Основы программирования

Самойлов Михаил Юрьевич

Общее понятие алгоритма

- Понятие алгоритма - одно из основных понятий программирования и математики.
- Алгоритм - это последовательность команд, предназначенная исполнителю, в результате выполнения которых он должен решить поставленную задачу.
- Алгоритм записывается на формальном языке, исключающем неоднозначность толкования.
- Исполнитель - это человек, компьютер, автоматическое устройство и т.п. Он должен уметь выполнять все команды, составляющие алгоритм.
- Программа - это запись алгоритма на конкретном формальном языке

Свойства алгоритмов

- Дискретность – это разбиение алгоритма на ряд отдельных законченных действий.
- 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 алгоритма

- Любой прибор, купленный в магазине, снабжается инструкцией по его использованию. Данная инструкция и является алгоритмом для правильной эксплуатации прибора.
- Массовый выпуск автомобилей. Определенный порядок сборки машины на конвейере – это набор действий, в результате которых получается автомобиль.

Способы записи алгоритма

- словесная
- псевдокоды (включает в себя как элементы языка программирования, так и фразы естественного языка);
- графическая (блок-схема);
- программная (тексты на языках программирования).

Пример словесной записи

- задать два числа, являющиеся делимым и делителем
- проверить, равняется ли делитель нулю
- если делитель не равен нулю, то найти частное, записать его в ответ
- если делитель равен нулю, то в ответ записать "нет решения"

Псевдокод

- При изложении идеи алгоритма не всегда целесообразно пользоваться каким-либо конкретным языком программирования, чтобы не загромождать изложение несущественными деталями. В таких случаях применяется неформальный алгоритмический язык, максимально приближенный к естественному. Язык такого типа называют псевдокодом.
- Для специалиста не составляет труда переписать программу с псевдокода на любой конкретный язык программирования.
- Псевдокод объединяет существенные черты множества алгоритмических языков.

Базовые элементы псевдокода

Название структуры	Псевдокод
Присваивание	переменная = число
Ввод	ввод(переменная)
Вывод	вывод(переменная) вывод("фраза")
Ветвление	если условие то действие1 иначе действие2
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 ≠ 0

    то частное = делимое / делитель

    вывод(частное)

иначе вывод("нет решения")

кон алг Нахождение частного двух чисел

Блок-схема

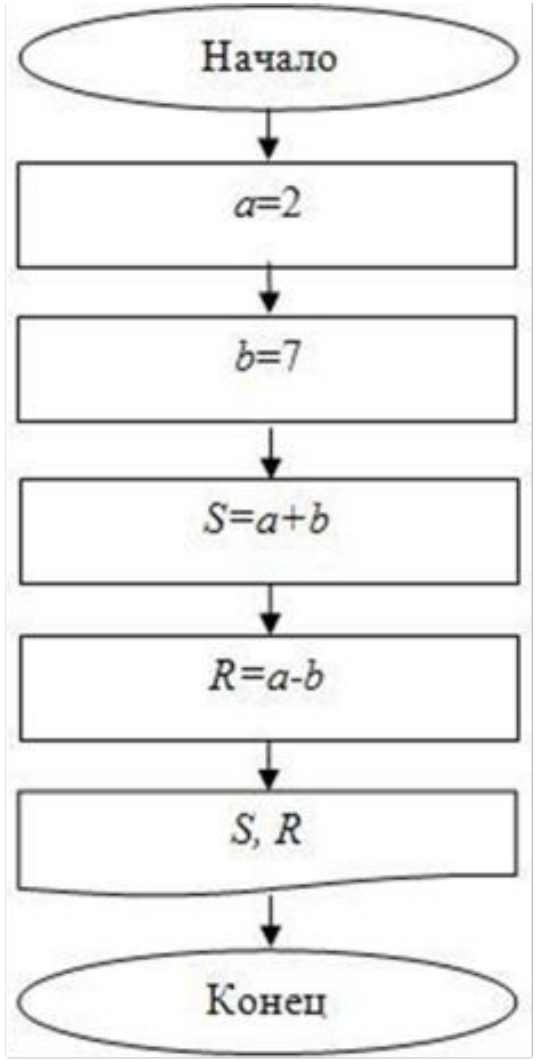
- это графическая реализация алгоритма.
- представляет собой удобный и наглядный способ записи алгоритма.
- состоит из функциональных блоков разной формы, связанных между собой стрелками. В каждом блоке описывается одно или несколько действий.

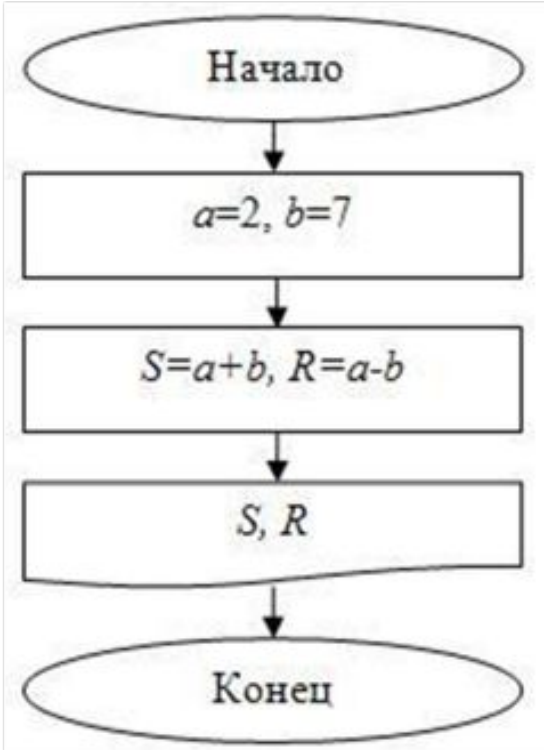
Основные виды блоков блок-схемы

Форма блока	Назначение блока
	начало и конец блок-схемы
	блок ввода данных
	блок выполнения действия
	блок условия
	блок вывода данных

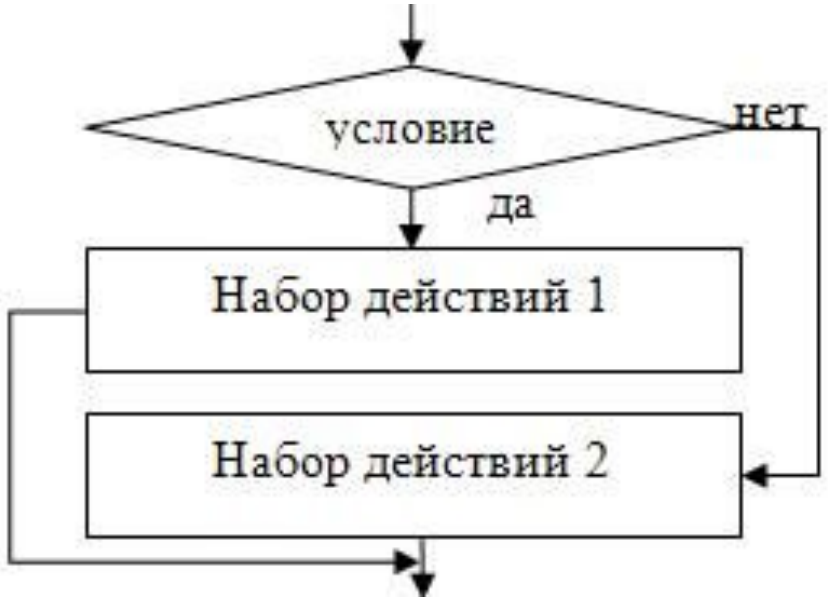
Блок-схема линейного алгоритма

Блок-схема линейного алгоритма



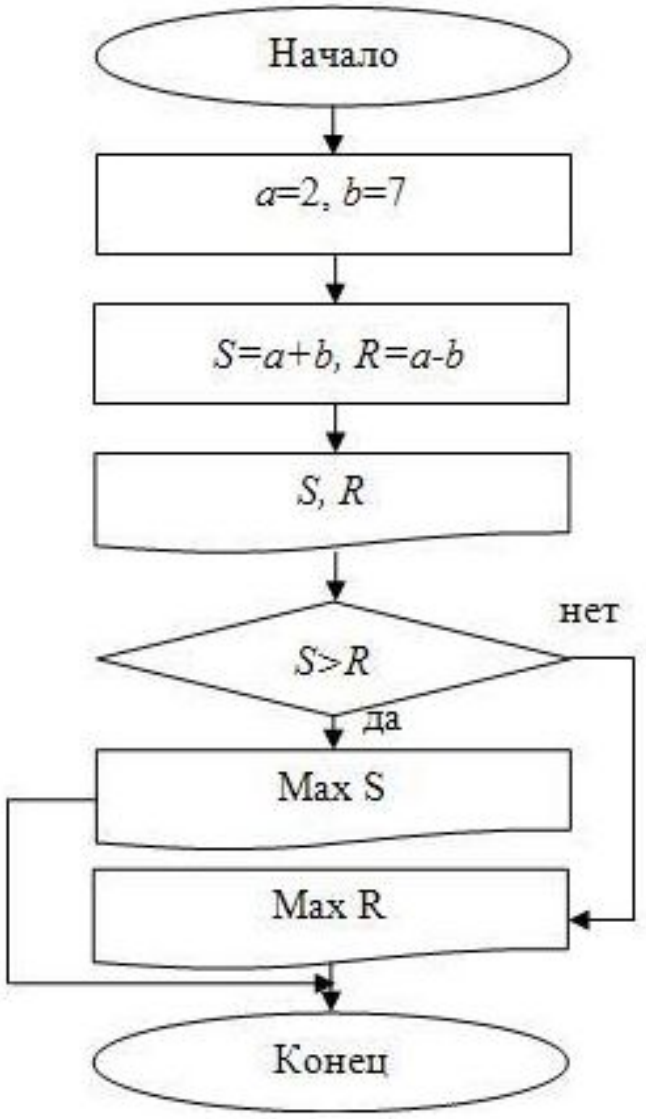


Блок условия в общем виде



Блок условия в сокращенном виде

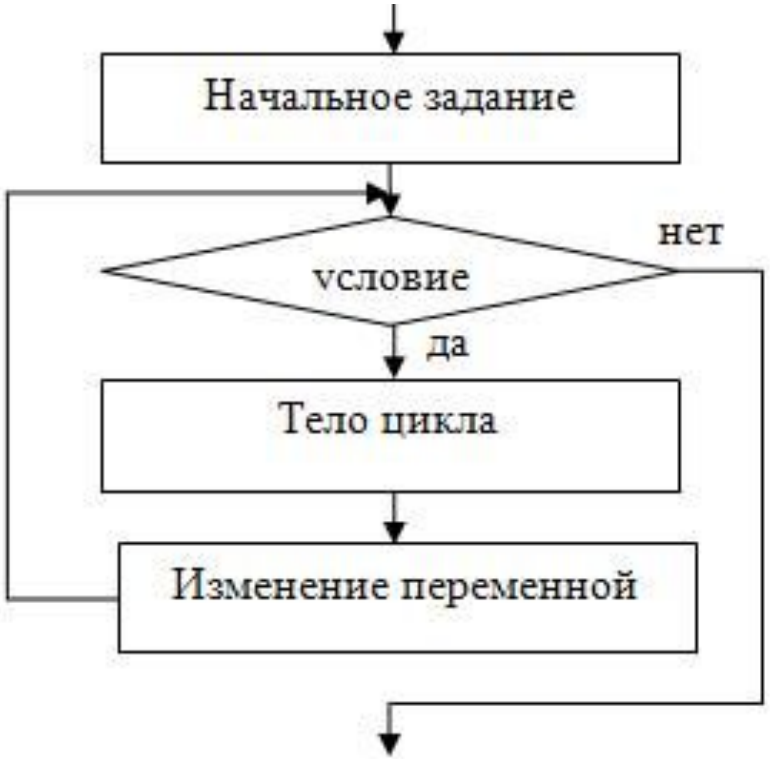
Блок-схема разветвляющегося алгоритма



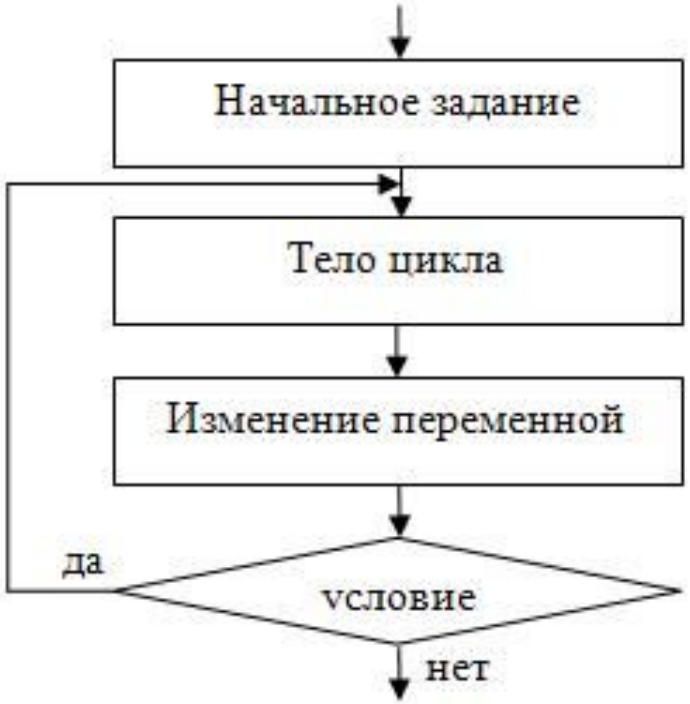
Графическая реализация циклического алгоритма

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

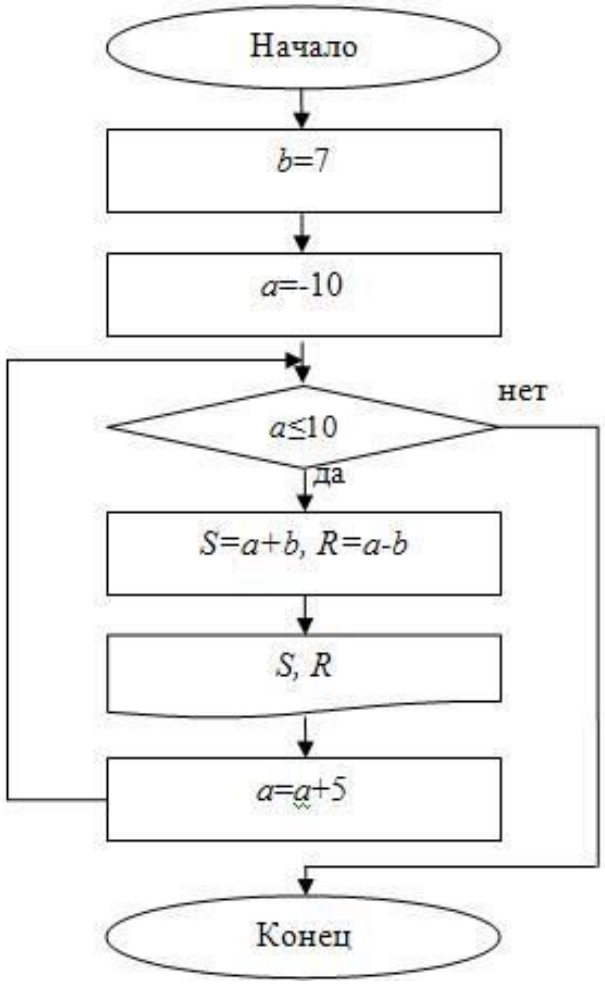
Блок цикл с предусловием

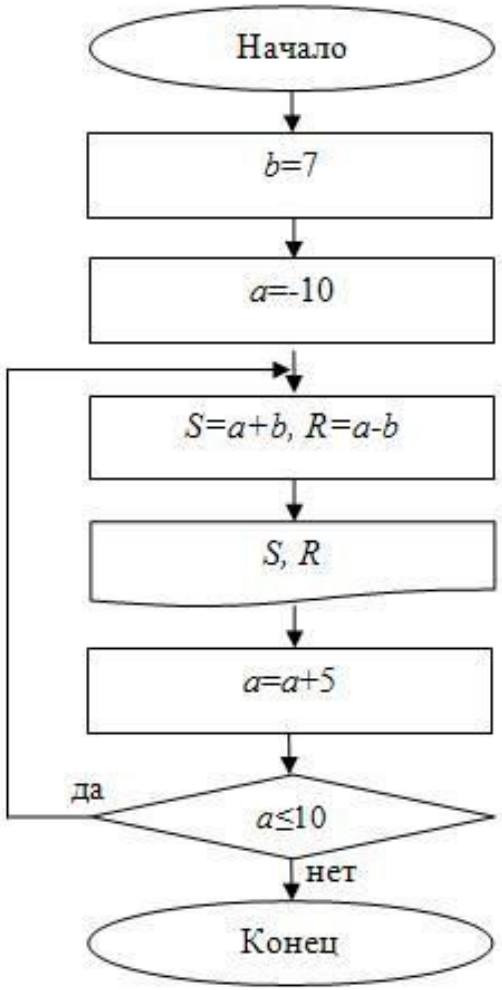


Блок цикл с постусловием



Блок-схема циклического алгоритма





Языки программирования

- Существует множество различных языков программирования.
- Все языки программирования эквивалентны друг другу и различаются лишь тем, насколько они удобны для решения конкретных задач.

Алгоритмические языки

- Программирование начиналось с записи программ непосредственно в виде машинных команд.
- Позже для облегчения кодирования был разработан язык Ассемблера, который позволяет записывать машинные команды в символическом виде.
  - удобен для программирования небольших задач
  - крупные проекты разрабатывать трудно
  - *программа* привязана к архитектуре конкретного компьютера и не может быть перенесена на другие машины.
- Языки высокого уровня.

Языки высокого уровня

- Для выполнения программы на языке высокого уровня ее нужно сначала перевести на язык машинных команд. Специальная *программа*, выполняющая такой перевод, называется **транслятором**.
- Транслятор:
  - Компилятор читает всю программу целиком, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется.
  - Интерпретатор переводит и выполняет программу строка за строкой.

C# и Visual Studio

- C# - язык программирования
- Visual Studio – среда программирования

Первая программа на языке C#

- Пойдем в меню File-New Project (Файл - Новый проект) или сразу кликнем на Create New Project.
- Выберем шаблон Console Application (Консольное приложение).
- Выберем каталог на диске, где будет расположен наш проект.

Первая программа на языке C#

```
namespace TestConsoleApplication
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

Первая программа на языке C#

```
namespace TestConsoleApplication
{
    class Program
    {
        static void Main(string[] args)
        {
            System.Console.WriteLine("Привет мир!");
            System.Console.ReadLine();
        }
    }
}
```

Первая программа на языке C#

- Нажмем F5 (можно выбрать в меню Debug - Start Debugging) и увидим черное окошко с нашим Привет мир, которое закроется после нажатия Enter.

Первая программа на языке C#

- System.Console.WriteLine("Привет мир!"); - выводит в консоль строку "Привет мир"
- System.Console.ReadLine(); - читает строку, которую мы вводим и переходит к следующей команде.

Комментарии

Однострочные комментарии

"//" Эта последовательность символов, маркирует весь последующий код в строке как комментарий.

Многострочные комментарии

При помощи них можно закомментировать любой отрезок кода. Такие комментарии начинаются с "/*" и заканчиваются "*/". Весь текст между ними, независимо от того, находится он на одной строке или нескольких, будет закомментирован.

Переменные, типы данных, константы

- **Переменная** – это именованная область памяти. В переменную можно записывать данные и считывать. Данные, записанные в переменной, называются **значением** переменной.
- **C#** – язык жесткой типизации. Каждая переменная должна быть определенного типа данных.

Целочисленный тип данных int

```
int a; // объявляем переменную a типа int
a = 5; // записываем в переменную a число 5

int b, c; // объявить можно сразу несколько переменных через запятую

int d = 10; // при объявлении переменной можно сразу же задавать ей значение, это называется инициализацией
```

Вещественный тип данных double

```
double a; // объявляем переменную a типа double
a = 5.7; // записываем в переменную a число 5

double b, c; // объявить можно сразу несколько переменных через запятую

double d = 10.3; // при объявлении переменной можно сразу же задавать ей значение, это называется инициализацией
```

Операции с числовыми типами

Операция	Запись
Сложение	a + b
Вычитание	a - b
Деление	a / b
Умножение	a * b
Нахождение остатка от деления	a % b

*При делении двух целых чисел результатом также будет целое число.

Пример

```
static void Main(string[] args)
{
	int a = 2, b = 3;
	int d = a + b;
	System.Console.WriteLine(d);
	d = a * b;
	System.Console.WriteLine(d);
}
```