

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ**

А.М. Ковальчук

Ю.А. Луцик

**ЛАБОРАТОРНЫЙ
ПРАКТИКУМ**

В 2-х частях

Часть 1

Минск – 2007

Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

А.М. Ковальчук, Ю.А. Луцик

ЛАБОРАТОРНЫЙ ПРАКТИКУМ
по дисциплине «Основы алгоритмизации и программирования»
для студентов специальности
I-40 02 01 «Вычислительные машины, системы и сети»

В 2-х частях

Часть 1

Минск 2007

УДК
ББК

Ковальчук, А.М., Луцик, Ю.А.

Лабораторный практикум по дисциплине «Основы алгоритмизации и программирования» для студентов специальности I-40 02 01 «Вычислительные машины, системы и сети»: в 2ч.

Ч.1/А.М. Ковальчук, Ю.А. Луцик. – Мн.:БГУИР,2007. –

ISBN

1-я часть лабораторного практикума содержит лабораторные работы по дисциплине «Основы алгоритмизации и программирования», читаемой студентам специальности «Вычислительные машины, системы и сети» в первом семестре. 1-я часть практикума посвящена построению блок-схем алгоритмов, изучению основных типов данных, операторов, циклов, массивов, указателей и функций. Приведены варианты заданий по изучаемым темам.

Введение

Курс "Основы алгоритмизации и программирования" (ОАиП) занимает важное место в учебном процессе по подготовке инженера-системотехника.

Методические указания составлены на основе программы дисциплины "ОАиП" для высших учебных заведений по специальности 40 02 01 "Вычислительные машины, системы и сети".

В результате выполнения лабораторных работ по курсу студенты должны получить знания по теоретическим основам алгоритмизации задач и проектирования программ, создания программ на алгоритмическом языке С(С++), научиться отлаживать и выполнять на ЭВМ конкретные задачи с использованием современных методов программирования.

В процессе выполнения лабораторных работ студенты изучают некоторые стандартные алгоритмы решения задач, а также разрабатывают на их основе новые алгоритмы. Графически разрабатываемые алгоритмы изображаются в виде блок-схем для построения, которых студенты знакомятся со стандартами. Разрабатываемые алгоритмы, реализуются на языке С. Для этого студенты используют знания, получаемые в теоретическом курсе.

Основные вопросы, рассматриваемые при выполнении лабораторных работ: организация ввода/вывода информации, операции и операторы языка, структурированные типы данных, массивы, указатели и операции над указателями, строковые данные, функции.

Материал рассматриваемый на лабораторных работах является основой для изучения дисциплин: "Конструирование программ и языки программирования", "Объектно-ориентированное программирование" и др.

Лабораторная работа №1

Построение блок-схем алгоритмов вычислительных процессов

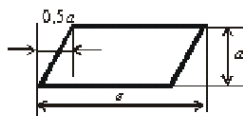
Цель работы: Изучить принципы построения блок-схем алгоритмов линейных, разветвляющихся, циклических и итерационных вычислительных процессов.

Краткие теоретические сведения

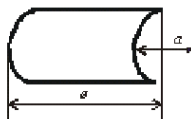
Линейный вычислительный процесс

Вычислительный процесс называется линейным, если направление его продолжения на любом этапе вычислений является единственным. Линейный процесс можно представить в виде следующих этапов: первый – задание исходных данных; второй – реализация вычислений; третий – вывод результатов. Правила построения блок-схем алгоритмов приводятся в ГОСТ 19-002-80 и ГОСТ 19-003-80. Согласно этим ГОСТ, все размеры фигур связаны с двумя величинами: a и b , где a – величина, кратная 5, а b вычисляется по формуле $b=1,5a$, допускается $b=2a$. В январе 1992 года введен новый ГОСТ 19-701-90. Он описывает, как и где следует использовать фигуры. Схемы алгоритмов состоят из имеющих заданное значение символов, краткого пояснительного текста и соединяющих линий.

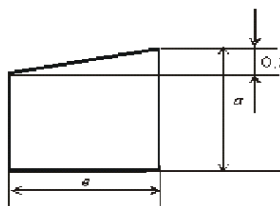
Описание символов



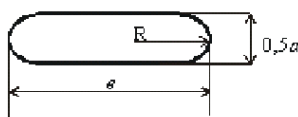
Символ отображает данные, носитель которых не определен



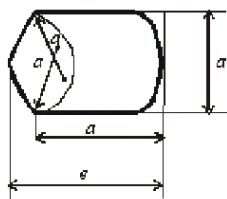
Символ отображает хранимые данные в виде, пригодном для обработки, носитель данных не определен



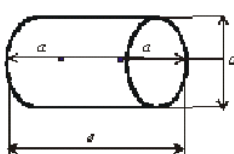
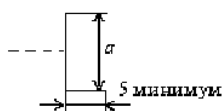
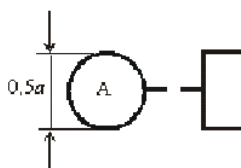
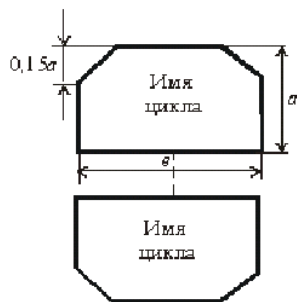
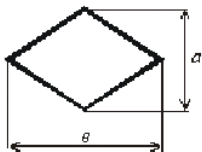
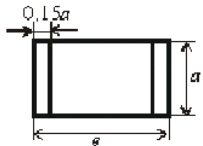
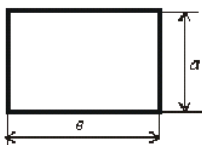
Ручной ввод. Символ отображает данные, вводимые вручную во время обработки устройств любого типа (клавиатура, переключатели, кнопки световое перо).



Символ отображает выход во внешнюю среду и вход из внешней среды (начало или конец схемы программы).



Символ отображает данные, представляемые в человекочитаемой форме на носителе в видеотображающего устройства (экран для визуального наблюдения)



Начальные значения переменных можно задать несколькими способами. Первый способ: исходные данные переменным задаются с помощью оператора присваивания. Второй способ: исходные данные переменным задаются путем ввода их из внешних устройств. На третьем этапе необходимо предусмотреть вывод результатов на удобный для чтения носитель информации.

Пример: Составить блок-схему вычисления значения функции

$$y = \frac{5a * x}{b} - \frac{cx^2}{x^3 - a}$$

при следующих значениях переменных: $x=0.1$; $a=2$; $b=2.5$; $c=3$.

Блок-схема алгоритма приведена на рис.1.

Ошибка! Ошибка связи.

Рисунок 1. Блок-схема линейного вычислительного процесса

Разветвляющимся называют вычислительный процесс, который в зависимости от исходных данных и результатов промежуточных вычислений осуществляется по одному из нескольких возможных вариантов. Варианты (направление вычислений), по которым может реализовываться вычислительный процесс, называют ветвями. Выбор ветви для продолжения вычислений зависит от результатов проверки некоторого условия. Если условие выполняется, то выбирается одна ветвь, если не выполняется, то другая ветвь (рис.2).

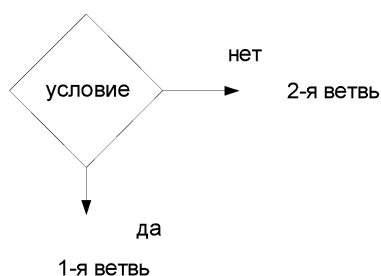
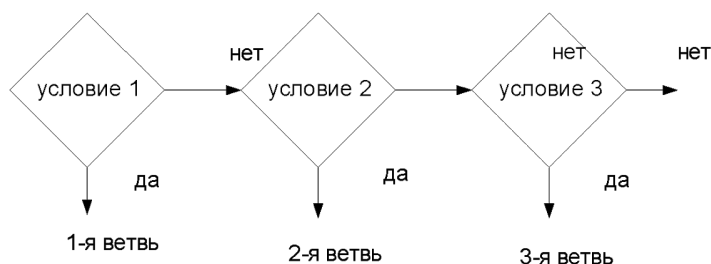


Рисунок 2. Выбор ветви выполнения вычислительного процесса

Условие, определяющее выбор той или иной ветви вычислений, может быть сложным, т.е. состоять из нескольких простых условий (рис.3).



Пример. Составить блок-схему алгоритма нахождения корней квадратного уравнения. Коэффициенты уравнения ввести из клавиатуры (рис.4).

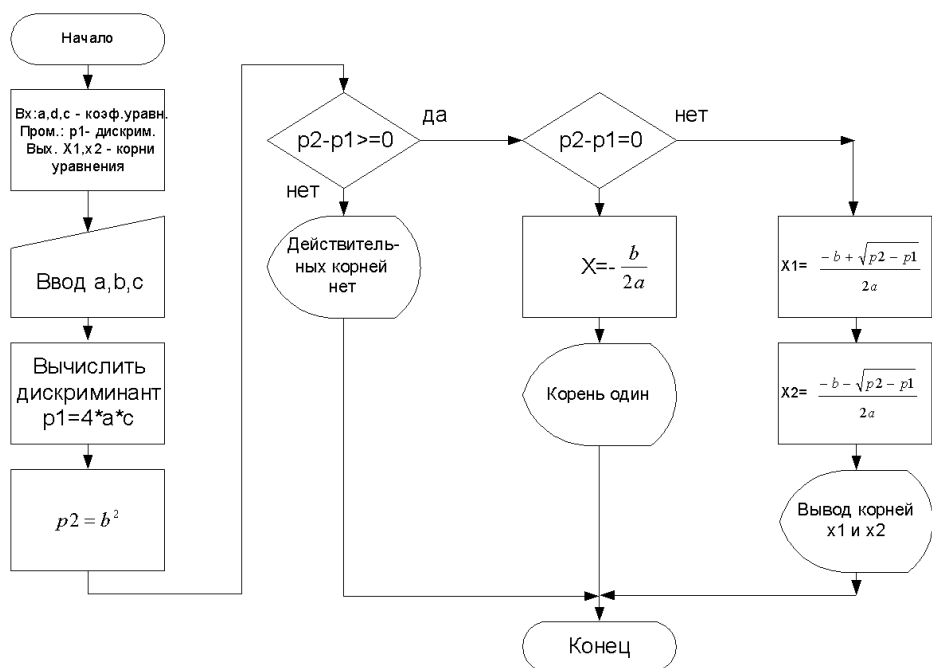


Рисунок 4. Блок-схема алгоритма вычисления корней квадратного уравнения

Построение блок-схем алгоритмов циклических вычислительных процессов

В большинстве задач, встречающихся на практике, необходимо вычисления по некоторой группе формул осуществлять многократно. Этот многократно повторяющийся участок вычислительного процесса называют циклом. Цикл, не содержащий в себя другие циклы, называют простым. Цикл называется сложным, если он содержит внутри себя другие циклы или разветвления. Обычно при каждом повторении цикла вычисления осуществляются с новыми значениями переменных. В любом циклическом процессе в ходе вычислений необходимо решать вопрос: повторять вычисления или нет? Ответ на этот вопрос получают в результате анализа значений одной или нескольких переменных, т.е. анализ некоторого условия. Анализируемую переменную называют параметром цикла. Из вышеизложенного следует, что циклический процесс является разветвляющимся вычислительным процессом с двумя ветвями, из которых одна возвращается на предыдущие блоки, т.е. реализует цикл. Блок-схема циклического процесса, независимо от многообразия сводящихся к нему задач, должна содержать блок задания начального значения параметру цикла (2), блок реализации необходимых вычислений (3), блок изменения параметра цикла (4) и блок проверки условия окончания цикла (5)(рис.5.)

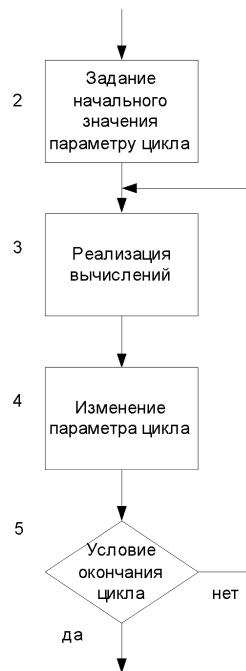


Рисунок 5. Блок-схема циклического процесса

При каждом выполнении цикла выполняются по формулам вычисления (3), изменяются параметры цикла (4), и в зависимости от результата проверки цикл повторяется, начиная с блока 3, или заканчивается.

Пример: составить блок-схему нахождения суммы чисел от 1 до 15 (рис.6).

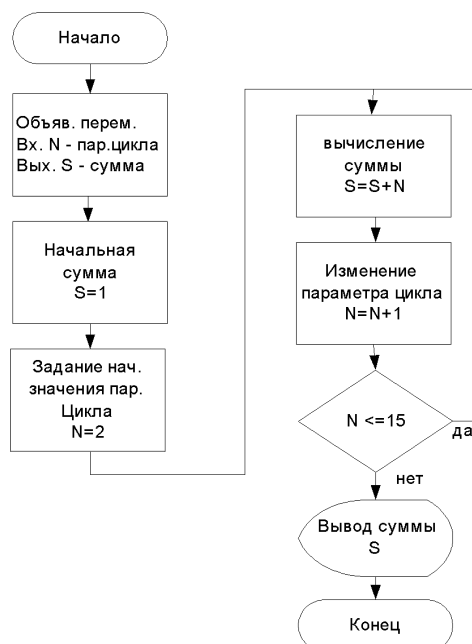


Рисунок 6. Блок-схема алгоритма нахождения суммы чисел

Используя символы для изображения циклов, эту блок-схему алгоритма можно представить в следующем виде (рис.7).

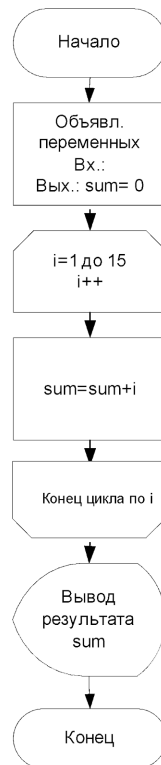


Рисунок 7. Блок-схема циклического процесса

Построение блок-схем алгоритмов итерационных процессов

Итерационным вычислительным процессом называется такой циклический процесс, который продолжается до тех пор, пока разность между соседними, уточняемыми на каждом шаге цикла (итерации) значениями, не окажется меньше или равной некоторой заданной величине. Характерной особенностью итерационного процесса является то, что в нем количество повторений вычислений заранее неизвестно и становится определенным только после окончания вычислений. Решение об окончании вычислений принимается тогда, когда результаты счета (значение функции, искомые величины на очередной отличаются от предыдущих или эталонных не более, чем на некоторую, наперед заданную величину, т.е. найдены с заданной точностью.

Второй особенностью итерационного процесса является то, что результаты вычислений очередного выполнения цикла используются как исходные данные при следующем выполнении цикла, т.е. решение находится последовательными приближениями, путем уточнения на каждом шаге цикла. Пример: для 15 значений переменной X вычислить функцию $\cos x$, используя формулу Тейлора.

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

Вычисления ряда необходимо прекратить, как только очередной учитываемый член ряда окажется по абсолютной величине не больше заданного числа ξ (но в разложении ряда не должно быть более 50 членов). Знаменатель очередного члена ряда рекомендуется вычислить используя значения предыдущего знаменателя и умножая его на очередные два числа натурального ряда чисел. Числитель каждого последующего члена получается из предыдущего умножением на $-x^2$. Это одновременно позволит изменять знак каждого очередного члена (рис.8).

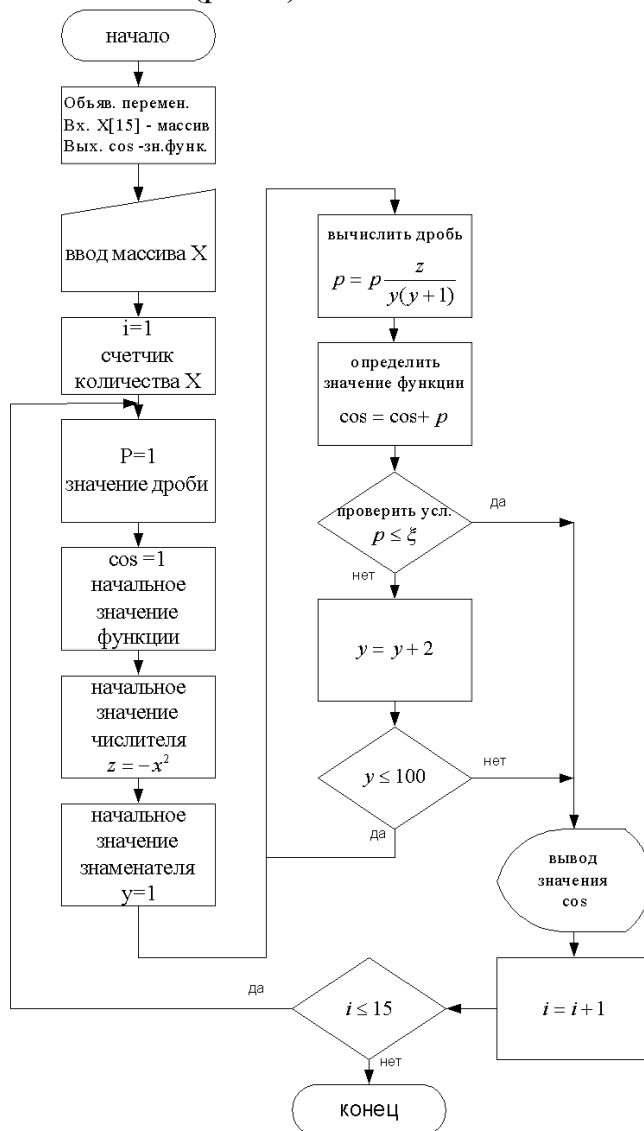


Рисунок 8. Блок-схема алгоритма итерационного вычислительного процесса.

Порядок выполнения работы

1. Изучить краткие теоретические сведения.

2. Составить блок-схему алгоритма.