

РОСЖЕЛДОР
Федеральное государственное бюджетное образовательное учреждение
высшего образования
Ростовский государственный университет путей сообщения
(ФГБОУ ВО РГУПС)
Лискинский техникум железнодорожного транспорта имени И.В. Ковалёва
(ЛТЖТ – филиал РГУПС)

Информатика
конспект лекций
по разделу:
«Принципы обработки информации при помощи
компьютера»

для специальностей
23.02.06 Техническая эксплуатация подвижного состава железных дорог
23.02.01 Организация перевозок и управление на транспорте

Лиски
2017

УДК 51

Конспект лекций по разделу «Принципы обработки информации при помощи компьютера» предназначены для студентов 1 курса очного отделения специальностей 23.02.01 Организация перевозок и управления на транспорте (по видам) (железнодорожный транспорт), 23.02.06 Техническая эксплуатация подвижного состава железных дорог

Автор

Лапыгина С.Н., преподаватель ЛТЖТ – филиала РГУПС

Рецензент

Меркулова В.А., преподаватель ЛТЖТ - филиала РГУПС

Рассмотрено на заседании цикловой комиссии математических и естественно-научных дисциплин протокол от « » 2017 г. №

_____ С.Н. Лапыгина

Рекомендовано методическим советом ЛТЖТ – филиала РГУПС,
протокол от « » 2017 г. №

_____ Т.В. Сергеева

Аннотация

Конспект лекций по разделу «Принципы обработки информации при помощи компьютера» может быть использован для изучения теоретического материала по данному разделу. Также в работе приведены и разобраны типовые задачи, что позволяет студентам освоить методику разработки различных видов алгоритмов и представления их в виде блок-схем. Конспект лекций содержит перечень задач, которые можно рекомендовать студентам для самостоятельного решения, а также задания для теста, используемые для контроля уровня усвоения студентами данной темы дисциплины «Информатика». Проверка знаний выполняется с помощью программы MytestStudent.

Содержание

Тема: Программный принцип работы компьютера. Понятие алгоритма. Свойства алгоритма. Понятие программы и языка программирования. Основные алгоритмические конструкции. Этапы создания программы для выполнения её на компьютере.....	5
Тема: Способы задания алгоритмов: словесное описание, графический способ задания алгоритма, псевдоалгоритмический язык. Разработка алгоритмов линейной структуры.	10
Тема: Разработка алгоритмов разветвляющейся структуры	14
Тема: Разработка алгоритмов циклической структуры.	20
Вопросы теста проверки усвоенных знаний и умений:	26

Тема: Программный принцип работы компьютера. Понятие алгоритма. Свойства алгоритма. Понятие программы и языка программирования. Основные алгоритмические конструкции. Этапы создания программы для выполнения её на компьютере

В основу работы компьютеров положен программный принцип управления, состоящий в том, что компьютер выполняет действия по заранее заданной программе. Этот принцип обеспечивает универсальность использования компьютера: в определенный момент времени решается задача соответственно выбранной программе. После ее завершения в память загружается другая программа и т.д.

Программа - это запись алгоритма решения задачи в виде последовательности команд или операторов языком, который понимает компьютер. Конечной целью любой компьютерной программы является управление аппаратными средствами.

Для нормального решения задач на компьютере нужно, чтобы программа была отлажена, не требовала доработок и имела соответствующую документацию. Поэтому, относительно работы на компьютере часто используют термин программное обеспечение (software), под которым понимают совокупность программ, процедур и правил, а также документации, касающихся функционирования системы обработки данных.

Программное и аппаратное обеспечение в компьютере работают в неразрывной связи и взаимодействии. Создание программы – сложный и кропотливый процесс, включающий несколько этапов. Решение задачи начинается с постановки задачи. Важным этапом является разработка алгоритма. Рассмотрим что же принято называть алгоритмом.

Алгоритм – полное и точное предписание исполнителю выполнить определенную последовательность действий, направленных на решение поставленной задачи или достижение указанной цели.

Происхождение самого термина «алгоритм» связано с математикой. Это слово происходит от Algorithmi – латинского написания имени Мухаммеда аль-Хорезми (787 – 850), выдающегося математика средневекового Востока. Ал(аль)-Хорезми (полное имя - Мухаммед бен Муса ал-Хорезми, т.е. Мухаммед сын Мусы из Хорезма) жил в IX веке в Багдаде, изучал математику, астрономию, географию. В одном из своих трудов он описал десятичную систему счисления, сформулировал правила записи натуральных чисел с помощью арабских цифр, правила выполнения арифметических действий над целыми числами столбиком, выполнение действий над обыкновенными дробями. Арабский оригинал этой книги был утерян, но остался латинский

перевод XII в., по которому Западная Европа познакомилась с десятичной системой счисления и правилами выполнения арифметических действий.

Ал-Хорезми стремился к тому, чтобы сформулированные им правила были понятными. Достичь этого в IX в., когда еще не была разработана математическая символика (знаки операций, скобки, буквенные обозначения и т.д.), было трудно. Однако ему удалось выработать четкий стиль строгого словесного предписания, который не давал читателю возможности уклониться от предписанного или пропустить какие-нибудь действия. Правила в книгах см-Хорезми в латинском переводе начинались словами «Алгоризми сказал». В других латинских переводах автор именовался как Алгоритмус. Со временем было забыто, что Алгоризми (Алгоритмус) - это автор правил, и эти правила стали называть алгоритмами. Многие столетия разрабатывались алгоритмы для решения все новых и новых классов задач, но само понятие алгоритма не имело точного математического определения.

Исполнители алгоритма

Алгоритм может быть предназначен для выполнения его человеком, вычислительной машиной (компьютером) или автоматическим устройством. Создание алгоритма, пусть даже самого простого, - процесс творческий.

При создании алгоритма необходимо следить, чтобы он удовлетворял свойствам алгоритмов, позволяющих отличать алгоритмы от других инструкций.

Свойства алгоритмов:

- *Дискретность (прерывность, раздельность)* – алгоритм должен быть разбит на конечное число шагов (действий). Каждое действие, предусмотренное алгоритмом, исполняется только после того, как закончилось исполнение предыдущего.

- *Определенность* – каждый шаг алгоритма должен быть четким, однозначным и задавать выполнение определенного действия. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.

- *Результативность (конечность)* – алгоритм за конечное число шагов должен приводить исполнителя к решению задачи.

- *Массовость* – алгоритм решения задачи разрабатывается в общем виде, то есть, он должен быть применим для некоторого класса задач, различающихся только исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

- *Понятность* – каждый шаг алгоритма должен быть понятен исполнителю.

Основные алгоритмические конструкции (виды алгоритмов)

Линейный алгоритм – это алгоритм, представляющий набор команд (указаний), выполняемых последовательно друг за другом в том порядке, в котором они записаны.

Разветвляющийся алгоритм – алгоритм, предусматривающий несколько путей выполнения действий, однако во время его выполнения выбирается только один путь в зависимости от некоторого условия.

Циклический алгоритм – алгоритм, в котором для получения результата требуется многократное повторение одного и того же действия (одних и тех же операций). К циклическим алгоритмам сводится большинство методов вычислений, перебора вариантов.

Программа – это алгоритм, записанный на одном из языков программирования. Этот язык понимает компьютер, то есть программа – это алгоритм, исполнителем которого является компьютер или другое автоматическое устройство. Существует много языков и систем программирования, отличающихся уровнем (машиноориентированные языки и языки высокого уровня), назначением (обработка численных задач, текстовых, коммерческих или разработка другого ПО) и т.д.

Этапы создания программы для выполнения её на компьютере

Процесс решения задач на компьютере – это совместная деятельность человека и ЭВМ. Этот процесс можно представить в виде нескольких последовательных этапов. На долю человека приходятся этапы, связанные с творческой деятельностью – постановкой задачи, алгоритмизацией, программированием задач и анализом результатов, а на долю компьютера – этапы обработки информации в соответствии с разработанным алгоритмом.

Первый этап – постановка задачи. На этом этапе работу выполняет человек, хорошо представляющий предметную область задачи. Он должен четко определить цель задачи, дать словесное описание содержания задачи и предложить подход к ее решению. Иначе говоря, на этом этапе формируется условие задачи. Этот этап очень важен, так как от правильного понимания целей и требований зависит в основном успех реализации создаваемой программы.

Второй этап – математическое или информационное моделирование. Цель этого этапа – создать такую математическую модель решаемой задачи, которая может быть реализована в компьютере. Таким образом формируется математическая модель с определенной точностью, допущениями и ограничениями. При этом в зависимости от специфики решаемой задачи могут быть использованы различные разделы математики и

других дисциплин. Могут быть разработаны и проанализированы несколько моделей, впоследствии из них выбирается более подходящая для реализации в виде компьютерной программы. Существует целый ряд задач, где математическая постановка сводится к простому перечислению формул и логических условий.

Третий этап - создание алгоритма и запись его одним из общедоступных способов. Создание алгоритма основывается на математической или информационной модели данной задачи. Разрабатывается алгоритм решения задачи согласно действиям, задаваемым выбранным методом решения.

Четвертый этап – создание программы. На этом этапе алгоритм решения задачи переводится на конкретный язык программирования. Для программирования обычно используются языки высокого уровня, поэтому составленная программа требует перевода ее на машинный язык. После такого перевода получается уже соответствующая машинная программа. *Язык программирования* – формальная знаковая система, предназначенная для записи программ. Со времени создания первых программируемых машин человечество придумало уже более восьми с половиной тысяч языков программирования. Каждый год их число пополняется новыми. Некоторыми языками умеет пользоваться только небольшое число их собственных разработчиков, другие становятся известны миллионам людей. Профессиональные программисты часто применяют в своей работе более десятка разнообразных языков программирования.

Пятый этап - отладка и тестирование программы: заключается в поиске и устранении синтаксических и логических ошибок в программе. В ходе синтаксического контроля программы транслятором выявляются конструкции и сочетания символов, недопустимые с точки зрения правил их построения или написания, принятых в данном языке. Сообщения об ошибках компьютер выдает программисту, при этом вид и форма выдачи подобных сообщений зависят от вида языка и версии используемого транслятора. После устранения синтаксических ошибок проверяется логика работы программы в процессе ее выполнения с конкретными исходными данными.

Несмотря на то, что создание программы происходит в несколько этапов, наиболее важным и трудоемким является этап отладки и тестирования программы. Именно на этом этапе устраняются все логические и синтаксические ошибки в создаваемой программе.

Отладка – это деятельность, направленная на обнаружение и исправление ошибок в программе.

Тестирование – это процесс выполнения программы на некотором наборе данных, для которого заранее известен результат или известны правила поведения этих программ.

Успех отладки в значительной степени предопределяет рациональная организация тестирования. При отладке отыскиваются и устраняются, в основном, те ошибки, наличие которых устанавливается при тестировании. Как было уже отмечено, тестирование не может доказать правильность программы, в лучшем случае оно может продемонстрировать наличие в нем ошибки. Программа, не содержащая синтаксических ошибок, тем не менее, может содержать логические ошибки, не позволяющие программе выполнять заложенные в ней функции. Логические ошибки могут быть связаны с алгоритмом программы или с неправильным пониманием работы аппаратуры, подключённой к портам микроконтроллера.

Для отладки программ обычно применяют три способа:

1. Пошаговая отладка программ с заходом в подпрограммы;
2. Пошаговая отладка программ с выполнением подпрограммы как одного оператора;
3. Выполнение программы до точки останова.

Основными средствами отладки программ можно назвать:

- встроенные программные отладчики,
- внутрисхемный эмулятор;
- внешний программный отладчик. В некоторых случаях используется не интегрированная среда программирования, а отдельный транслятор с выбранного языка программирования.

Отладка программ заключается в проверке правильности работы программы и аппаратуры.

Шестой этап - опытная эксплуатация. На этом этапе производится исполнение отлаженной программы и анализ результатов: после отладки программы ее можно использовать для решения прикладной задачи. При этом обычно выполняется многократное решение задачи на компьютере для различных наборов исходных данных. Получаемые результаты интерпретируются и анализируются специалистом или пользователем, поставившим задачу.

Седьмой этап – промышленная эксплуатация.

Тема: Способы задания алгоритмов: словесное описание, графический способ задания алгоритма, псевдоалгоритмический язык. Разработка алгоритмов линейной структуры.

Любой алгоритм может быть предназначен для выполнения его человеком, компьютером или автоматическим устройством. Создание алгоритма, пусть даже самого простого, - процесс творческий. Существует несколько способов записи алгоритма:

- ✓ словесное описание (или словесно-формульный);
- ✓ графический способ задания алгоритма (блок-схема);
- ✓ запись алгоритма с использованием специальных псевдоалгоритмических языков.

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

Словесное описание (словесно-формульный способ)

При словесно-формульном способе алгоритм записывается в виде текста, может быть с формулами по шагам, определяющим последовательность действий. Каждый шаг есть предложение в повелительном наклонении. Шаги в данной записи алгоритма принято нумеровать.

Пример 1

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 , результат обозначить $R1$.
3. Умножить a на 2 , результат обозначить $R2$.
4. Вычесть $R1$ из $R2$, результат считать значением y .
5. Вывести y как результат вычисления выражения.

Пример 2

Рассмотрим пример алгоритма для нахождения середины отрезка при помощи циркуля и линейки. Зададим алгоритм в виде словесного описания

1. взять циркуль и раскрыть его на величину, большую половины отрезка
2. поставить ножку циркуля в левый конец отрезка и провести дугу;
3. установить ножку циркуля в правый конец отрезка и провести дугу того же радиуса;
4. через точки пересечения дуг провести прямую;
5. точку пересечения этой прямой с отрезком считать серединой отрезка.

Блок-схемы

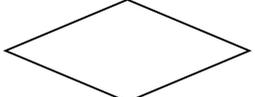
Графический способ задания алгоритма в виде блок-схемы имеет более компактный и наглядный вид. Алгоритм изображается геометрическими фигурами (блоками), соединенными линиями со стрелками. В блоках записывается последовательность действий. Линии определяют порядок их выполнения.

В пределах одной схемы рекомендуется изображать блоки одинаковых размеров. Виды и назначение основных блоков приведены в таблице.

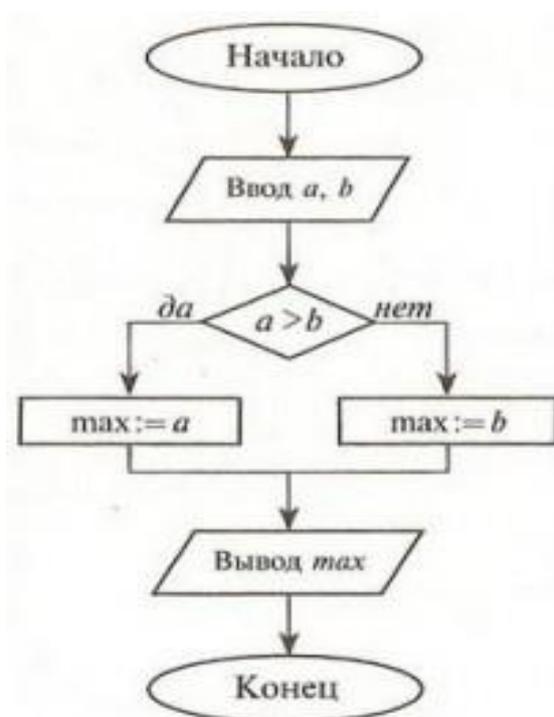
Линии, соединяющие блоки и указывающие последовательность выполнения действий, указанных блоками. Стрелка в конце линии может не ставиться, если линия направлена слева направо или сверху вниз. В блок может входить несколько линий, то есть блок может являться приемником любого числа блоков. Из блока (кроме логического) может выходить только одна линия.

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

Условные обозначения блоков схем алгоритмов

<i>Наименование</i>	<i>Обозначение</i>	<i>Функции</i>
Процесс		Выполнение операции или группы операций
Ввод-вывод		Преобразование данных в форму, пригодную для ввода или вывода результатов обработки.
Блок принятия решения		Выбор направления выполнения алгоритма в зависимости от некоторых переменных условий.
Начало или конец алгоритма		Начало, конец алгоритма
Соединитель		Указание связи между прерванными линиями, соединяющими блоки.
Цикл с параметром		Используется для организации цикла.

Пример 3. Графическое задание алгоритма определения максимального из двух чисел a и b .



Псевдоалгоритмический язык (псевдокод)

Псевдоалгоритмический язык (псевдокод).— компактный (зачастую неформальный) язык описания алгоритмов, использующий ключевые слова некоторых языков программирования, но опускающий несущественные подробности и специфический синтаксис. Главная цель использования псевдокода — обеспечить понимание алгоритма человеком, сделать описание более воспринимаемым, чем на языке программирования. Псевдокод широко используется в учебниках и научно-технических публикациях.

Разработка алгоритмов линейной структуры

Задача. Определить время встречи двух пешеходов, идущих навстречу друг другу, если известно, что расстояние между пешеходами L , скорость первого пешехода V_1 , скорость второго пешехода V_2 . Алгоритм решения задачи представить в виде блок-схемы.

1. *Постановка задачи.*

Дано: L, V_1, V_2 .

Найти: $t, L > 0, V_1 > 0, V_2 > 0, T > 0$

2. *Математическая модель.*

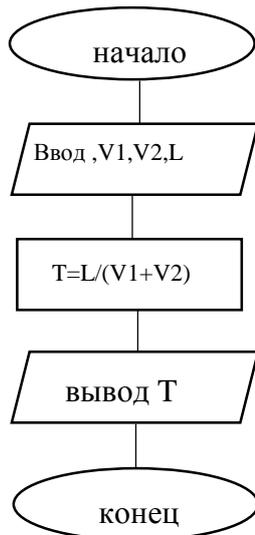
$$L = S_1 + S_2$$

$$S_1 = V_1 * T; S_2 = V_2 * T$$

$$L = V_1 * T + V_2 * T = T * (V_1 + V_2)$$

$$T = L / (V_1 + V_2)$$

Блок-схема:



Задача №1. Длина стороны треугольника равна A , периметр равен P , длины двух других сторон равны между собой. Составить алгоритм определения неизвестные длины сторон, алгоритм представить в виде блок-схемы.

Задача №2. Длины сторон первого прямоугольника A и B , его площадь в 6 раз меньше площади второго прямоугольника. Составить алгоритм определения длины стороны второго прямоугольника, если длина одной из его сторон равна C , алгоритм записать в виде блок-схемы.

Задача №3. Составить алгоритм вычисления площади треугольника со сторонами a , b , c .

Задача №4 Дана длина ребра куба. Составить алгоритм (блок-схему) определения объема куба и площади всей его поверхности.

Задача №5 Известна диагональ квадрата. Составить блок-схему вычисления площади этого квадрата.

Задачи для самостоятельного решения:

Вариант №1.

1 Город A находится в x милях от Лондона. Составьте алгоритм, который определяет расстояние между двумя этими городами в километрах, если известно, что 1.миля=1,609 км. Алгоритм представьте в виде блок-схемы.

2 В прямоугольном треугольнике известны катет и гипотенуза. Составьте алгоритм нахождения площади и периметра данного треугольника.

Вариант №2.

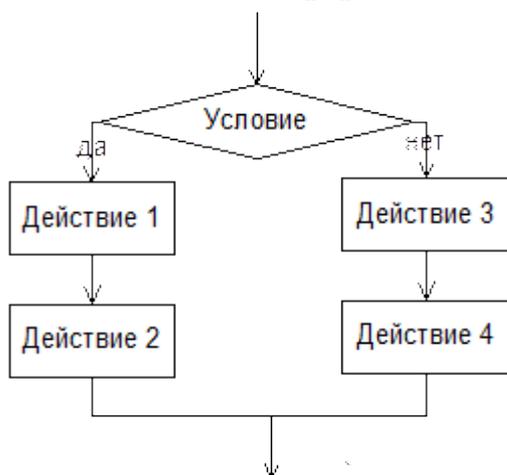
1 Известно, что 1 дюйм=2,54 см. Составьте алгоритм, который переводит размер дискеты из дюймов в сантиметры, алгоритм запишите в виде блок-схемы.

2 Составьте алгоритм вычисления площади равнобедренного треугольника, если известны боковая сторона и высота, проведенная к основанию треугольника. Алгоритм представьте в виде блок-схемы

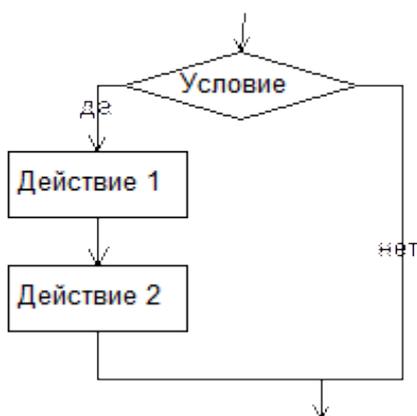
Тема: Разработка алгоритмов разветвляющейся структуры

Алгоритм разветвляющейся структуры

В таком алгоритме данные влияют на последовательность выполнения шагов алгоритма. В алгоритме заложены разные пути следования, по ходу действия выбирается один из возможных вариантов, образуя разветвленную структуру. В зависимости от результата проверки условия выполняются только действия ветви «да» или ветви «нет». Такая структура разветвления носит название полной формы команды ветвления.



В другом случае, в зависимости от результата проверки условия либо выполняются действия ветви «да», либо не происходит выполнения никаких действий, в этом случае команда носит характер неполного ветвления.



Рассмотрим несколько задач, решение которых получается с помощью разветвляющихся алгоритмов.

Задача 1: даны два числа; выбрать большее из них.

Постановка задачи:

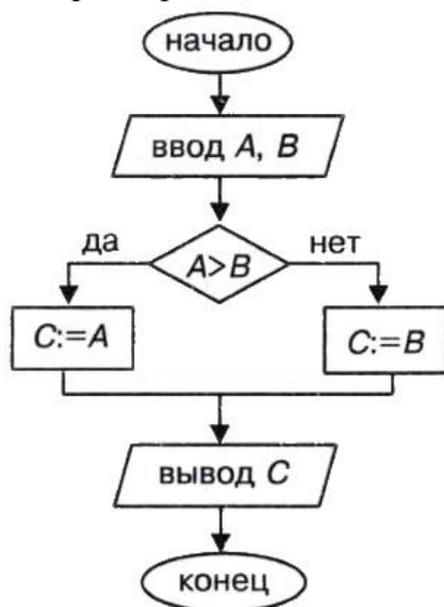
исходными данными являются переменные А и В. Их значения будут задаваться вводом.

Значение большего из чисел должно быть присвоено переменной С и выведено. Например, если $A = 5$, $B = 8$, то должно получиться: $C = 8$.

Нетрудно понять смысл этого алгоритма. Если значение переменной A больше, чем B , то переменной C присвоится значение A . В противном случае, когда $A \leq B$, переменной C присвоится значение B .

Условием, по которому разветвляется алгоритм, является отношение $A > B$. Такое отношение является логическим выражением. Если оно справедливо, то результатом будет логическая величина «истина» и выполнение алгоритма продолжится по ветви «да»; в противном случае логическое выражение примет значение «ложь» и выполнение алгоритма пойдет по ветви «нет».

Запишем алгоритм решения в виде блок-схемы.



Правильность алгоритма проверим путем заполнения таблицы пошагового выполнения алгоритма. Вот как будет выглядеть пошаговая проверка алгоритма для исходных значений $A = 5$, $B = 8$.

шаг	операция	A	B	C	Проверка условия
1	Ввод A,B	5	8	-	
2	Сравнение $A > B$	5	8	-	ложь
3	$C:=B$	5	8	8	
4	Вывод C	5	8	8	

Исполнение команды ветвления происходит в несколько шагов: проверка условия (выполнения логического выражения) и выполнение команд на одной из ветвей «да» или «нет». В пошаговой таблице записываются не команды алгоритма, а операции, выполняемые исполнителем на каждом шаге. Рассмотрим выполнение алгоритма для различных исходных значений переменных A и B

1) $A = 0,2, B = 0,3$

шаг	операция	A	B	C	Проверка условия
1	Ввод A,B	0,2	0,3	-	
2	Сравнение $A > B$	0,2	0,3	-	ложь
3	$C := B$	0,2	0,3	0,3	
4	Вывод C			0,3	

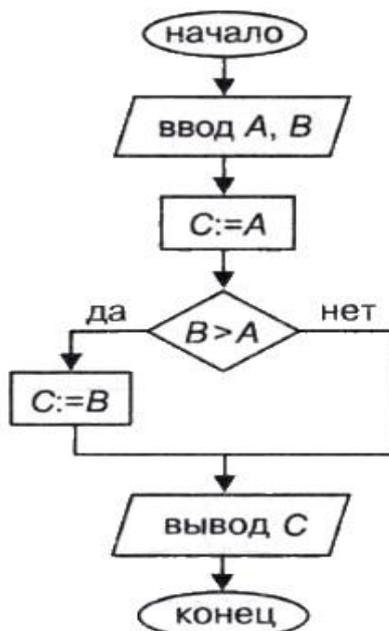
2) $A = 7, B = 4$

шаг	операция	A	B	C	Проверка условия
1	Ввод A,B	7	4	-	
2	Сравнение $A > B$	7	4	-	истина
3	$C := B$	7	4	7	
4	Вывод C	7	4	7	

3) $A = 5, B = 5$

шаг	операция	A	B	C	Проверка условия
1	Ввод A,B	5	5	-	
2	Сравнение $A > B$	5	5	-	ложь
3	$C := B$	5	5	5	
4	Вывод C	5	5	5	

В приведенном выше алгоритме используется полное ветвление. Эту же самую задачу можно решить, применяя команду неполного ветвления. Рассмотрим вариант такого решения поставленной проблемы



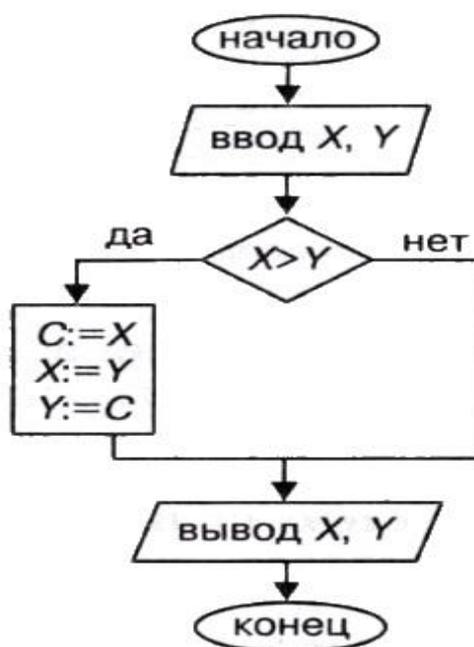
Для программирования характерно то, что одна и та же задача может быть решена с помощью разных алгоритмов. И чем сложнее задача, тем больше существует различных алгоритмов ее решения. Для больших задач

(производственных, научных) практически невозможно точное совпадение алгоритмов, составленных разными программистами.

Задача 2. Упорядочить значения двух переменных X и Y по возрастанию.

Смысл этой задачи следующий: если для исходных значений переменных справедливо отношение $X \leq Y$ (например, $X = 1, Y = 2$), то оставить их значения без изменения; если же $X > Y$ (например, $X = 2, Y = 1$), то выполнить обмен значениями.

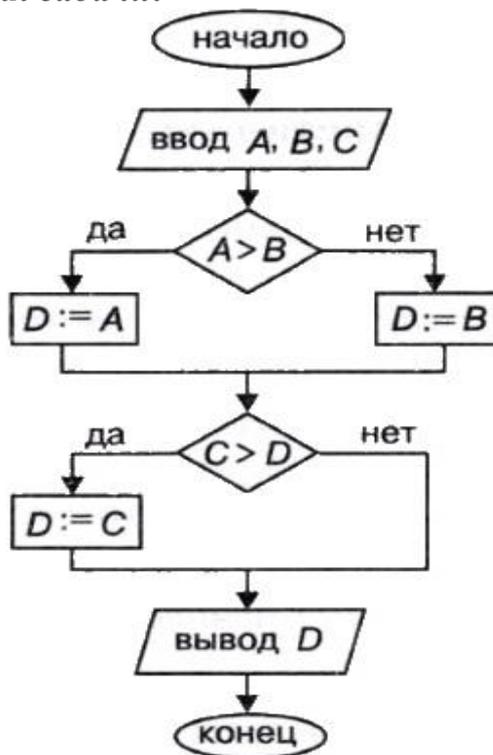
Для обмена значений необходимо использование третьей, вспомогательной переменной C . Кроме того, в алгоритме решения данной задачи рационально используется неполное ветвление. Рассмотрим блок-схему решения задачи:



Задача 3: Составить алгоритм определения наибольшего значения среди трех величин: A, B, C .

Логично будет сначала найти большее из значений A и B и присвоить его вспомогательной переменной, например D ; затем найти большее среди D и C , найденное большее значение можно присвоить все той же переменной D , которая, в результате, получит наибольшее значение среди всех трех величин.

Блок-схема решения задачи:



В структуре этого алгоритма содержатся два последовательных ветвления: первое — полное, второе — неполное.

Рассмотрим пошаговое выполнение алгоритма для $A = 5, B = 7, C = 2$:

шаг	операция	A	B	C	D	Проверка условия
1	Ввод A,B,C	5	7	2	-	
2	Сравнение $A > B$	5	7	2	-	ложь
3	$D := B$	5	7	2	7	
4	Сравнение $C > D$	5	7	2	7	ложь
5	Вывод D	5	7	2	7	

Выводы:

1. В команде ветвления в качестве условия практически всегда используется отношение между величинами, которое может принимать значение истина или ложь.

2. Для решения одной и той же задачи всегда можно построить несколько вариантов алгоритмов.

3. Ветвления в алгоритме могут быть последовательными и вложенными.

Задачи для самостоятельного решения

- Задача 4:** Прием на работу идет на конкурсной основе. Условия приема требуют 20 лет рабочего стажа и возраста не более 42 лет. Составьте алгоритм определения, будет ли человек принят на работу.
- Задача 5:** Вводится оценка за контрольную работу по математике. Выведите "Молодец", если оценка равна 5, «Хорошо», если – 4, «Лентяй» в остальных случаях. Разработайте алгоритм для выполнения задачи.
- Задача 6:** Составьте алгоритм, выбирающий из трех чисел то, которое лежит между двумя другими.
- Задача 7:** Составьте алгоритм упорядочения значений трех переменных по возрастанию, т. е. при любых исходных значениях A, B, C после выполнения алгоритма должно выполняться неравенство $A \leq B \leq C$. Проверьте алгоритм трассировкой для различных вариантов значений исходных данных.
- Задача 8:** Даны значения переменных A, B, C . Составить алгоритм определения количества отрицательных чисел среди них.
- Задача 9:** Даны значения переменных A, B, C . Составить алгоритм определения суммы положительных чисел.
- Задача 10:** Даны два угла треугольника (в градусах). Определить, существует ли такой треугольник. Если да, то будет ли он прямоугольным.

САМОСТОЯТЕЛЬНАЯ РАБОТА

1 вариант

1. Даны два числа m, n . Составить алгоритм, в результате выполнения которого, большее из этих чисел уменьшить на 3, а меньшее – увеличится в 2 раза. Проверить его правильность трассировочной таблицей.
2. Заданы координаты точки x и y , составить алгоритм определения попадания точки в третью четверть окружности.

2 вариант

1. Даны два числа q, w . Большее из этих чисел утроить, а меньшее – уменьшить на два. Составить алгоритм. Проверить его правильность трассировочной таблицей.
2. Заданы координаты точки x_1 и y_1 , составить алгоритм определения попадания точки в четвертую четверть окружности.

Тема: Разработка алгоритмов циклической структуры.

Ранее мы рассмотрели структуры линейного и разветвляющегося алгоритма. Как вы уже успели заметить, обычно, алгоритмы содержат достаточно большое количество команд. Практически не существует чисто линейных, разветвляющихся или иных алгоритмов, чаще всего все эти конструкции можно выделить в любом алгоритме, обычно в нем просматривается и линейный алгоритм, и ветвление и существует несколько команд, которые требуется повторить несколько раз. Повторения в алгоритмах организуют с помощью циклических конструкций. Рассмотрим варианты создания циклов

Алгоритм, в котором для получения результата требуется многократное повторение одних и тех же действий, называется циклическим.

В отличие от линейных алгоритмов, в которых команды выполняются последовательно одна за другой, в циклические алгоритмы входит последовательность команд, выполняемая многократно. Такая последовательность команд называется телом цикла.

Различают три основных типа циклов – *цикл с предусловием*, *цикл с постусловием* и *цикл с параметром*. Отличие первых двух циклов состоит в том, что в них количество повторений заранее не определено.

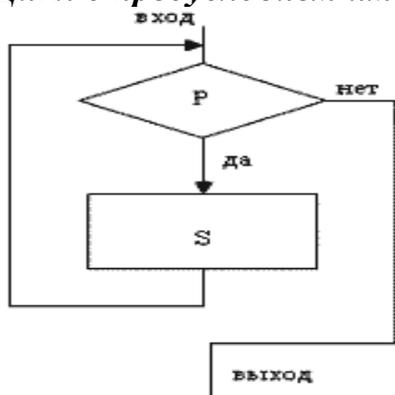
Цикл, как и любая другая алгоритмическая структура, может быть:

- записан на естественном языке;
- изображен в виде блок-схемы;
- записан на алгоритмическом языке;
- закодирован на языке программирования.

Рассмотрим блок-схемы циклических конструкций:

Цикл, для которого нельзя указать число повторения, и проверка окончания которого происходит по достижению нужного условия, называется итерационным. Такие циклические алгоритмы реализуют с помощью цикла с предусловием или цикла с постусловием.

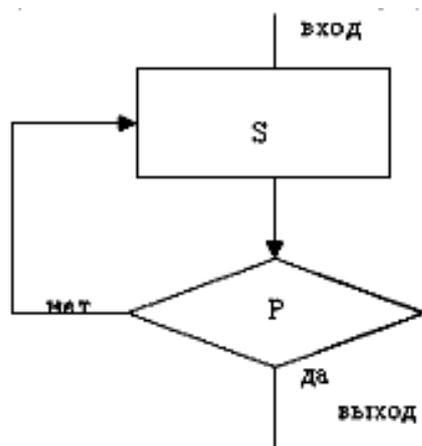
Цикл с предусловием имеет вид:



Здесь P- некоторое условие (отношение), S- команда или серия команд.

Особенностью данной конструкции является та ситуация, когда тело цикла (в данном случае S – и есть тело цикла) может не выполниться ни разу (например при первой проверке условие сразу имеет значение ложь)

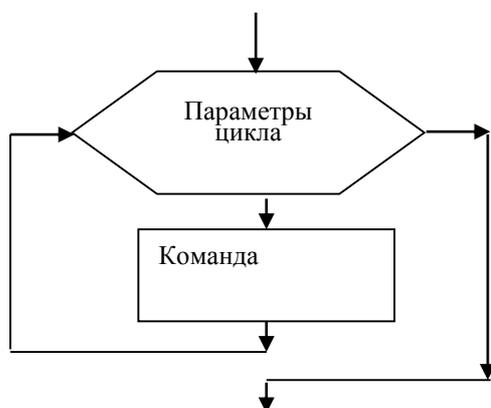
Цикл с постусловием имеет вид:



В таком цикле серия команд (тело цикла) всегда выполнится хотя бы один раз.

Тело цикла с предусловием, как и с постусловием, обязательно должно содержать команду изменения переменной, проверяемой в условии. Если такая команда в теле цикла отсутствует, то происходит заикливание алгоритма, потому что отсутствует возможность выхода из цикла.

Цикл с параметром (счетчиком)



Параметр цикла – управляющая переменная, которая изменяется автоматически от начального значения до конечного с заданным шагом (по умолчанию шаг=1). Этот цикл обычно используют когда известно точное количество повторений.

Рассмотрим использование различных приемов организации циклов на примерах.

Задача № 1: Студент в первый день выучил 5 английских слов. В каждый следующий день он выучивал на 2 слова больше, чем в предыдущий. Сколько английских слов выучит ученик в 10-ый день занятий? Составьте блок-схему

определения количества выученных слов. Представим решение этой задачи разными циклическими конструкциями.

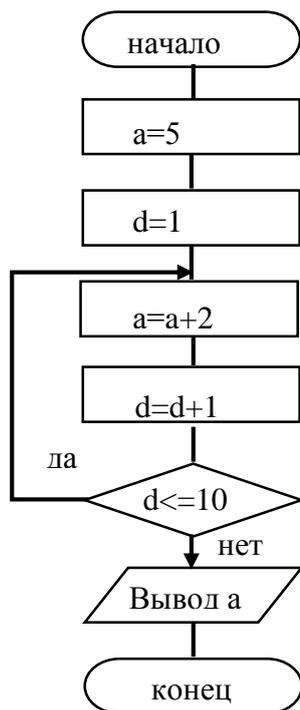
Постановка задачи:

a - количество выученных слов за один день. Первоначальное значение $a=5$
 d – счетчик дней.

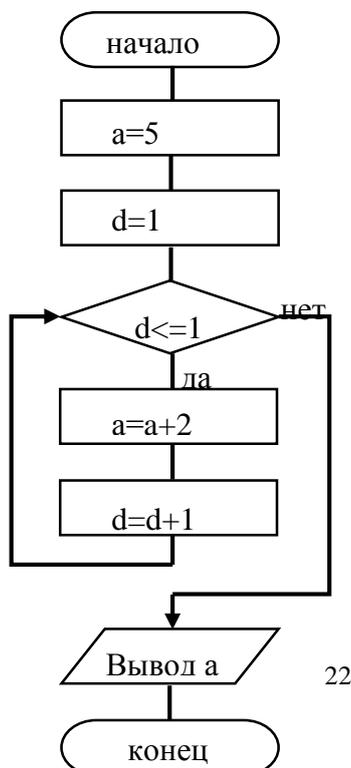
Математическая модель:

1. $a=5$ – выучил в первый день
2. $d=1$ – начальное значение счетчика дней
3. если $d \leq 10$, то $a = a + 2$ и $d = d + 1$, возврат на проверку d , иначе вывод a

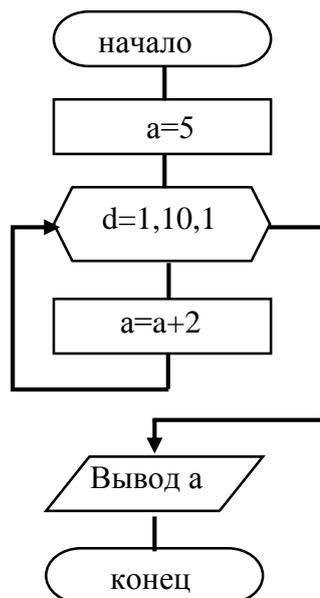
Блок схема алгоритма с предусловием имеет вид:



Блок схема алгоритма с постусловием имеет вид



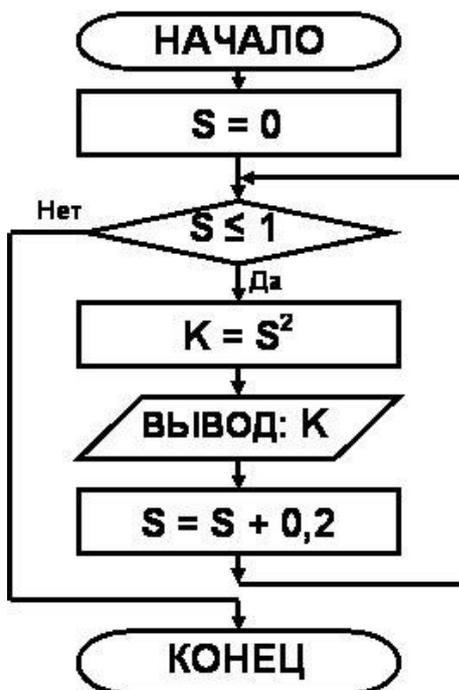
Цикл с параметром:



Задача № 2: Составить алгоритм в виде блок-схемы, в результате выполнения которого вычисляются и выводятся квадраты всех чисел, находящихся в диапазоне от 0 до 1, изменение числа выполняется с шагом 0,2. Выполнить алгоритм в виде цикла с предусловием.

Постановка задачи: S- текущее число, начальное значение S=0, шаг изменения =0,2.

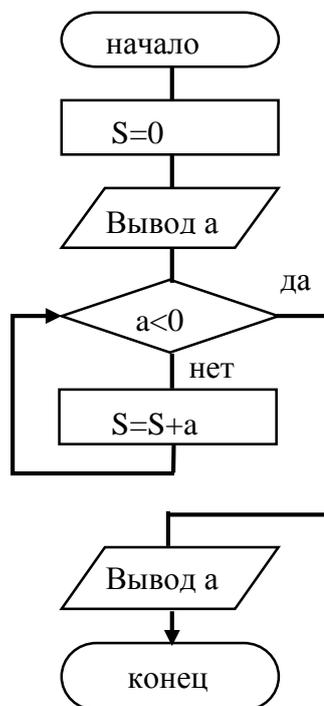
K- значение, равное квадрату числа S



Задача № 3: Рассмотрим алгоритм, в котором требуется вводить поочередно числа и подсчитывать их сумму, до первого введенного отрицательного числа. Составить блок-схему такого алгоритма.

Постановка задачи: S - значение суммы неотрицательных чисел, нетрудно догадаться что первое же введенное число может оказаться отрицательным, поэтому начальное значение $S=0$;

a -число вводимое во время исполнения алгоритма



Задача № 4: Составить алгоритм планирования закупки товара в магазине на сумму, не превышающую заданной величины.

Постановка задачи:

Исходные данные: c – цена товара;

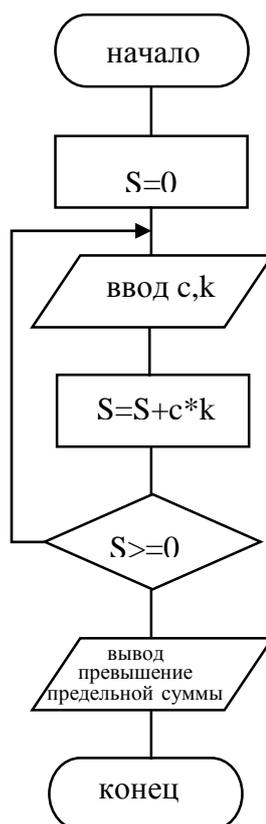
k – количество товара;

p – заданная предельная величина расходов;

s – начальное значение общей стоимости покупки;

Начальное значение $s=0$, значение предельной суммы p – вводится с клавиатуры. Необходимо повторять запрос цены и количества выбранного товара, вычислять его стоимость и выводить результат на экран до тех пор, пока она не превысит предельную сумму, в этом случае на экран нужно вывести сообщение о превышении.

Составить блок-схему для задачи



Вложенные циклы – это такие циклы, когда телом одного цикла является ещё один или несколько новых циклов. Конструкция из вложенных циклов напоминает матрешку, в которую вложена меньшая матрешка, в меньшую ещё более меньшая и. т. д. Вложенные циклы можно выполнять с помощью любых циклических конструкций.

Задачи для самостоятельного решения

Задача №5: Составить алгоритм, в результате выполнения которого определяются и выводятся табличные значения функции $y=x^2+2$. Диапазон изменения аргумента от -4 до 4 , приращение аргумента 1

Задача № 6: Создать алгоритм определения произведения всех целых чисел A и B , если начальное значение $A=17$, шаг изменения $=-3$, начальное значение $B=-25$ шагом изменения $=5$. Вычисление проводить до тех пор пока это произведение – отрицательное число.

Задача № 7: Создать алгоритм определения количества слагаемых в сумме $1 + 1/2 + 1/3 + \dots + 1/N$ для того чтобы сумма оказалась больше 5 .

Вопросы теста проверки усвоенных знаний и умений:

Тест

Задание #1

Вопрос:

Укажите как называется свойство алгоритма, в соответствии с которым алгоритм должен быть разбит на конечное число шагов

Запишите ответ:

Задание #2

Вопрос:

Определите какое значение примет переменная y в результате выполнения алгоритма, если входные данные задачи $x=2$:

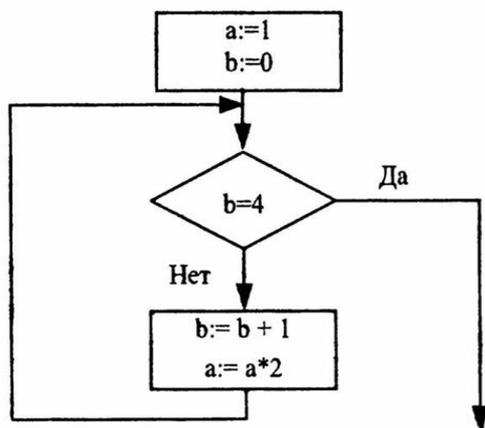
1. Задайте x
2. Сложите x и 2, результат сохраните в $R1$
3. Возведите $R1$ в квадрат, результат сохраните в $R2$
4. Разделите $R2$ на 4, результат сохраните в y
5. Получите результат y

Запишите число:

Задание #3

Вопрос:

Определите значение переменной a после выполнения фрагмента алгоритма:



Выберите один из 4 вариантов ответа:

- 1) 8
- 2) 16
- 3) 30
- 4) 2

Задание #4

Вопрос:

Укажите тип алгоритма:

1. Задайте x
2. Сложите x и 2, результат сохраните в $R1$
3. Возведите $R1$ в квадрат, результат сохраните в $R2$
4. Разделите $R2$ на 4, результат сохраните в y
5. Получите результат y

Запишите ответ:

Задание #5

Вопрос:

Вставьте пропущенное слово в формулировку:

... это запись алгоритма на одном из языков программирования

Запишите ответ:

Задание #6

Вопрос:

Установите порядок выполнения этапов при создании программы для решения её на ЭВМ

Укажите порядок следования всех 7 вариантов ответа:

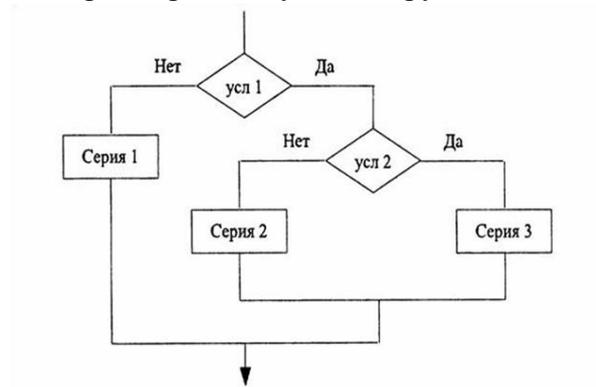
- ___ отладка программы
- ___ создание математической модели задачи
- ___ разработка алгоритма
- ___ постановка задачи
- ___ промышленная эксплуатация
- ___ опытная эксплуатация
- ___ создание программы

Задание #7

Вопрос:

Фрагмент блок-схемы представляет алгоритм, который содержит две команды ветвления.

Выберите правильную конструкцию:



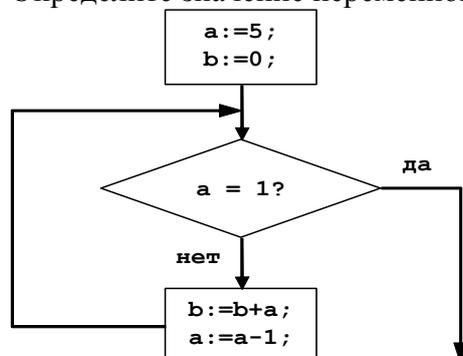
Выберите один из 4 вариантов ответа:

- 1) команду ветвления в неполной форме, в которую вложена команда ветвления в полной форме
- 2) команду ветвления в полной форме, в которую вложена команда ветвления в неполной форме
- 3) обе команды ветвления в полной форме, одна из которых вложена в другую
- 4) обе команды ветвления в неполной форме, одна из которых вложена в другую

Задание #8

Вопрос:

Определите значение переменной b после выполнения фрагмента алгоритма

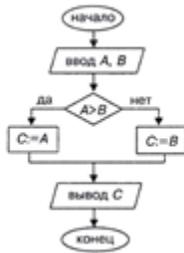


Запишите число:

Задание #9

Вопрос:

Укажите тип заданного алгоритма:



Запишите ответ:

Задание #10

Вопрос:

Определите какое значение примет переменная y в результате выполнения алгоритма, если входные данные задачи $x = -4$:

1. Задайте x
2. Сложите x и 2, результат сохраните в $R1$
3. Возведите $R1$ в квадрат, результат сохраните в $R2$
4. Разделите $R2$ на 4, результат сохраните в y
5. Получите результат y

Запишите число:

Задание #11

Вопрос:

как называется серия команд в циклическом алгоритме, выполнение которой повторяется несколько раз?

Запишите ответ:

Задание #12

Вопрос:

Процесс написания программы никогда не включает

Выберите один из 4 вариантов ответа:

- 1) изменение физических параметров компьютера
- 2) процесс отладки
- 3) редактирование текста программы
- 4) запись операторов соответствующей языку форме

Задание #13

Вопрос:

Укажите название способа задания алгоритма:

1. Задайте значение x
2. Прибавьте к x число 2, результат сохраните в $g1$
3. Возведите $g1$ в квадрат, результат сохраните в y
4. Получите результат y .

Запишите ответ:

Задание #14

Вопрос:

Укажите возможных исполнителей алгоритмов

Выберите несколько из 6 вариантов ответа:

- 1) арифметический калькулятор
- 2) человек
- 3) компьютер
- 4) луноход

- 5) собака
- 6) стиральная машина

Задание #15

Вопрос:

Как называется графическое представление алгоритма?

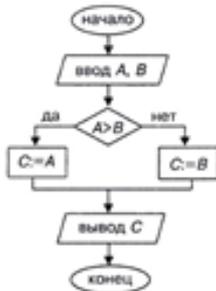
Выберите один из 4 вариантов ответа:

- 1) последовательность формул
- 2) словесное описание
- 3) таблица
- 4) блок-схема

Задание #16

Вопрос:

Укажите способ задания алгоритма:



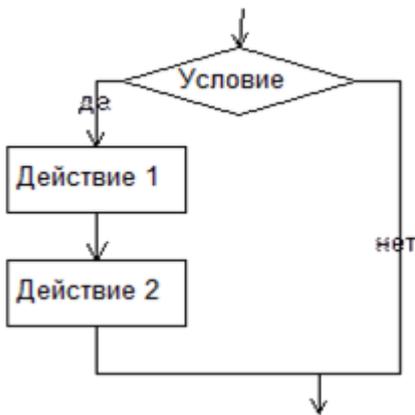
Выберите один из 3 вариантов ответа:

- 1) словесное описание
- 2) запись на псевдоалгоритмическом языке
- 3) графический способ

Задание #17

Вопрос:

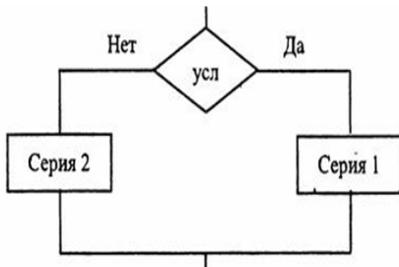
определите форму разветвляющегося алгоритма:



Запишите ответ:

Задание #18

Вопрос:



Алгоритмическая конструкция какого типа изображена на фрагменте блок-схемы?

Выберите один из 4 вариантов ответа:

- 1) линейная
- 2) разветвляющаяся
- 3) циклическая
- 4) структурная

Задание #19

Вопрос:

укажите возможные варианты циклического алгоритма

Выберите несколько из 4 вариантов ответа:

- 1) цикл с параметром
- 2) цикл с предусловием
- 3) цикл с постусловием
- 4) цикл с проверкой

Задание #20

Вопрос:

Вставьте пропущенное слово в определение:

Алгоритм - это полное и точное ... исполнителю выполнить определенную последовательность действий, направленную на решение поставленной задачи или достижение указанной цели.

Составьте слово из букв:

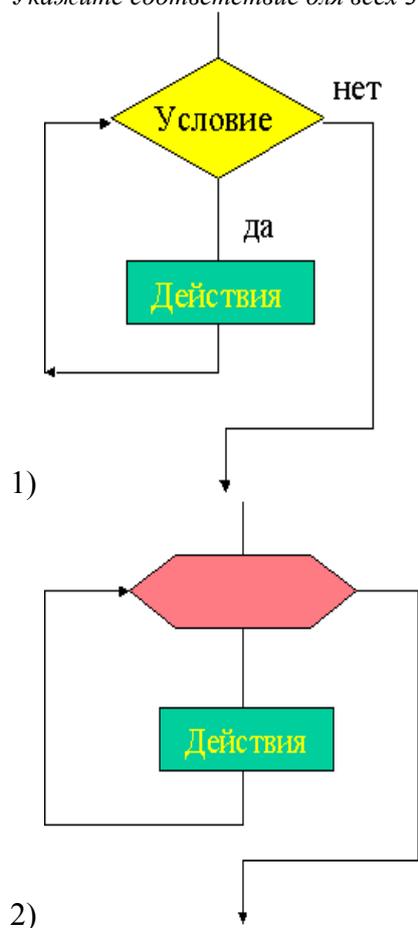
ДПШИИИНРАЕЕС -> _____

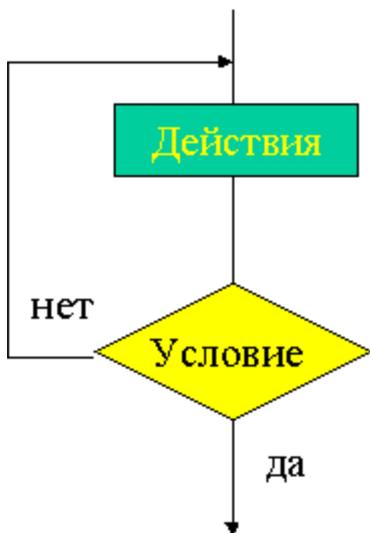
Задание #21

Вопрос:

Установите соответствие между конструкцией и её графическим изображением

Укажите соответствие для всех 3 вариантов ответа:



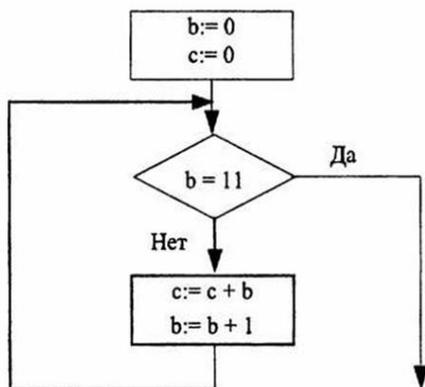


- 3)
 цикл с параметром
 цикл с постусловием
 цикл с предусловием

Задание #22

Вопрос:

Определите значение переменной c после выполнения фрагмента алгоритма:



Запишите число:

Задание #23

Вопрос:

Установите соответствие между определением алгоритма и его типом:

Укажите соответствие для всех 3 вариантов ответа:

- 1) разветвляющимся
 - 2) циклическим
 - 3) линейным
- Алгоритм, в котором для достижения результата требуется многократное повторение одних и тех же действий, называется...
- Алгоритм, в котором для достижения результата все действия выполняются строго по порядку, называется....
- Алгоритм, в котором возможны несколько путей вычислительного процесса, называется

Задание #24

Вопрос:

Деятельность, направленная на обнаружение и исправление ошибок в программной системе называется

Выберите несколько из 4 вариантов ответа:

- 1) отладкой программы
- 2) демонстрацией программы

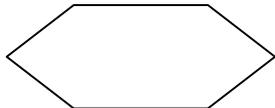
3) редактированием программы

4) тестированием программы

Задание #25

Вопрос:

Как называется конструкция блок-схемы, изображенная на рисунке ?



Выберите один из 4 вариантов ответа:

1) блок ввода-вывода значений

2) блок организации цикла

3) блок принятия решения

4) начало или конец алгоритма

Задание #26

Вопрос:

Свойство алгоритма, состоящее в том, что выполнение этого алгоритма всегда приводит к результату через конечное, возможно, очень большое число шагов называется

Составьте слово из букв:

НАОБТТУЛРИСВЗЪЕТ -> _____

Задание #27

Вопрос:

Свойство алгоритма обеспечения решения не одной задачи, а целого класса задач этого типа, называется:

Выберите один из 4 вариантов ответа:

1) массовость

2) понятность

3) дискретность

4) определенность

Задание #28

Вопрос:

На этапе постановки задачи

Выберите один из 4 вариантов ответа:

1) проверяется корректность работы программы

2) определяется состав входных и выходных данных

3) выполняется анализ физических характеристик

4) строится математическая модель решаемой задачи

Задание #29

Вопрос:

Установите соответствие между обозначениями основных блоков блок-схемы и их назначением:

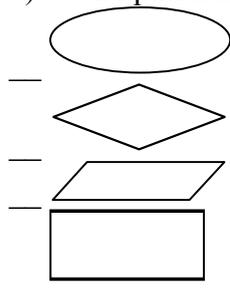
Укажите соответствие для всех 4 вариантов ответа:

1) блок выполнения действия

2) блок начала, конца алгоритма

3) блок ввода, вывода значений

4) блок принятия решения



Задание #30

Вопрос:

Как называется конструкция блок-схемы, изображенная на рисунке ?



Выберите один из 4 вариантов ответа:

- 1) начало-конец алгоритма
- 2) блок выполнения действия
- 3) блок организации цикла
- 4) ввод/вывод данных

Список рекомендуемых источников

1 Новожилов, О. П. Информатика : учебник для СПО / О. П. Новожилов. — 3-е изд., перераб. и доп. — М. : Издательство Юрайт, 2016. — 620 с. — (Профессиональное образование).

2 Гаврилов, М. В. Информатика и информационные технологии : учебник для СПО / М. В. Гаврилов, В. А. Климов. — 4-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 383 с. — (Профессиональное образование).

3 Зимин, В. П. Информатика. Лабораторный практикум в 2 ч. Часть 1 : учебное пособие для СПО / В. П. Зимин. — М. : Издательство Юрайт, 2017. — 110 с. — (Профессиональное образование).

4 Зимин, В. П. Информатика. Лабораторный практикум в 2 ч. Часть 2 : учебное пособие для СПО / В. П. Зимин. — М. : Издательство Юрайт, 2017. — 145 с. — (Профессиональное образование).

5 Информационные технологии: Сети и телекоммуникации : учебник и практикум для академического бакалавриата / К. Е. Самуйлов [и др.] ; под ред. К. Е. Самуйлова, И. А. Шалимова, Д. С. Кулябова. — М. : Издательство Юрайт, 2017. — 363 с. — (Бакалавр. Академический курс).

6 Сидорова, А. А. Электронное правительство : учебник и практикум для бакалавриата и магистратуры / А. А. Сидорова. — М. : Издательство Юрайт, 2017. — 165 с. — (Бакалавр и магистр. Академический курс).

7 Черпаков, И. В. Основы программирования : учебник и практикум для СПО / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 219 с. — (Профессиональное образование).

8 Черпаков, И. В. Теоретические основы информатики : учебник и практикум для академического бакалавриата / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 353 с. — (Бакалавр. Академический курс).