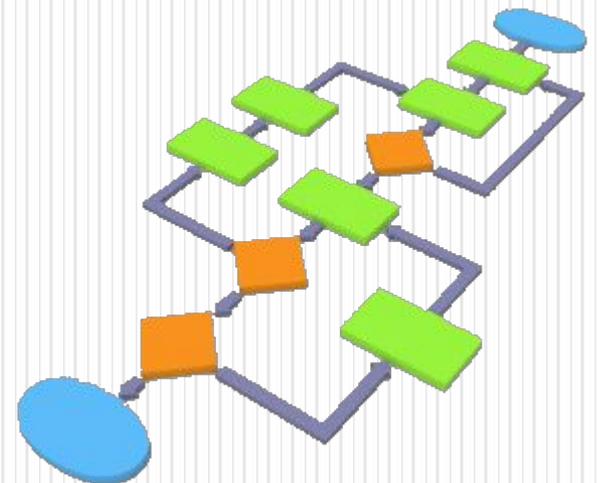


Лекция 1

Введение в алгоритмизацию



Вопросы

- Введение в алгоритмизацию
- Свойства алгоритмов
- Примеры алгоритмов
- Способы записи алгоритмов
- Структуры алгоритма (блок-схемы)
- Типы алгоритмов решения задачи
- Примеры решения задач
- Контрольные вопросы
- Список литературы

Введение в алгоритмизацию

Алгоритм - последовательность чётко определенных действий, выполнение которых ведёт к решению задачи.

Алгоритм, записанный на языке машины, есть программа решения задачи.

Алгоритм - это совокупность действий, приводящих к достижению результата за конечное число шагов.



Свойства алгоритмов

- Дискретность – это разбиение алгоритма на ряд отдельных законченных действий (шагов).
- Детерминированность - любое действие алгоритма должно быть строго и недвусмысленно определено в каждом случае.
- Конечность - каждое действие в отдельности и алгоритм в целом должны иметь возможность завершения.
- Массовость - один и тот же алгоритм можно использовать с разными исходными данными.
- Результативность - алгоритм должен приводить к достоверному решению.



Примеры алгоритмов

- Любой прибор, купленный в магазине, снабжается инструкцией по его использованию. Данная инструкция и является алгоритмом для правильной эксплуатации прибора.
- Каждый шофер должен знать правила дорожного движения. Правила дорожного движения однозначно регламентируют поведение каждого участника движения. Зная эти правила, шофер должен действовать по определенному алгоритму.
- Массовый выпуск автомобилей стал возможен только тогда, когда был придуман порядок сборки машины на конвейере. Определенный порядок сборки автомобилей – это набор действий, в результате которых получается автомобиль.



Способы записи алгоритмов

Существует несколько способов записи алгоритмов. На практике наиболее распространены следующие формы представления алгоритмов:

- **словесная** (запись на естественном языке);
- **псевдокоды** (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);
- **графическая** (изображения из графических символов - блок-схема);
- **программная** (тексты на языках программирования - код программы).

Рассмотрим подробно каждый вариант записи алгоритмов на примере следующей задачи.

Требуется найти частное двух чисел.



Словесный способ записи алгоритмов

Способ представляет собой описание последовательных этапов обработки данных. Алгоритм задается в произвольном изложении на естественном языке. Ответ при этом получает человек, который выполняет команды согласно словесной записи.

Пример словесной записи:

1. задать два числа, являющиеся делимым и делителем;
2. проверить, равняется ли делитель нулю;
3. если делитель не равен нулю, то найти частное, записать его в ответ;
4. если делитель равен нулю, то в ответ записать "нет решения".



Псевдокод

Псевдокод занимает промежуточное место между *естественным* и *формальным* языками. С одной стороны, он близок к обычному естественному языку, поэтому алгоритмы могут на нем записываться и читаться как обычный текст. С другой стороны, в псевдокоде используются некоторые формальные конструкции и математическая символика, что приближает запись алгоритма к общепринятой математической записи.



Основные управляющие структуры псевдокода

Название структуры	Псевдокод
Присваивание	переменная = число
Ввод	ввод (переменная)
Вывод	вывод (переменная) вывод ("фраза")
Ветвление	если условие то действие 1 иначе действие 2
Повторение	пока условие начало пока действие конец пока



Пример псевдокода

- алг Нахождение частного двух чисел
- начало
- вывод ("задайте делимое и делитель")
- ввод (делимое, делитель)
- если делитель $\neq 0$
- то частное = делимое / делитель
- вывод(частное)
- иначе вывод("нет решения")
- кон алг Нахождение частного двух чисел



Программная реализация алгоритма

Программная реализация алгоритма - это компьютерная программа, написанная на каком-либо алгоритмическом языке программирования, например: C++, Pascal, Basic и т.д.

Программа состоит из команд определенного языка программирования. Отметим, что одна и та же схема алгоритма может быть реализована на разных языках программирования. Ответ при этом получает ЭВМ, а не человек.



Графическая реализация алгоритма

Графическая реализация алгоритма представляет собой схему алгоритма. Схема алгоритма состоит из блоков определенной формы, соединенных стрелками. Ответ при этом получает человек, который выполняет команды согласно блок-схеме.

Схема алгоритма - это графическая реализация алгоритма.

Схема алгоритма представляет собой удобный и наглядный способ записи алгоритма.

Схема алгоритма состоит из функциональных блоков разной формы, связанных между собой стрелками.



Структуры алгоритма

Форма структур	Назначение структуры
	Начало и конец блок-схемы
	Блок ввода данных
	Блок выполнения действия
	Блок условия
	Блок вывода данных

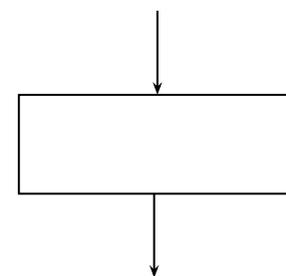
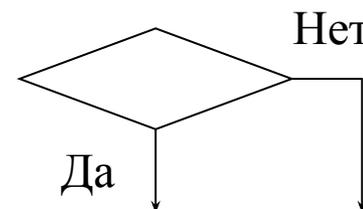
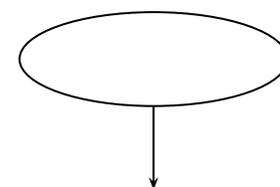


Любая команда алгоритма записывается в схеме алгоритма в виде графического элемента - блока, и дополняется словесным описанием.

Блоки в схемах алгоритма соединяются линиями потока информации. Направление потока информации указывается стрелкой. В случае потока информации сверху вниз и слева направо стрелку ставить не обязательно. Блоки в схемах алгоритма имеют только один вход и один выход (за исключением логического блока – блока с условием).



- Блок начала в схеме алгоритма имеет один выход и не имеет входов.
- Блок конца схемы алгоритма имеет один вход и не имеет выходов.
- Блок условия - единственный блок, имеющий два выхода, т.к. соответствует разветвляющемуся алгоритму. На одном выходе указывается «да», на другом – «нет».
- Все остальные блоки имеют один вход и один выход.
- Блок выполнения действия может содержать присвоение значения переменной (« $X = 5$ ») или вычисление (« $y = x - 4$ »).



Алгоритмы решения задач

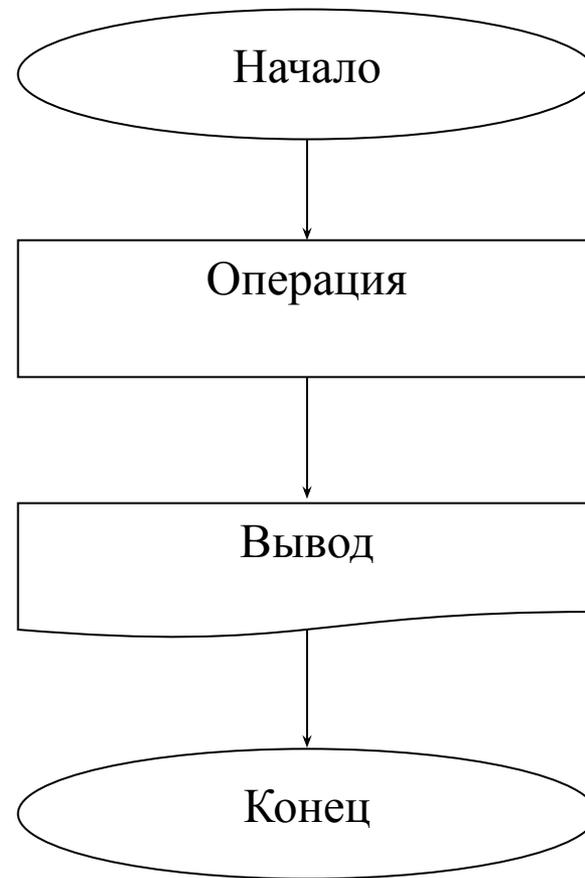
Различают три основных вида алгоритмов:

1. Линейный алгоритм - это алгоритм, в котором действия выполняются однократно и строго последовательно.
2. Разветвляющийся алгоритм - это алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий.
3. Циклический алгоритм - это алгоритм, команды которого повторяются некое количество раз подряд.



Линейный алгоритм

Линейный алгоритм - это алгоритм, в котором действия выполняются однократно и строго последовательно.



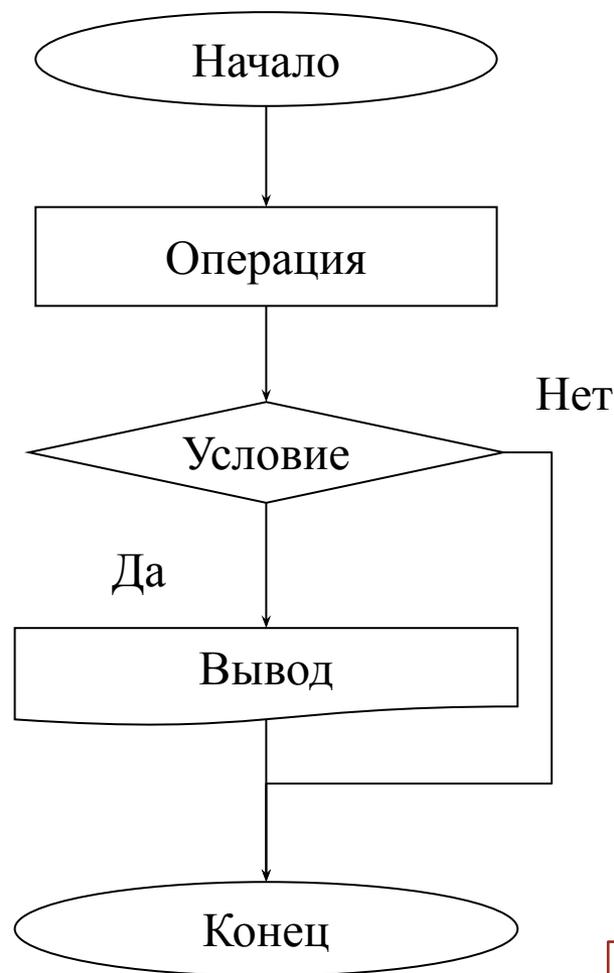
Разветвляющийся алгоритм

Разветвляющийся алгоритм - это алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий.

Внутри блока условия записывается условие.

Если данное условие верно, то выполняются блоки, идущие по стрелке «да». Если условие оказывается неверным, ложным, то выполняются блоки, идущие по стрелке «нет».

Разветвление заканчивается, когда обе стрелки («да» и «нет») соединяются.



Циклический алгоритм

Циклический алгоритм - это алгоритм, команды которого повторяются некоторое количество раз подряд.

- Тело цикла - это набор инструкций, предназначенный для многократного выполнения.
- Итерация - это единичное выполнение тела цикла.
- Переменная цикла — это величина, изменяющаяся на каждой итерации цикла.

Каждый цикл должен содержать следующие необходимые элементы:

- первоначальное задание переменной цикла,
- проверку условия,
- выполнение тела цикла,
- изменение переменной цикла.



Циклический алгоритм

Циклы бывают двух видов – с предусловием и с постусловием.

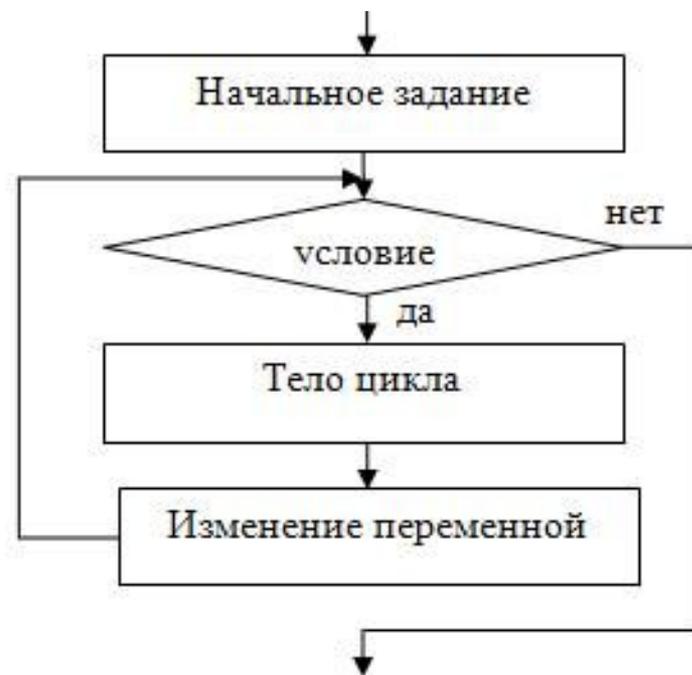
В цикле с **предусловием** сначала проверяется условие входа в цикл, а затем выполняется тело цикла, если условие верно. Цикл с предусловием также может быть задан с помощью счетчика. Это удобно в тех случаях, когда точно известно количество итераций.



Цикл с предусловием

Сначала задается начальное значение переменной цикла, затем условие входа в цикл, тело цикла и изменение переменной цикла. Выход из цикла осуществляется в момент проверки условия входа в цикл, когда оно не выполняется, т.е. условие ложно.

Цикл с предусловием может ни разу не выполниться, если при первой проверке условия входа в цикл оно оказывается ложным.

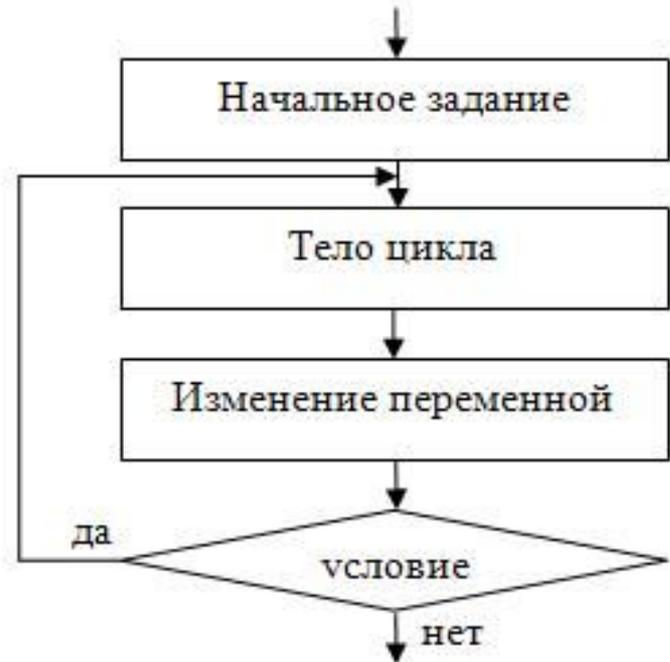


Циклический алгоритм с предусловием в общем виде



Цикл с постусловием

В цикле с постусловием сначала выполняется тело цикла, а потом проверяется условие. Если условие верно, то итерация повторяется, если же неверно, то осуществляется выход из цикла. В отличие от цикла с предусловием, любой цикл с постусловием всегда выполнится хотя бы раз.

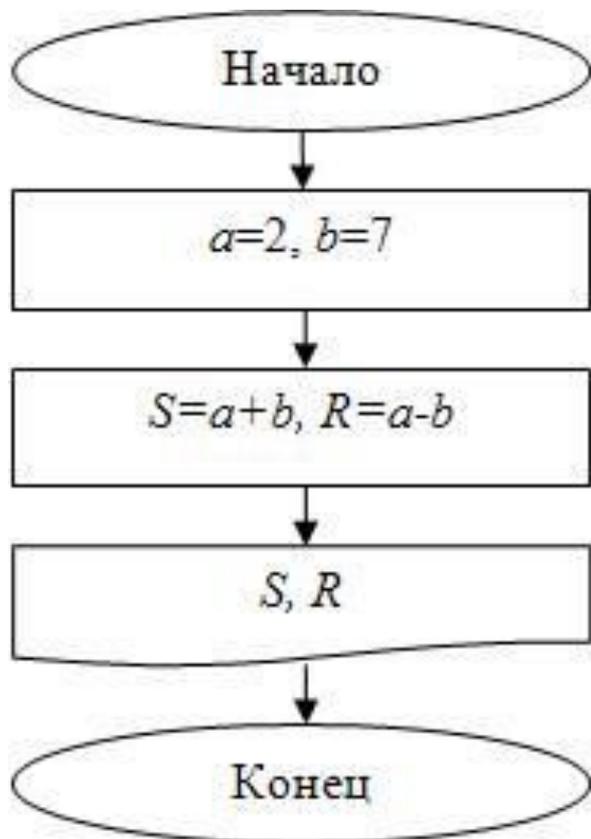


Циклический алгоритм с постусловием в общем виде



Пример 1. Линейный алгоритм.

Даны числа $a = 2$, $b = 7$. Вычислить сумму S и разность R чисел a и b .



Расчеты:

Начало

$$A = 2 \quad B = 7$$

$$S = a + b \text{ \{подставляем значения } a \text{ и } b\} = 2 + 7 = 9$$

$$R = a - b \text{ \{подставляем значения } a \text{ и } b\} = 2 - 7 = -5$$

Выводим на экран $S = 9$, $R = -5$:

Вывод данных (экран)

$$S = 9$$

$$R = -5$$

Конец.



Пример 2. Разветвляющийся алгоритм.

Даны числа $a = 2$, $b = 7$. Вычислить сумму S и разность R чисел a и b . Сравнить полученные значения S и R и указать большее из них.

Расчеты:

Начало

$$A = 2 \quad B = 7$$

$$S = a + b = 2 + 7 = 9$$

$$R = a - b = 2 - 7 = -5$$

Выводим на экран $S = 9$, $R = -5$:

Вывод данных (экран)

$$S = 9 \quad R = -5$$

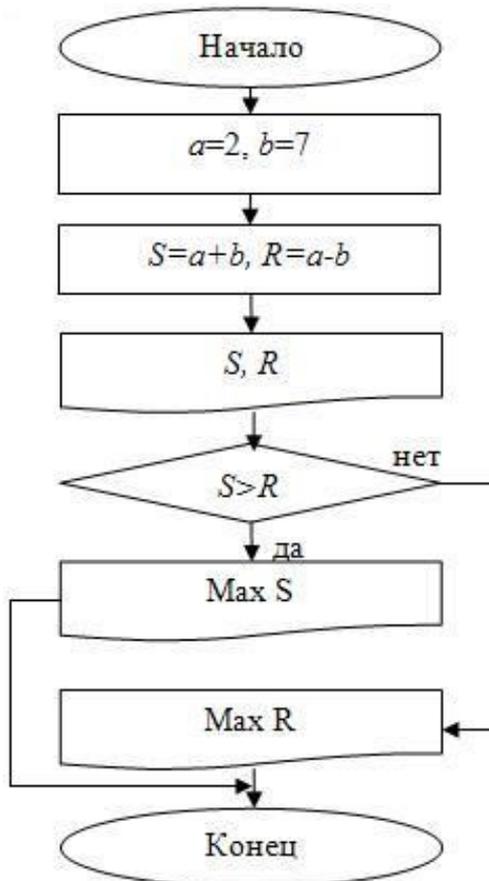
$S > R$ $9 > -5$ да, верно

Выводим на экран «Max S»

Вывод данных (экран)

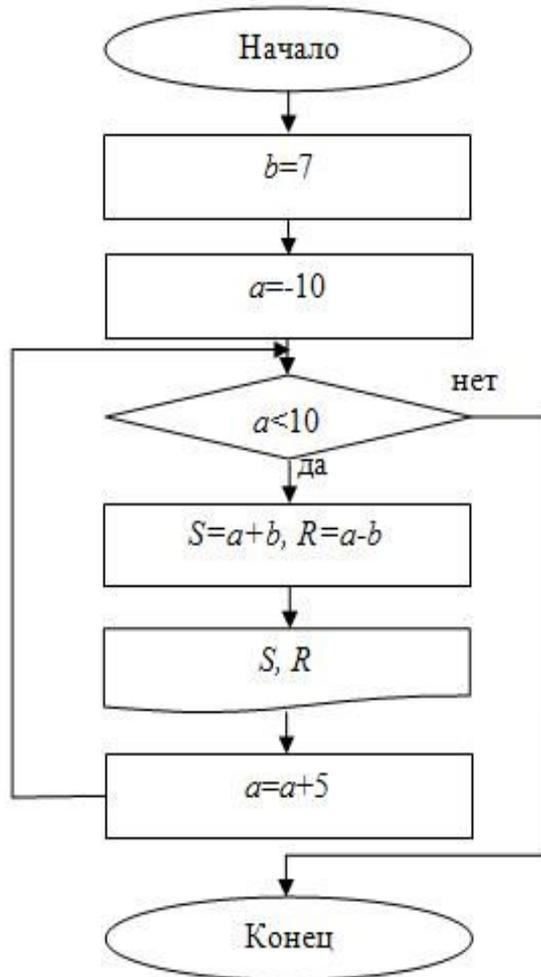
$$S = 9 \quad R = -5 \quad \text{Max } S$$

Конец.



Пример 3. Циклический алгоритм.

Даны числа a, b . Известно, что число a меняется от -10 до 10 с шагом 5 , $b = 7$ и не изменяется. Вычислить сумму S и разность R чисел a и b для всех значений a и b .



Расчеты:

Начало

$$b = 7 \quad a = -10$$

$$a \leq 10 \quad -10 \leq 10 \quad \text{да, верно}$$

$$S = a + b = -10 + 7 = -3$$

$$R = a - b = -10 - 7 = -17$$

Вывод S, R

Экран:

$$S = -3 \quad R = -17$$

$$a = a + 5 = -10 + 5 = -5$$

{Идем по стрелке вверх}

$$a \leq 10 \quad -5 \leq 10 \quad \text{да, верно}$$

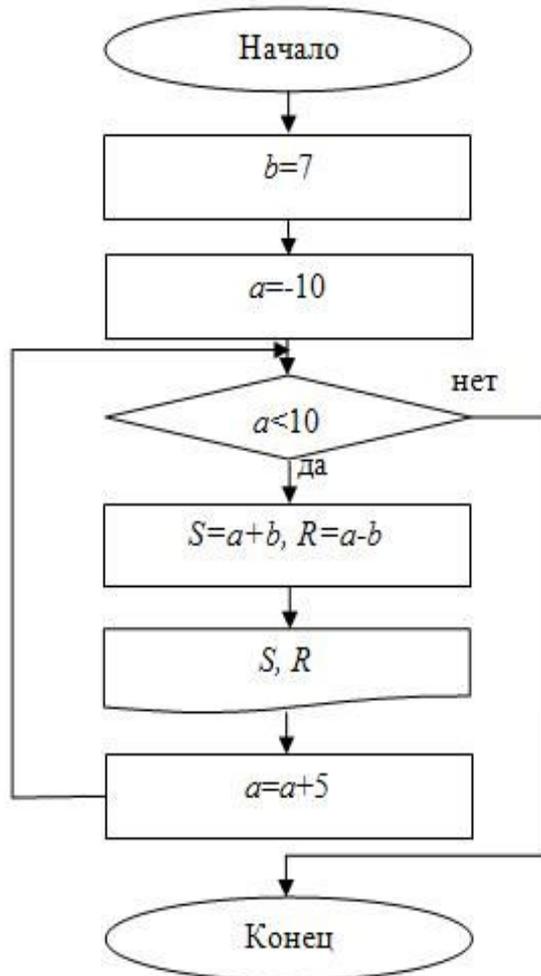
$$S = a + b = -5 + 7 = 2$$

$$R = a - b = -5 - 7 = -12$$

Вывод S, R



Пример 3. Часть 2



Экран:

$S=-3$ $R=-17$

$S=2$ $R=-12$

$a=a+5=-5+5=0$

{Идем по стрелке вверх}

$a \leq 10$ $0 \leq 10$ да, верно

$S=a+b=0+7=7$

$R=a-b=0-7=-7$

Вывод S, R

Экран:

$S=-3$ $R=-17$

$S=2$ $R=-12$

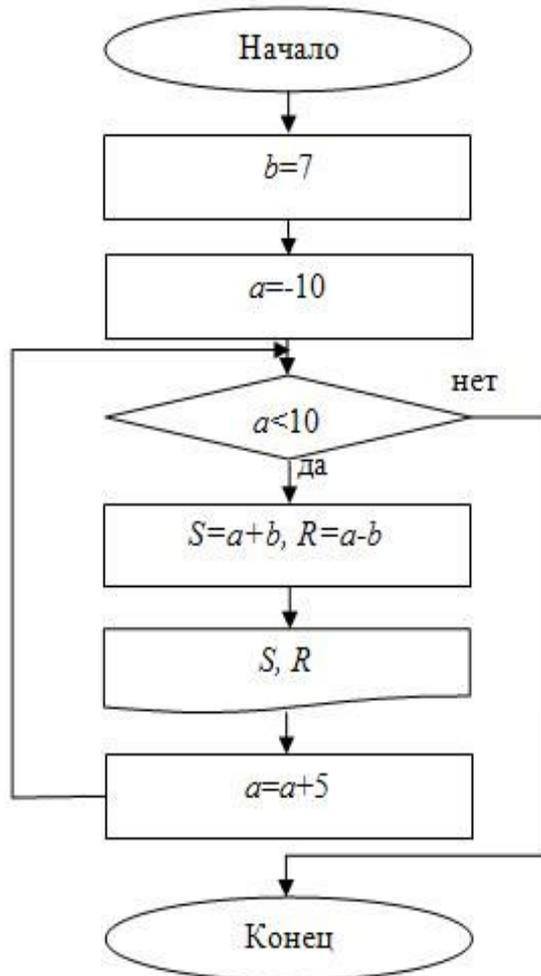
$S=7$ $R=-7$

$a=a+5=0+5=5$

{Идем по стрелке вверх}



Пример 3. Часть 3



$a \leq 10$ $5 \leq 10$ да, верно

$$S = a + b = 5 + 7 = 12$$

$$R = a - b = 5 - 7 = -2$$

Вывод S, R

Экран:

$$S = -3 \quad R = -17$$

$$S = 2 \quad R = -12$$

$$S = 7 \quad R = -7$$

$$S = 12 \quad R = -2$$

$$a = a + 5 = 5 + 5 = 10$$

{Идем по стрелке вверх}

$a \leq 10$ $10 \leq 10$ да, верно

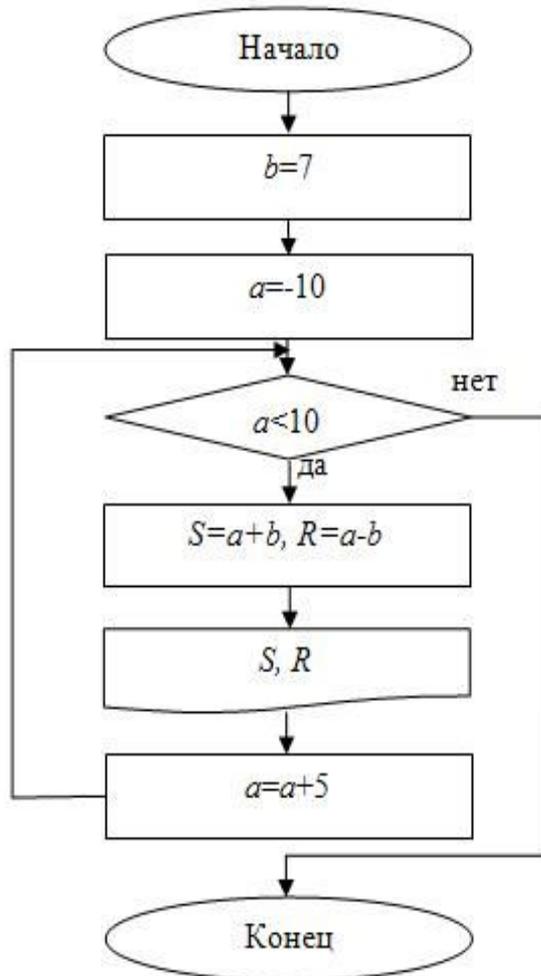
$$S = a + b = 10 + 7 = 17$$

$$R = a - b = 10 - 7 = 3$$

Вывод S, R



Пример 3. Часть 4



Экран:

S=-3 R=-17

S=2 R=-12

S=7 R=-7

S=12 R=-2

S=17 R=3

$a = a + 5 = 10 + 5 = 15$

{Идем по стрелке вверх}

$a \leq 10$ $15 \leq 10$ нет, ложно

{ВЫХОДИМ ИЗ ЦИКЛА}

Конец.



Контрольные вопросы для закрепления изученного материала

1. Что такое алгоритм?
2. В чем состоит задача алгоритмизации?
3. Какими свойствами обладает алгоритм?
4. Какие виды алгоритма бывают?
5. Чем отличается цикл с постусловием и предусловием?
6. Что такое схема алгоритма?
7. Какие типы блоков бывают?



Список литературы

- Павловская Т.А. С/С++. Программирование на языке высокого уровня / Т. А. Павловская. - СПб.: Питер, 2004. - 461 с.: ил.
- Павловская Т.А. С/С ++. Структурное программирование: Практикум / Т.А. Павловская, Ю.А. Щупак. СПб.: Питер, 2007. - 239 с.: ил.
- Павловская Т. А., Щупак Ю. А. С++. Объектно-ориентированное программирование: Практикум. - СПб.: Питер, 2006. - 265 с: ил.
- Кольцов Д.М. 100 примеров на Си. - СПб.: “Наука и техника”, 2017 - 256 с.
- 5 Доусон М. Изучаем С++ через программирование игр. - СПб.: “Питер”, 2016. - 352.
- Седжвик Р. Фундаментальные алгоритмы на С++. Анализ/Структуры данных/Сортировка/Поиск: Пер. с англ. Роберт Седжвик. - К.: Издательство “Диасофт”, 2001. - 688с.
- Сиддхартха Р. Освой самостоятельно С++ за 21 день. - М.: SAMS, 2013. - 651 с.
- Стивен, П. Язык программирования С++. Лекции и упражнения, 6-е изд. Пер. с англ. - М.: ООО "И.Д. Вильямс", 2012. - 1248 с.
- Черносивов, А. Visual С++: руководство по практическому изучению / А. Черносивов . - СПб. : Питер, 2002. - 528 с. : ил.



Список литературы

- Страуструп Б. Дизайн и эволюция языка С++. - М.: ДМК, 2000. - 448 с.
- Мейерс С. Эффективное использование С++. - М.: ДМК, 2000. - 240 с.
- Бадд Т. Объектно-ориентированное программирование в действии. - СПб: Питер, 1997. - 464 с.
- Лаптев В.В. С ++. Объектно-ориентированное программирование: Учебное пособие.- СПб.: Питер, 2008. - 464 с.: ил.
- Страуструп Б. Язык программирования С++. Режим доступа: http://8361.ru/6sem/books/Straustrup-Yazyk_programmirovaniya_c.pdf.
- Керниган Б., Ритчи Д. Язык программирования Си. Режим доступа: http://cpp.com.ru/kr_cbook/index.html.
- Герберт Шилдт: С++ базовый курс. Режим доступа: https://www.bsuir.by/m/12_100229_1_98220.pdf,
- Богуславский А.А., Соколов С.М. Основы программирования на языке Си++. Режим доступа: http://www.ict.edu.ru/ft/004246/cpp_pl.pdf.
- Линский, Е. Основы С++. Режим доступа: <https://www.lektorium.tv/lecture/13373>.
- Конова Е. А., Поллак Г. А. Алгоритмы и программы. Язык С++: Учебное пособие. Режим доступа: https://vk.com/doc7608079_489807856?hash=e279524206b2efd567&dl=f85cf2703018eaa2

