Министерство науки и высшего образования

Российской Федерации

Братский педагогический колледж

федерального государственного бюджетного
образовательного учреждения высшего образования

«Братский государственный университет»

АЛГОРИТМЫ

Практические рекомендации по учебной дисциплине

«Информатика»

для студентов
очной формы обучения
специальности
09.02.04 Информационные системы (по отраслям)

Автор: Е.А. Пичугина

Братск, 2021

Практические рекомендации по дисциплине «Информатика» для студентов специальности 09.02.04 Информационные системы (по отраслям)/ Сост. Е.А.Пичугина – Братск.: БПК ФГБОУ ВО «БрГУ», 2021 г. – 31 с.

Методические рекомендации по учебной дисциплине «Информатика» для студентов специальности 09.02.04 Информационные системы (по отраслям) содержат практический материал по теме «Алгоритмы».

Печатается по решению научно-методического совета
Братского педагогического колледжа ФГБОУ ВО «БрГУ»
665709, г. Братск, ул. Макаренко, 40

СОДЕРЖАНИЕ:

ВВЕДЕНИЕ

Под компьютерной грамотностью понимается умение находить и воспринимать информацию, применяя компьютерные технологии, создавать объекты и устанавливать связи в гиперсреде, включающей в себя все типы и носители информации; конструировать объекты и действия в реальном мире и его моделях с помощью компьютера (Институт новых технологий образования).

Она является элементом информационной культуры личности, предполагающей способность человека осознать и освоить информационную картину мира как систему символов и знаков, прямых и обратных информационных связей и свободно ориентироваться в информационном обществе, адаптироваться к нему.

Для этого ему необходимо овладеть сводом правил поведения в таком обществе, способами общения с системами телекоммуникаций, локальными и глобальными информационно-вычислительными сетями.

«Умея работать с необходимыми в повседневной жизни вычислительными и информационными системами, базами данных и электронными таблицами, персональными компьютерами и информационными сетями, человек информационного общества приобретает не только инструменты деятельности, но и новое видение мира».

ЭТАПЫ РЕШЕНИЯ ЗАДАЧ НА ЭВМ

Моделирование и формализация

С различными моделями и модельными представлениями мы встречаемся каждодневно и ежечасно. Моделями являются фотография, карта дорог, географическая карта, рисунок, картина, различные описания и т.д.

Модели незаменимы в физике, химии, биологии, математике, информатике. Выбор модельных представлений часто определяет успех научных исследований, т.к. от него зависит точность и достоверность получаемых выводов, прогнозов и рекомендаций.

Моделирование – это метод познания, состоящий в создании и исследовании моделей.

Модели по своей сути – чисто информационное понятие.

Модель – отражение наиболее существенных признаков, свойств, отношений, явлений, объектов или процессов предметного мира (фотографии и рисунки – это представление внешнего вида предметов, а чертежи раскрывают их структуру (внутреннюю организацию)).

Для одних и тех же объектов, процессов и явлений можно построить различные модели, а разные объекты могут описываться одной моделью. Многообразие модельных представлений, связываемых с одними и теми же объектами, отражает различие точек зрения, интересов, потребностей людей в изучении этих объектов. В математике и информатике модели служат основой для постановки задачи.

Формы представления моделей

Все модели можно разбить на два больших класса: моделии*предметные (материальные)*и моделии*информационные.*Предметные модели воспроизводят геометрические, физические и другие свойства объектов в материальной форме (глобус, анатомические муляжи, модели кристаллических решеток, макеты зданий и сооружений и др.).

Информационные модели представляют объекты и процессы в*образной*или*знаковой форме.*

Образные модели (рисунки, фотографии и др.) представляют собой зрительные образы объектов, зафиксированные на каком-либо носителе информации (бумаге, фото- и кинопленке и др.). Широко используются образные информационные модели в образовании (учебные плакаты по различным предметам) и науках, где требуется классификация объектов по их внешним признакам (в ботанике, биологии, палеонтологии и др.).

Знаковые информационные модели строятся с использованием различных языков (знаковых систем). Знаковая информационная модель может быть представлена в форме текста (например, программы на языке программирования), формулы (например, второго закона Ньютона$P = m \cdot a$), таблицы (например, периодической таблицы элементов Д.И. Менделеева) и так далее.

Работа по решению задач с использованием компьютера проходит через следующие этапы:

1) Постановка задачи.

На этапе постановки задачи должно быть четко определено, что дано, и что требуется найти. Так, если задача конкретная, то под постановкой задачи понимают ответ на два вопроса: какие исходные данные известны и что требуется определить. Если задача обобщенная, то при постановке задачи понадобится еще ответ на третий вопрос: какие данные допустимы. Таким образом, постановка задачи включает в себя следующие моменты: сбор информации о задаче; формулировку условия задачи; определение конечных целей решения задачи; определение формы выдачи результатов; описание данных (их типов, диапазонов величин, структуры и т. п.).

Поиск решения любой задачи начинается с анализа ее условий. Результатом анализа условий должна стать четкая постановка задачи, в которой должны быть ответы на четыре вопроса:

1. Что дано? Что требуется? Какие данные допустимы? Какие результаты будут правильными, а какие нет?

*Результатом первого этапа*должно стать построение описательной информационной модели.

2) Формализация.

Правильность результатов решения задачи с помощью компьютера зависит, прежде всего, от правильности выбранного

метода решения. Метод решения является правильным, если для любых допустимых исходных данных он приводит к получению результатов, соответствующих постановке задачи. Для решения задач с помощью компьютера соответствующим методам необходимо дать математическую интерпретацию.

На этом этапе строится математическая модель - система математических соотношений - формул, уравнений, неравенств и т. д., отражающих существенные свойства объекта или явления. Необходимо отметить, что при построении математических моделей далеко не всегда удается найти формулы, явно выражающие искомые величины через данные. В таких случаях используются математические методы, позволяющие дать ответы той или иной степени точности.

В случае большого числа параметров, ограничений, возможных вариантов исходных данных модель явления может иметь очень сложное математическое описание (правда, реальное явление еще более сложно), поэтому часто построение математической модели требует упрощения требований задачи. Необходимо выявить самые существенные свойства объекта, явления или процесса, закономерности; внутренние связи, роль отдельных характеристик. Выделив наиболее важные факторы, можно пренебречь менее существенными.

Итак, создавая математическую модель для решения задачи, нужно: выделить предположения, на которых будет основываться математическая модель; определить, что считать исходными данными и результатами; записать математические соотношения, связывающие результаты с исходными данными.

3) Алгоритмизация.

Наиболее эффективно математическую модель можно реализовать на компьютере в виде алгоритмической модели. Для этого может быть использован язык блок-схем или какой-нибудь псевдокод, например учебный алгоритмический язык. Разработка алгоритма включает в себя выбор метода проектирования алгоритма; выбор формы записи алгоритма (блок-схемы, псевдокод и др.); выбор тестов и метода тестирования; проектирование самого алгоритма.

4) Программирование.

Первые три этапа - это работа без компьютера. Дальше следует собственно программирование на определенном языке в определенной системе программирования. Программирование включает в себя следующие виды работ: выбор языка программирования; уточнение способов организации данных; запись алгоритма на выбранном языке программирования.

Справедливости ради, надо сказать, что этот этап решения задачи было бы правильнее назвать "Компьютерным моделированием", т.к. при решении некоторых задач можно обойтись без составления программы на языке программирования, это можно успешно сделать, используя современные приложения (электронные таблицы, системы управления базами данных и пр.). В этом случае не понадобится и следующий этап - отладка и тестирование программы, а вот проведение расчетов и анализ полученных результатов следует проводить с особой тщательностью.

5) Отладка и тестирование программы.

Под отладкой программы понимается процесс испытания работы программы и исправления обнаруженных при этом ошибок. Обнаружить ошибки, связанные с нарушением правил записи программы на языке программирования (синтаксические и семантические ошибки), помогает используемая система программирования. Пользователь получает сообщение об ошибке, исправляет ее и снова повторяет попытку исполнить программу.

Проверка на компьютере правильности алгоритма производится с помощью тестов. Тест - это конкретный вариант значений исходных данных, для которого известен ожидаемый результат. Прохождение теста - необходимое условие правильности программы. На тестах проверяется правильность реализации программой запланированного сценария.

Таким образом, тестирование и отладка включают в себя синтаксическую отладку; отладку семантики (семантика определяет смысловое значение предложений алгоритмического языка) и логической структуры программы; тестовые расчеты и анализ результатов тестирования; совершенствование программы.

6) Компьютерный эксперимент.

Последний этап - это использование уже разработанной программы для получения искомых результатов. Производится

анализ результатов решения задачи и в случае необходимости - уточнение математической модели (с последующей корректировкой алгоритма и программы). Программы, имеющие большое практическое или научное значение, используются длительное время. Иногда даже в процессе эксплуатации программы могут исправляться, дорабатываться.

АЛГОРИТМЫ И СПОСОБЫ ИХ ОПИСАНИЯ

Понятие алгоритма

Для составления программы, предназначенной для решения на ЭВМ какой-либо задачи, требуется составление алгоритма ее решения.

Каждый из нас ежедневно использует различные алгоритмы: инструкции, рецепты, правила и т.п. Обычно мы это делаем не задумываясь.

А Л Г О Р И Т М - это организованная последовательность действий. Эта формулировка не может считаться определением алгоритма, т.к. не объяснены понятия «организованная» и «действия». Абсолютно строгого определения алгоритма не существует. Это - одно из фундаментальных понятий информатики. Такое же, как понятие точки, прямой и плоскости в геометрии, пространства и времени в физике, вещества в химии и т.д.

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.

Слова «algorism» берет начало от имени автора знаменитого персидского учебника по математике — Абу Абд Аллах Мухаммед ибн Муса аль-Хорезми (ок. 825 г.), означающего буквально «Отец Абдуллы, Мухаммед, сын Мусы, уроженец Хорезма». Аральское море в Центральной Азии когда-то называлось озером Хорезм, и район Хорезма (Khwarizm) расположен в бассейне реки Амударьи южнее этого моря. Аль-Хорезми написал знаменитую книгу Китаб аль-джебрваль-мукабала — «Книга о восстановлении и противопоставлении». От названия этой книги, которая была посвящена решению линейных и квадратных уравнений, произошло еще одно слово — «алгебра». Мухаммеда бена Муса аль-Хорезми описал десятичную систему счисления и впервые сформулировал правила выполнения арифметических действий над целыми числами и обыкновенными дробями. Аль-Хорезми стремился к тому, чтобы сформулированные им правила были

понятными. Ему удалось выработать четкий стиль строгого словесного предписания, который не давал читателю возможность уклонится от предписанного или пропустить какие-нибудь действия. Правила в книгах АлХорезми в латинском переводе начинались словами «Алгоризми сказал». В других латинских переводах автор именовался как Адгоритмус. Со временем сами правила стали называть алгоритмами.

Современное значение слова «алгоритм» во многом аналогично таким понятиям, как рецепт, процесс, метод, способ, процедура, программа, но всё-таки слово «algorithm» имеет дополнительный смысл.

Научное определение понятия алгоритма дал АлонзоЧѐрч в 1930 году. Алгоритм означает точное описание некоторого процесса, инструкцию по его выполнению.

Последующие авторы уточняли это определение. Алгоритм — это конечный набор правил, который определяет последовательность операций для решения конкретного множества задач и обладает пятью важными чертами: конечность, определённость, ввод, вывод, эффективность.

По А. А. Маркову в математическом обиходе под алгоритмом принято понимать «точное предписание, определяющее вычислительный процесс, ведущий от варьируемых исходных данных к искомому результату».

Определение, которое дает в своей книге Н.А. Криницкий, звучит так: алгоритм — это правило, сформулированное на некотором языке и определяющее процесс переработки допустимых исходных данных в искомые результаты.

Алгоритм — это всякая система вычислений, выполняемых по строго определённым правилам, которая после какого-либо числа шагов заведомо приводит к решению поставленной задачи.

Общее в этих определениях то, что алгоритм — это предписание. Единого «истинного» определения понятия «алгоритм» нет, так как понятие алгоритма является фундаментальным и не может быть выражено через другие, поэтому его следует рассматривать как неопределяемое.

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:***

- дискретность (прерывность);
- понятность;
- результативность;
- определенность;
- массовость.

Дискретность (прерывность). Описываемый процесс должен быть разбит на последовательность отдельных шагов. Возникающая при этом запись представляет собой упорядоченную совокупность четко разделенных друг от друга предписаний (директив, команд), образующих прерывную (дискретную) структуру алгоритма: только выполнив требования одного предписания, можно приступить к выполнению следующего.

Понятность. Используемые на практике записи алгоритмов составляются с ориентацией на определенного исполнителя. Составляя алгоритм, нужно знать, какие предписания этот исполнитель может понять и исполнить, а какие нет. У каждого исполнителя имеется свой перечень предписаний, которые для него понятны и которые он может исполнить. Такой перечень называют системой команд исполнителя (СКИ). Составляя запись алгоритма для определенного исполнителя, можно использовать лишь те команды, которые входят в его СКИ.

Результативностьозначает возможность получения результата после выполнения конечного количества операций.

Определенностьсостоит в совпадении получаемых результатов независимо от пользователя и применяемых технических средств.

Массовостьзаключается в возможности применения алгоритма к целому классу однотипных задач, различающихся конкретными значениями исходных данных.

Разработка алгоритма – трудоемкая задача, требующая от человека глубоких знаний и больших временных затрат. Решение задачи по готовому алгоритму требует от исполнителя только строгого следования заданным предписаниям. Исполнитель не обязан вникать в смысл того, что он делает, и рассуждать, почему он поступает так, а не иначе, т.е. он действует формально – по командам.

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 ЭВМпредставляет собой описание алгоритма и данных на некотором языке программирования, предназначенное для последующего автоматического выполнения.

Способы описания алгоритмов

К основным способам описания алгоритмов можно отнести следующие:

- словесно-формульный;
- с помощью сетей Петри;
- с помощью граф-схем;
- структурный или блок-схемный.

Перед составлением программ чаще всего используются словесно-формульный и блок-схемный способы.

При этом словесная форма записи не так широко распространена в литературе из-за ее многословности и отсутствия наглядности.

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

Пусть, например, необходимо найти значение следующего выражения:

y = 2a – (x+6).

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

1. Ввести значения *a* и *x*.
2. Сложить x и 6.
3. Умножить *a* на 2.
4. Вычесть из *2a* сумму (x+6).
5. Вывести *y* как результат вычисления выражения.

Сети Петри— математический аппарат длямоделированиядинамических дискретных систем. Впервые описаныв1962 году.

Сети Петри разрабатывались для моделирования систем с параллельными взаимодействующими компонентами. Сети Петри впервые предложилКарл Адам Петри. В докторской диссертации «Связь автоматов» он сформулировал основные понятия теории связи асинхронных компонент вычислительной системы.

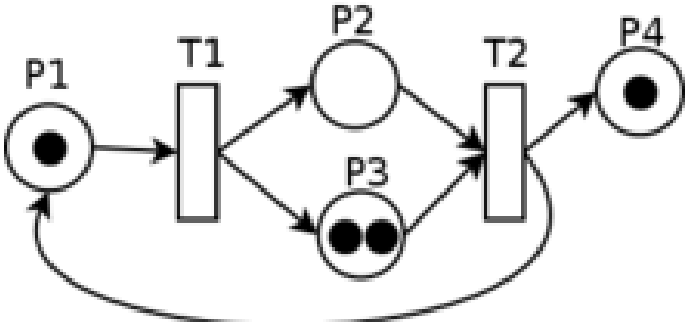


Рис. 1. Пример сети Петри.

На рисунке 1 белыми кружками обозначены позиции, полосками— переходы, чёрными кружками— метки.

Графический способ оказался очень удобным средством изображения алгоритмов и получил широкое распространение в научной и учебной литературе. Связи между шагами можно изобразить в виде графа. Граф, в котором вершинам соответствуют шаги, а ребрам — переходы между шагами, называется блок-схемой алгоритма. Его вершины могут быть двух видов: – из которой выходит одно ребро — операторы; – из которой выходит два ребра — логические условия или предикаты.

Граф— абстрактный математический объект, представляющий собой множество*вершин*графа и набор*рёбер*, то есть соединений между парами вершин (рисунок 2, 3).

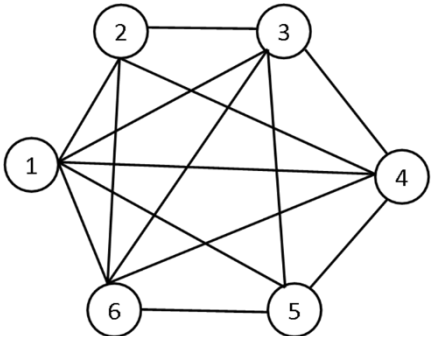


Рисунок 2. Неориентированный граф с шестью вершинами и двенадцатью рёбрами

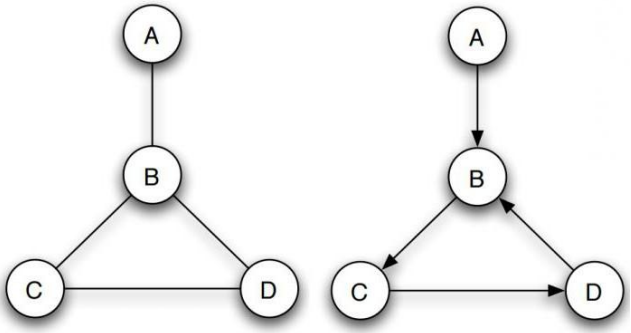


Рисунок 3. Неориентированный и ориентированный граф

Особенностью блок-схем является то, что связи, которые она описывает, не зависят от того, являются ли шаги элементарными или представляют собой самостоятельные алгоритмы — блоки. Для данного блока неважно, как устроены другие блоки; для программирования блока достаточно знать, где лежит исходная информация, какова форма её представления, что должен делать блок и куда записывать результат.

Блок-схемы соответствуют логике, которой пользуется программист для создания сложных, многовариантных, итеративных планов действий. При всей наглядности языка блок-схем не следует переоценивать его возможности. Он отражает связи лишь по управлению.

Блок-схемы не содержат сведений ни о данных, ни о памяти, ни о используемом наборе элементарных шагов. Функциональным блоком называется часть алгоритма, имеющая один вход и один выход.

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.

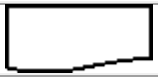
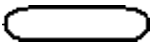
Оформление программ должно соответствовать определенным требованиям. В настоящее время действует единая система программной документации (ЕСПД), которая устанавливает правила разработки, оформления программ и программной документации. В ЕСПД определены и правила оформления блок-схем алгоритмов (ГОСТ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

Операции обработки данных и носители информации изображаются на схеме соответствующими***блоками.***Большая часть блоков по построению условно вписана в прямоугольник со сторонами,*a и b.*Минимальное значение***a* = 10 мм,** увеличение*a*производится на число, кратное**5 мм**. Размер**b=1,5a.** Для отдельных блоков допускается соотношение между*a 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. 1.

Таблица 1. Условные обозначения блоков схем алгоритмов

Наименование	**Обозначение**	**Функции**
Процесс		Выполнение операции или группы операций, в результате которых изменяется значение, форма представления или расположение данных.
Ввод-вывод		Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод).
Решение		Выбор направления выполнения алгоритма в зависимости от некоторых переменных условий.
Предопределенный процесс		Использование ранее созданных и отдельно написанных программ

		(подпрограмм).
		Продолжение Таблицы 1.
Документ		Вывод данных на бумажный носитель.
Магнитный диск		Ввод-вывод данных, носителем которых служит магнитный диск.
Пуск-останов		Начало, конец, прерывание процесса обработки данных.
Соединитель		Указание связи между прерванными линиями, соединяющими блоки.
Межстраничный соединитель		Указание связи между прерванными линиями, соединяющими блоки, расположенные на разных листах.
Комментарий		Связь между элементом схемы и пояснением.

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.

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:

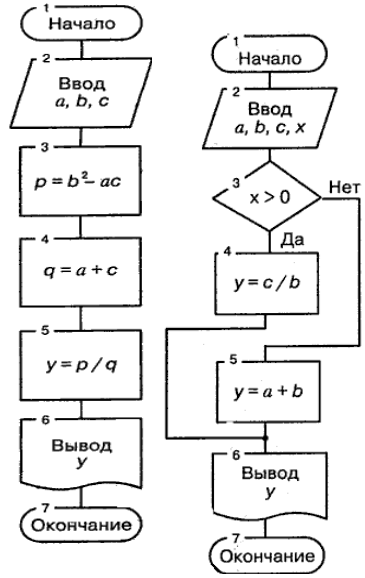
- линейные;
- ветвящиеся;
- циклические.

Линейнымпринятоназывать 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. 4, а показан пример линейного алгоритма, определяющего процесс вычисления арифметического выражения$y=(b^2-ac):(a+c)$.

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.



а б

Рис. 4. Примеры алгоритмов:

а) линейный алгоритм; б) ветвящийся алгоритм

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. 4бпоказан пример алгоритма с разветвлением для вычисления следующего выражения:

$Y = (a+b)$, *если* $X < 0$;
c/b, *если* $X > 0$.

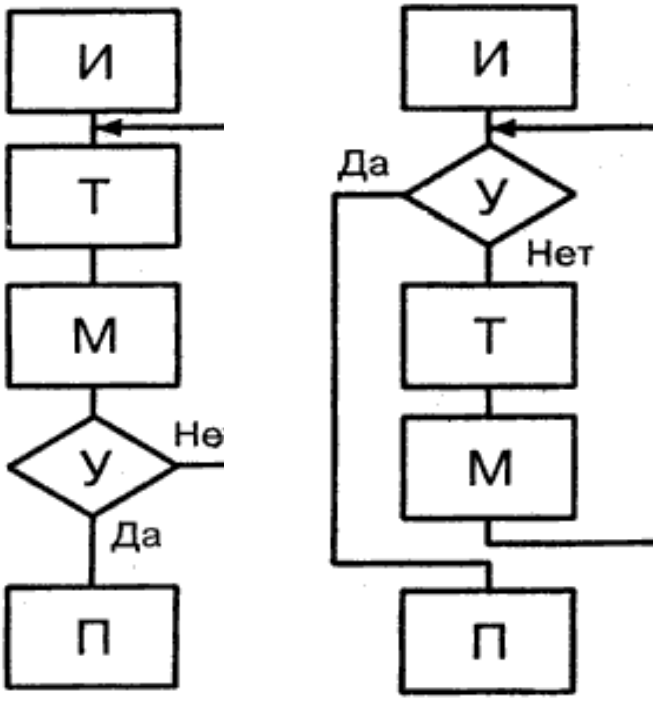
Циклическиминазываются программы, содержащие циклы.

Цикл — это многократно повторяемый участок программы.

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. 5). Для цикла с нижним окончанием (рис. 5а) 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. 5б) тело цикла может не выполниться ни разу в случае, если сразу соблюдается условие выхода.



а б

Рис. 5. Примеры циклических алгоритмов

Цикл называется***детерминированным,*** если число повторений тела цикла заранее известно или определено.

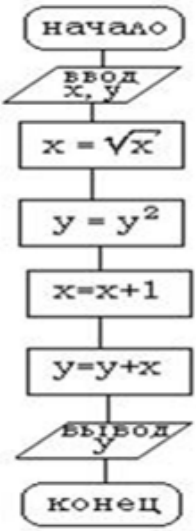
Цикл называется ***итерационным,*** если число повторений тела цикла заранее неизвестно, а зависит от значений параметров (некоторых переменных), участвующих в вычислениях.

На рис. 6 показан пример циклического алгоритма вычисления суммы десяти чисел.

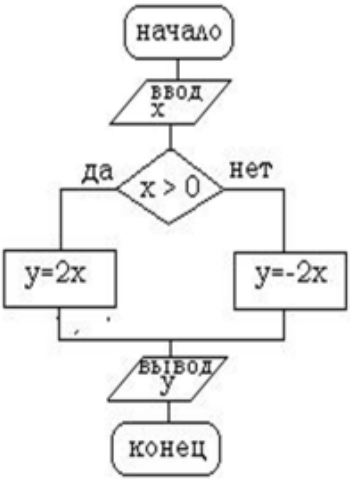
Рис. 6. Алгоритм нахождения суммы 10-ти чисел.

ПРИМЕРЫ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

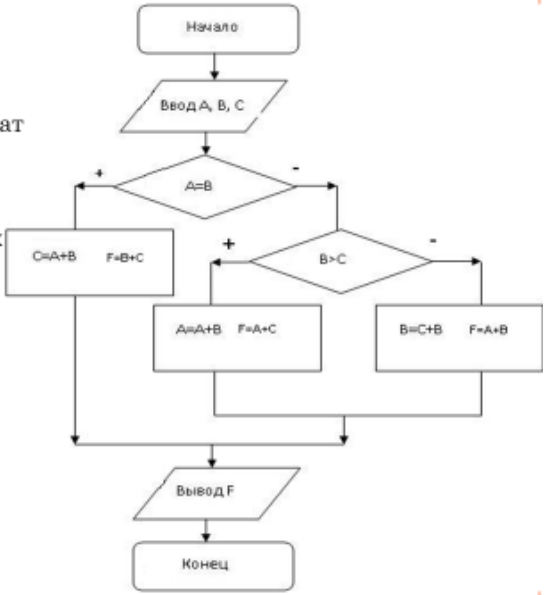
- ***Пример 1.*** Дана блок-схема алгоритма
- Определить результат выполнения алгоритма при определённых значениях исходных данных при x=25 и y=-3



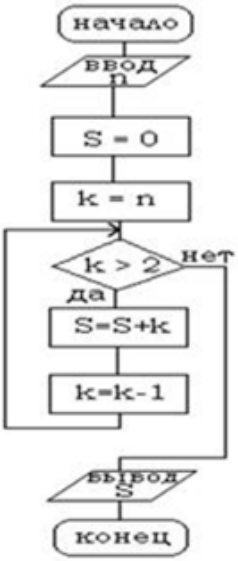
- ***Пример 2.*** Дана блок-схема алгоритма
- Определить результат выполнения алгоритма при определённых значениях исходных данных при
- x=-6
- x=0
- x=7



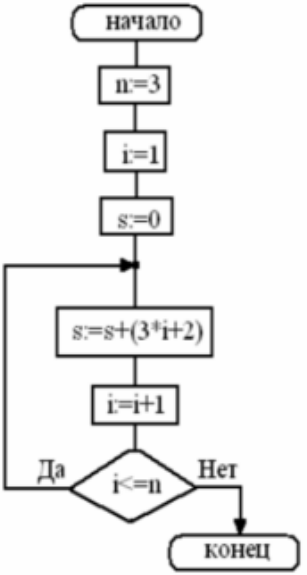
- ***Пример 3.***
- Дана блок-схема алгоритма
- Определить результат выполнения алгоритма при определённых значениях исходных данных
- A=17; B=5; C=4
- A=9; B=9; C=-11
- A=7; B=11; C=-9



- ***Пример 4.***
- Дана блок-схема алгоритма
- Определить результат выполнения алгоритма при определённых значениях исходных данных
- n=8
- n=0

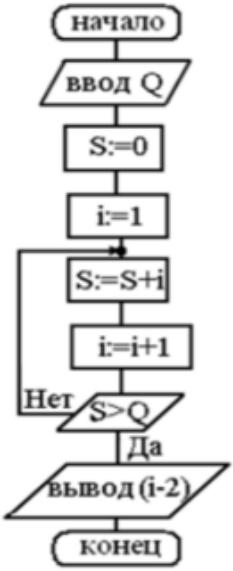


Пример 5.



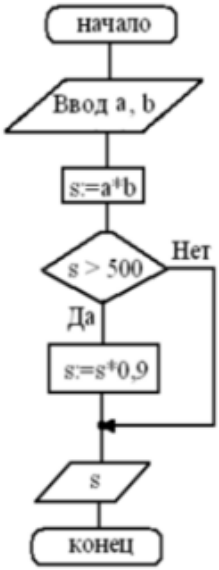
Пример 6.

Реализован некоторый алгоритм в виде блок-схем
получится на выходе блок-схемы, если: Q=2



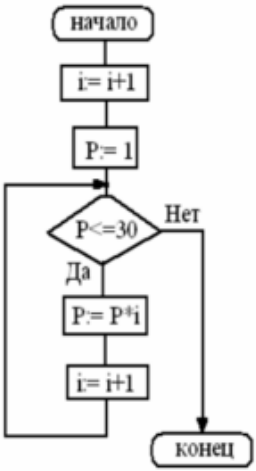
Пример 7.

На блок-схеме представлен алгоритм вычисления стоимости поку с учетом скидки, где a – цена, b количество, s – сумма. Какой будет результат на выходе блок-схемы: a=50, b=8;



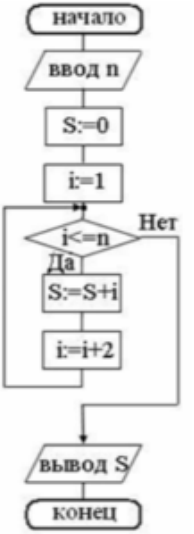
Пример 8.

Дана блок-схема. Тогда после исполнения алгоритма переменная i примет значение...
Начальное значение i = 2.



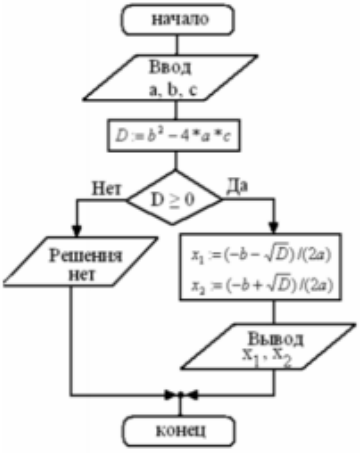
Пример 9.

Реализован некоторый алгоритм в виде блок-схемы. Что получится на выходе блок-схемы, если n = 2.



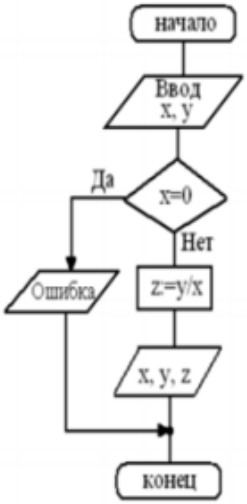
Пример 10.

Используя блок-схему, найти корни уравнения на картинке если: a=1, b=4, c=5;

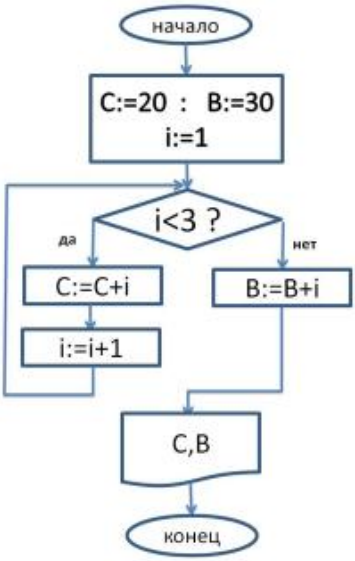


Пример 11.

Реализован некоторый алгоритм в виде блок-схемы
Что получится выходе блок-схемы, если: x=2, y=4.

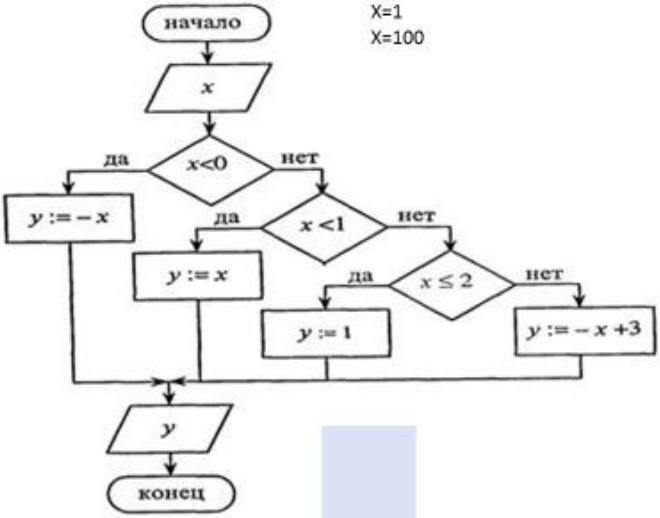


Пример 12.



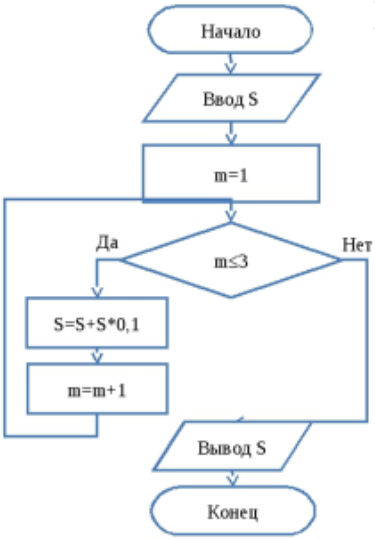
Пример 13.

Вычислить значение y при
X= -10
X=1
X=100



Пример 14.

Гражданин в первый день месяца открыл в банке счет, вложив 10000 у.е.(S) Через каждый месяц (m) увеличивается на 10% от имеющейся суммы. Выполнить блок-схему и определить сумму вклада через 3 месяца.



СПИСОК РЕКОМЕНДУЕМЫХ ИСТОЧНИКОВ:

1. Информатика: учебное пособие / сост. И.П. Хвостова; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». - Ставрополь: СКФУ, 2019. – 178 с. [Электронный ресурс]. - URL:http://biblioclub.ru/index.php?page=book&id=459050.

2. Теория алгоритмов. Учебное пособие. Режим доступа: [http://openedo.mrsu.ru/pluginfile.php/78170/mod_resource/content/1/Теория%20алгоритмов.pdf]

3. Цветкова М.С. Информатика: учеб. Для студ. Учреждений сред.проф. Образования / М.С. Цветкова, И.Ю. Хлобыстоыв.-5-е изд., стер.-М.: Издательский центр "Академия", 2018-352с.: ил.