограмм). |
| Документ | http://mognovse.ru/mogno/961/960040/960040_html_m2734bef6.png | Вывод данных на бумажный носитель. |
| Магнитный диск | http://mognovse.ru/mogno/961/960040/960040_html_7a795a34.png | Ввод-вывод данных, носителем которых служит магнитный диск. |
| Дисплей | http://mognovse.ru/mogno/961/960040/960040_html_61dd6d36.png | Ввод-вывод данных, если непосредственно подключенное к процессу устройство воспроизводит данные и позволяет оператору ЭВМ вносить изменения в процессе их обработки |
| Пуск-останов | http://mognovse.ru/mogno/961/960040/960040_html_m149aea84.png | Начало, конец, прерывание процесса обработки данных. |
| Соединитель | http://mognovse.ru/mogno/961/960040/960040_html_382f3c48.png | Указание связи между прерванными линиями, соединяющими блоки. |
| Межстраничный соединитель | http://mognovse.ru/mogno/961/960040/960040_html_46dc438b.png | Указание связи между прерванными линиями, соединяющими блоки, расположенные на разных листах. |
| Комментарий | http://mognovse.ru/mogno/961/960040/960040_html_m111e44d3.png | Связь между элементом схемы и пояснением. |

**3.Структурные схемы алгоритмов**

Одним из свойств алгоритма является ***дискретность****—*возможность расчленения процесса вычислений, предписанных алгоритмом, на отдельные этапы, возможность выделения участков программы с определенной структурой. Можно выделить и наглядно представить графически три простейшие структуры:

• последовательность двух или более операций;

• выбор направления;

• повторение.

Любой вычислительный процесс может быть представлен как комбинация этих элементарных алгоритмических структур. Соответственно, вычислительные процессы, выполняемые на ЭВМ по заданной программе, можно разделить на три основных вида:

• линейные;  
  
• ветвящиеся;

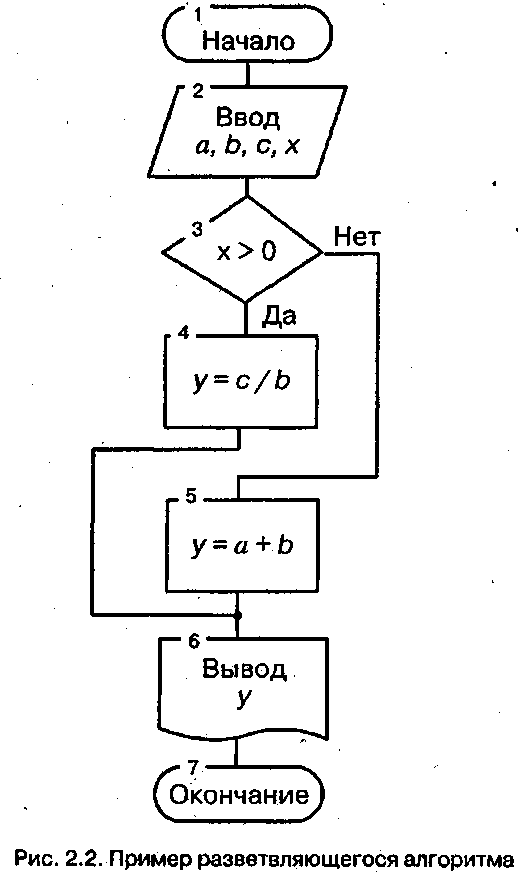
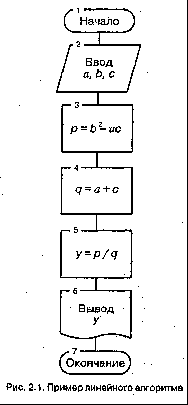
• циклические.

***Линейным***принятоназывать вычислительный процесс, в котором операции выполняются последовательно, в порядке их записи. Каждая операция является самостоятельной, независимой от каких-либо условий. На схеме блоки, отображающие эти операции, располагаются в линейной последовательности.

Линейные вычислительные процессы имеют место, например, при вычислении арифметических выражений, когда имеются конкретные числовые данные и над ними выполняются соответствующие условию задачи действия. На рис. 2.1 показан пример линейного алгоритма, определяющего процесс вычисления арифметического выражения

*у=(b2-ас):(а+с).*

Вычислительный процесс называется ***ветвящимся,***если для его реализации предусмотрено несколько направлений (ветвей). Каждое отдельное направление процесса обработки данных является отдельной ветвью вычислений. Ветвление в программе — это выбор одной из нескольких последовательностей команд при выполнении программы. Выбор направления зависит от заранее определенного признака, который может относиться к исходным данным, к промежуточным или конечным результатам. Признак характеризует свойство данных и имеет два или более значений.

Ветвящийся процесс, включающий в себя две ветви, называется простым, более двух ветвей — сложным. Сложный ветвящийся процесс можно представить с помощью простых ветвящихся процессов.   
  
   
Направление ветвления выбирается логической проверкой, в результате которой возможны два ответа: «да» — условие выполнено и «нет» — условие не выполнено.

Следует иметь в виду, что, хотя на схеме алгоритма должны быть показаны все возможные направления вычислений в зависимости от выполнения определенного условия (или условий), при однократном прохождении программы процесс реализуется только по одной ветви, а остальные исключаются. Любая ветвь, по которой осуществляются вычисления, должна приводить к завершению вычислительного процесса.

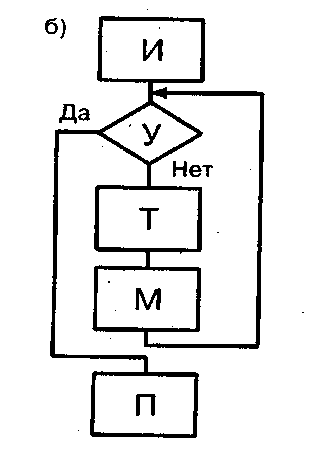
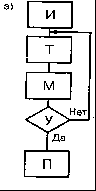
На рис. 2.2. показан пример алгоритма с разветвлением для вычисления следующего выражения:   
  
*Y = (а+b), если Х*<0;   
  
*с/b, если Х>0.*   
  
***Циклическими***называются программы, содержащие циклы. Цикл — это многократно повторяемый участок программы.   
  


Рис. 2.3. Примеры циклических алгоритмов

В организации цикла можно выделить следующие *этапы:*

• подготовка (инициализация) цикла (И);

• выполнение вычислений цикла (тело цикла) (Т);

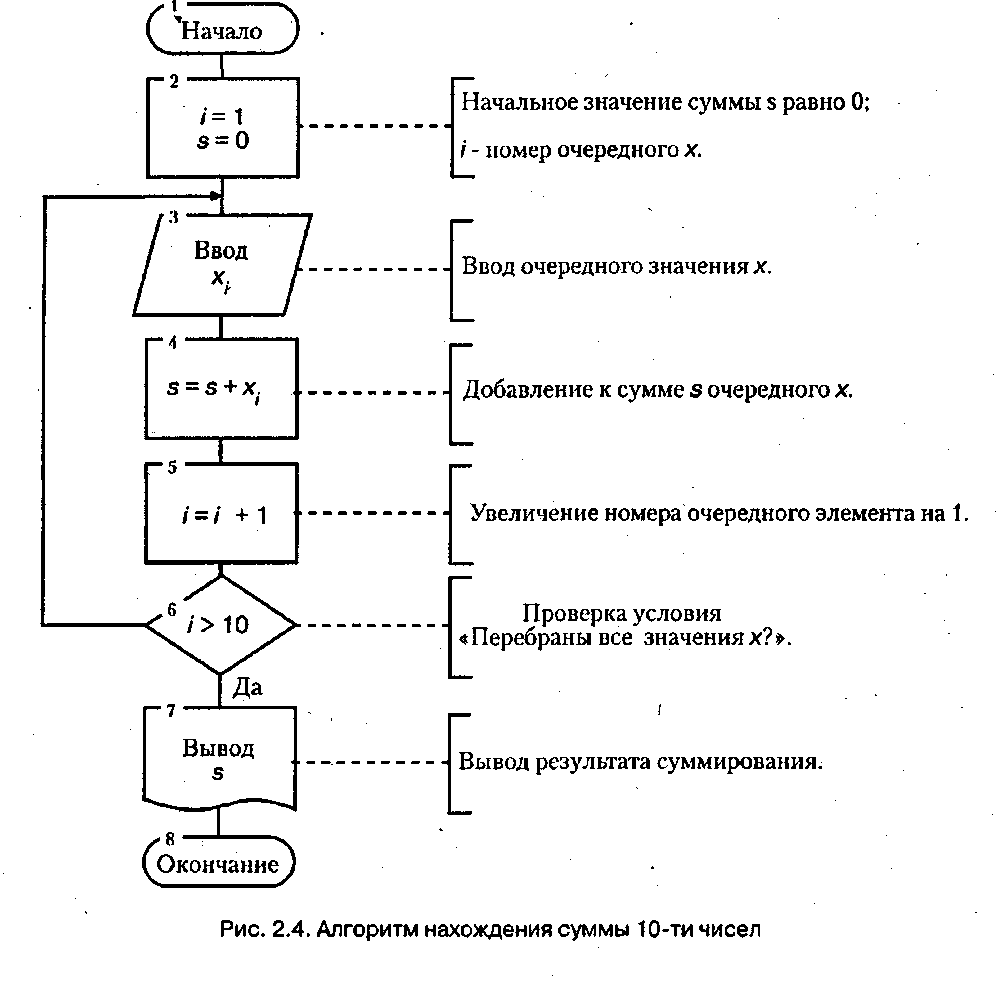
• модификация параметров (М);

• проверка условия окончания цикла (У).

Порядок выполнения этих этапов, например, Т и М, может изменяться. В зависимости от расположения проверки условия окончания цикла различают циклы с нижним и верхним окончаниями (рис. 2.3). Для цикла с нижним окончанием (рис. 2.3 а) тело цикла выполняется как минимум один раз, так как сначала производятся вычисления, а затем проверяется условие выхода из цикла. В случае цикла с верхним окончанием (рис. 2,3 б) тело цикла может не выполниться ни разу в случае, если сразу соблюдается условие выхода.

Цикл называется ***детерминированным,***если число повторений тела цикла заранее известно или определено. Цикл называется ***итерационным,***если число повторений тела цикла заранее неизвестно, а зависит от значений параметров (некоторых переменных), участвующих в вычислениях.

На рис. 2.4 показан пример циклического алгоритма вычисления суммы десяти чисел.

**Список индивидуальных заданий**

Выбор задач для индивидуальной контрольной работы осуществляется согласно вашему номеру в списке группы.

**Задание 1. Линейное программирование**

1. Банк принимает вклады до востребования по простой ставке (i%) годовых. Определить сумму начисленных процентов (I) и наращенную сумму (S) на вклад в размере (Р) тыс. рублей, размещённый на срок (n) лет



2. Вычислить ставку простых процентов по вкладу, при которой первоначальный капитал в размере (Р) тыс. рублей, достигнет (S), размещённый на (n) лет



3. Банк в своих расчётах простую ставку процентов в размере (i%) годовых. Определить период начислении (n), при котором вклад в размере (Р) тыс. рублей достигнет (S)



4. Банк принимает вклады на депозит по простой ставке (i%) годовых. Какую сумму (Р) тыс. рублей необходимо положить в банк на (n) лет, что бы получить (S) тыс. рублей.



5. Банк принимает вклады на годовой депозит с ежемесячной капитализацией процентов под (j%) годовых. Найти будущую стоимость вклада (S), размещённый на срок (n) лет, если минимальный размер составляет (Р).



6. Какой величины достигнет долг, равный (S) тыс. руб., через (n) лет при росте по сложной ставке (i%) годовых.



7. Кредит в размере (Р) тыс. рублей выдан на (а) лет и (b) дней под сложную ставку (i%) годовых. Определить сумму долга (S).



8. Какой размер эффективное ставки сложных процентов (i%) годовых, если номинальная ставка (j%) годовых при ежемесячной капитализации процентов?



9. Инвестиции производятся на протяжении (n) лет один раз в конце года по (R) млн. руб. Ставка сложных процентов (i%) годовых. Найти сумму инвестиций (S) к концу срока



10. Кредит взят под годовую сложную ставку (i%) годовых. Рассчитать размер ежегодного погасительного платежа для погашения кредита объемом (S) тыс. руб. равными выплатами в конце каждого года в течение (n) лет.



**Задание 2. Программирование ветвлений**

1. Найти среди заданных чисел а, b, с, d количество чисел, равных нулю.

2. Переменные а, b, с содержат некоторые целые значения. Если значение переменной b неотрицательно, то поменять местами значения переменных а и с. Ответ выдавать в виде:

а=...; b=...; с=...

3. Найти среди заданных чисел а, b, с, d количество положительных и количество отрицательных чисел.

4. Найти корни квадратного уравнения ax2 + bx + c =0 , заданного коэффициентами а, b, и с. Использовать следующий алгоритм:

если D < 0, то выдать сообщение "Уравнение имеет только мнимые корни";

если D = 0, то вычислить корень и выдать результат в виде "х= ...";

если D > 0, то вычислить корни и выдать результат в виде "х1= ..., х2=…";

Для повторяющихся выражений ввести переменные. Подобрать коэффициенты квадратного уравнения так, чтобы можно было полностью оттестировать программу.

5. Найти корни биквадратного уравнения ax4 +bx2 +c =0 , заданного коэффициентами а, b и с.

6. Дано: а,b ∈ N. Вычислить значение переменной с по формуле



7. Дано: x, y, R∈ℜ, r ≥1 . Выяснить, принадлежит ли точка с координатами (х, у):

А) кругу радиуса R с центром в начале координат;

Б) кольцу с центром в начале координат с внешним радиусом 2R и внутренним — R.

8. Проверить, лежит ли точка (x1; y1 ) на прямой y = ax + b . При положительном ответе найти расстояние от точки Р до начала координат; при отрицательном — найти на прямой точку, имеющую такую же ординату, как у точки Р.

9. Проверить, лежат ли три заданные точки P1 (x1; y1 ), P2(x2; y2 ), P3(x3; y3 ) на одной прямой. Замечание. Три точки P1 (x1; y1 ), P2(x2; y2 ), P3(x3; y3 ) лежат на одной прямой в том и только в том случае, когда



10. Дано х. Вычислить значение функции:



Задание 3. Программирование циклов

1. Найти сумму ряда с точностью Е=10-3 , общий член которого



2. Найти сумму ряда с точностью Е=10-3 , общий член которого



3. Найти сумму ряда с точностью Е=10-3 , общий член которого



4. Найти сумму ряда с точностью Е=10-3 , общий член которого



5. Найти сумму ряда с точностью Е=10-3 , общий член которого



6. Найти сумму ряда с точностью Е=10-3 , общий член которого



7. Найти сумму ряда с точностью Е=10-2 , общий член которого



8. Найти сумму ряда с точностью Е=10-2 , общий член которого



9. Найти сумму ряда с точностью Е=10-2 , общий член которого



10. Найти сумму ряда с точностью Е=10-2 , общий член которого



**Задание 4.Массивы**

1. Cформировать матрицу А (3х3), элементы которой получить случайным образом (0,..., 9). Если среди элементов матрицы А есть нулевые, то заменить их значения на —1.

2. Написать программу для сложения двух матриц (2x2), элементы которых получены случайным образом (0,..., 9)

3. Дана целочисленная матрица А (3 х 3). Элементы каждой строки матрицы поделить на максимальный элемент данной строки. Выдавать 1) исходную матрицу; 2) вектор-столбец максимальных элементов каждой строки; 3) преобразованную матрицу.

4. Дана целочисленная матрица А (3x3). Элементы каждой строки матрицы А умножить на минимальный элемент данной строки. Выдавать 1) исходную матрицу; 2) вектор-столбец минимальных элементов каждой строки; 3) преобразованную матрицу.

5. Дана целочисленная матрица А (m х n). Найти номер строки с максимальной суммой элементов.

6. Дана целочисленная матрица А (m х n). Найти элемент матрицы, расположенный на пересечении строки с максимальной суммой элементов и столбца с минимальной суммой элементов.

7. Дана матрица А (m х n), состоящая из элементов целого типа. Вычислить среднее арифметическое элементов для каждого столбца матрицы.

8. Дана матрица А (m х n), состоящая из элементов целого типа. Вычислить среднее арифметическое элементов для каждой строки матрицы.

9. Поменять местами в матрице А две строки с заданными номерами (номера строк вводятся с клавиатуры).

10. Поменять местами в матрице А два столбца с заданными номерами (номера столбцов вводятся с клавиатуры).