

Понятие алгоритма

Лекция 13

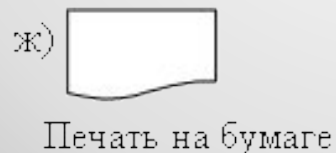
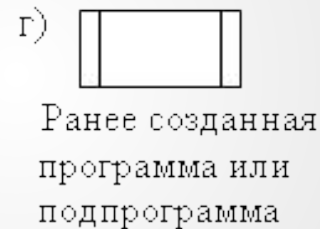
- **Алгоритм** - это четкое описание последовательности действий, которые необходимо выполнить для решения поставленной задачи
- **Программа** - это алгоритм, записанный на языке программирования
- **Языком программирования** называется специальный язык, понятный для компьютера (например, VBA)
- **Программирование** - это процесс создания, отладки и тестирования программ
- Создание любой программы начинается с разработки **алгоритма**. Именно четкое описание последовательности действий позволяет мысленно представить будущую программу. Построив алгоритм, программист мыслит четко, последовательно, однозначно - так, как и будет впоследствии мыслить компьютер

Определения

- **Дискретность** - алгоритм должен быть представлен как последовательное выполнение простых шагов
Шагом называется каждое действие алгоритма
- **Определенность** - каждое действие алгоритма должно быть четким и однозначным
- **Результативность** - алгоритм должен приводить к решению задачи за определенное число шагов
- **Массовость** - алгоритм составляется в общем виде, т.е. он должен быть применим к ряду задач, различающихся исходными данными

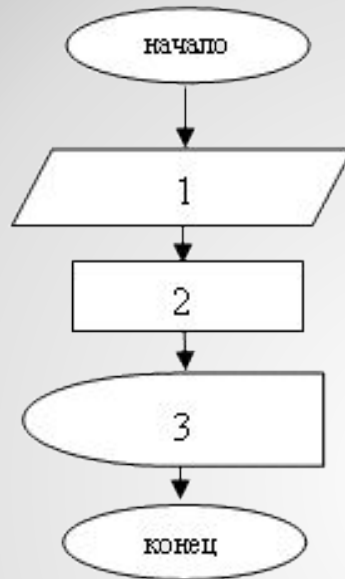
Свойства алгоритма

- **Формальный** – словесная запись алгоритма на естественном языке
- **Графический** - изображение алгоритма в виде блок-схемы
- **В блок-схеме действия алгоритма (блоки) изображаются следующими геометрическими фигурами:**



Способы записи алгоритма

Общий вид:



Пример: Кран



Блоки алгоритма 1, 2, 3 выполняются именно в такой последовательности, после чего алгоритм достигает цели и заканчивается

Блок-схемы алгоритмы

- **Постановка задачи** - составление точного и понятного словесного описания того, как должна работать будущая программа, что должен делать пользователь в процессе ее работы
- **Разработка интерфейса** (интерфейс - способ общения) - создание экранной формы (окна программы)
- **Составление алгоритма**
- **Программирование** - создание программного кода на языке программирования
- **Отладка программы** - устранение ошибок
- **Тестирование программы** - проверка правильности ее работы
- **Создание документации, помощи**

Этапы создания программы

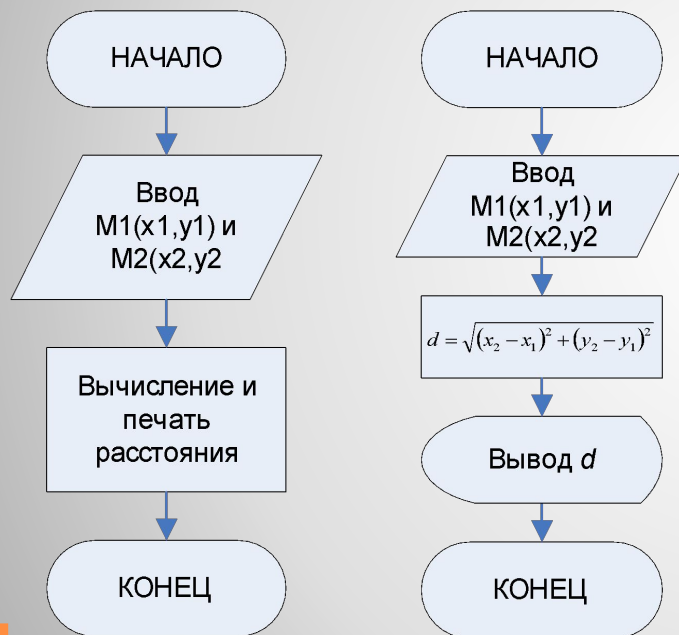
- Составьте и **опишите словесно** следующие алгоритмы:
 - Приготовление чая
 - Переход улицы по светофору
 - Окраска забора
- Опишите составленные алгоритмы **с помощью блок-схемы**

Практическое задание

- Основные типы:
 - Линейные алгоритмы
 - Разветвляющиеся алгоритмы
 - Циклические алгоритмы
- Дополнительные типы:
 - Алгоритмы работы с массивами
 - Алгоритмы работы со строками
 - Алгоритмы с использованием процедур и функций

Основные типы алгоритмов

- Линейным называется алгоритм, в котором результат получается путем однократного выполнения заданной последовательности действий при любых значениях исходных данных. Операторы программы выполняются последовательно, один за другим, в соответствии с их расположением в программе
- **Пример:** Определить расстояние на плоскости между двумя точками с заданными координатами $M1(x1, y1)$ и $M2(x2, y2)$



```

Void main()
{

```

```

    Double x1, y1, x2, y2, d;
    x1 = Double.Parse(Console.ReadLine());
    y1 = Double.Parse(Console.ReadLine());
    x2 = Double.Parse(Console.ReadLine());
    y2 = Double.Parse(Console.ReadLine());

    d = Math.Sqrt((x2 - x1)*(x2-x1) + (y2 - y1)*(y2-y1));

    Console.WriteLine("d="+d);

```

```

}

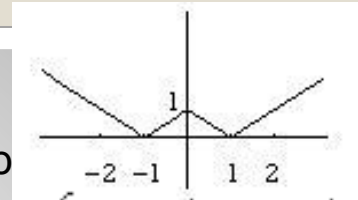
```

Программирование линейных алгоритмов

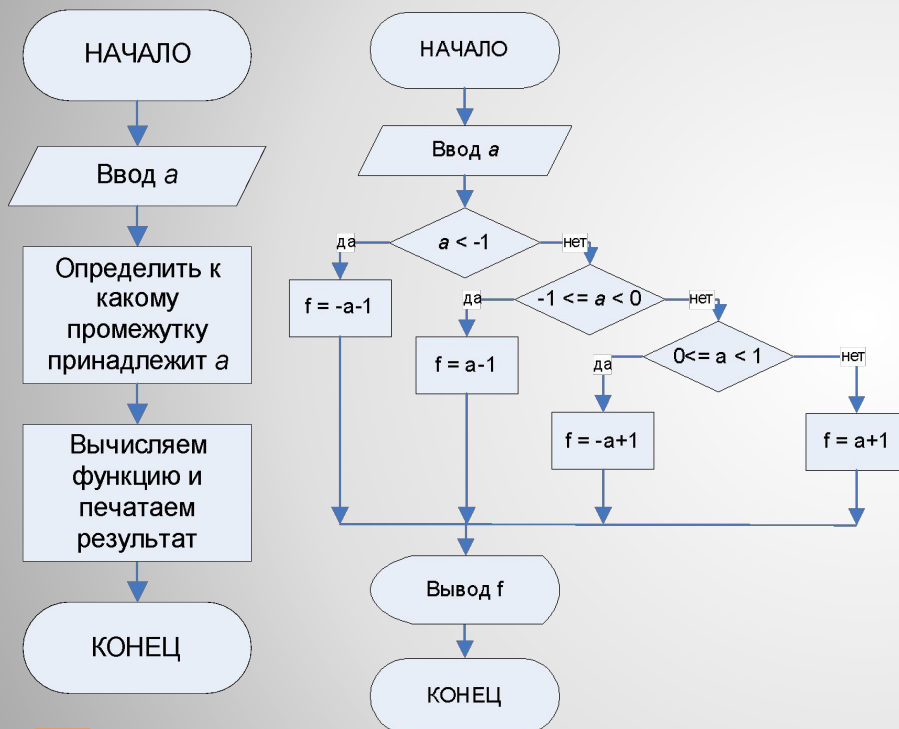
- 1) Дана длина ребра куба. Найти объем куба и площадь его боковой поверхности
- 2) Известна длина окружности. Найти площадь круга, ограниченного этой окружностью
- 3) По данным сторонам прямоугольника вычислить его периметр, площадь и длину диагонали

Задачи на использование линейных алгоритмов

- Алгоритм называется разветвляющимся, если он содержит несколько ветвей, отличающихся друг от друга содержанием вычислений. Выход вычислительного процесса на ту или иную ветвь алгоритма определяется исходными данными задачи
- Пример:** Дано действительное a . Для функций $f(a)$, график которой представлен на рисунке, вычислить $f(a)$



$$f(x) = \begin{cases} -x - 1, & x < -1 \\ x - 1, & -1 \leq x < 0 \\ -x + 1, & 0 \leq x < 1 \\ x + 1, & x \geq 1 \end{cases}$$



Void main()

{

Double a,f;

a = Double.Parse(Console.ReadLine());

If (a < -1) f = -a - 1;

Else If (a >= -1 && a < 0) f = a - 1;

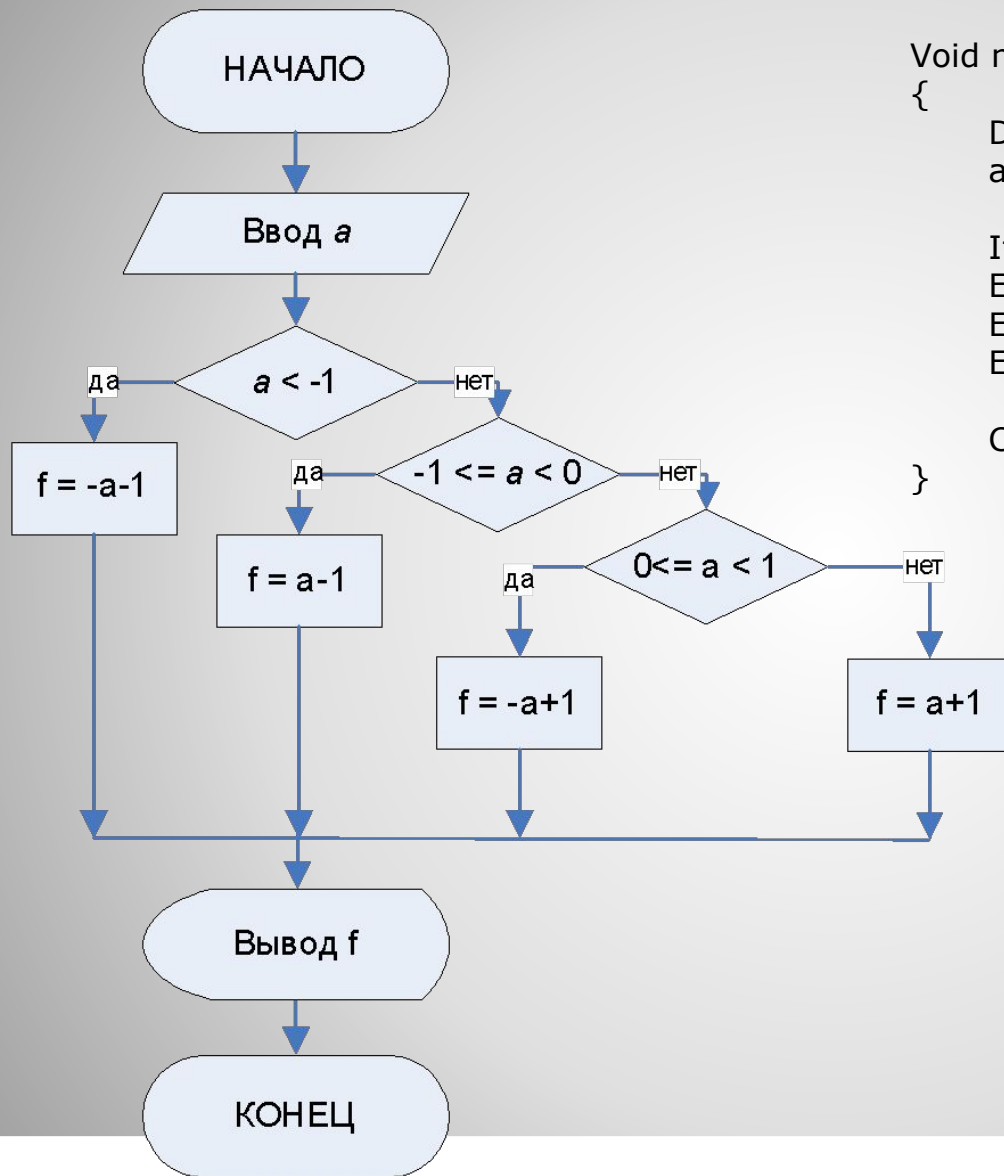
Else If (a >= 0 && a < 1) f = -a + 1;

Else f = a + 1;

Console.WriteLine("f="+f);

}

Программирование разветвляющихся алгоритмов



Void main()

{

Double a,f;

a = Double.Parse(Console.ReadLine());

If (a < -1) f = -a - 1;

Else If (a >= -1 && a < 0) f = a - 1;

Else If (a >= 0 && a < 1) f = -a + 1;

Else f = a + 1;

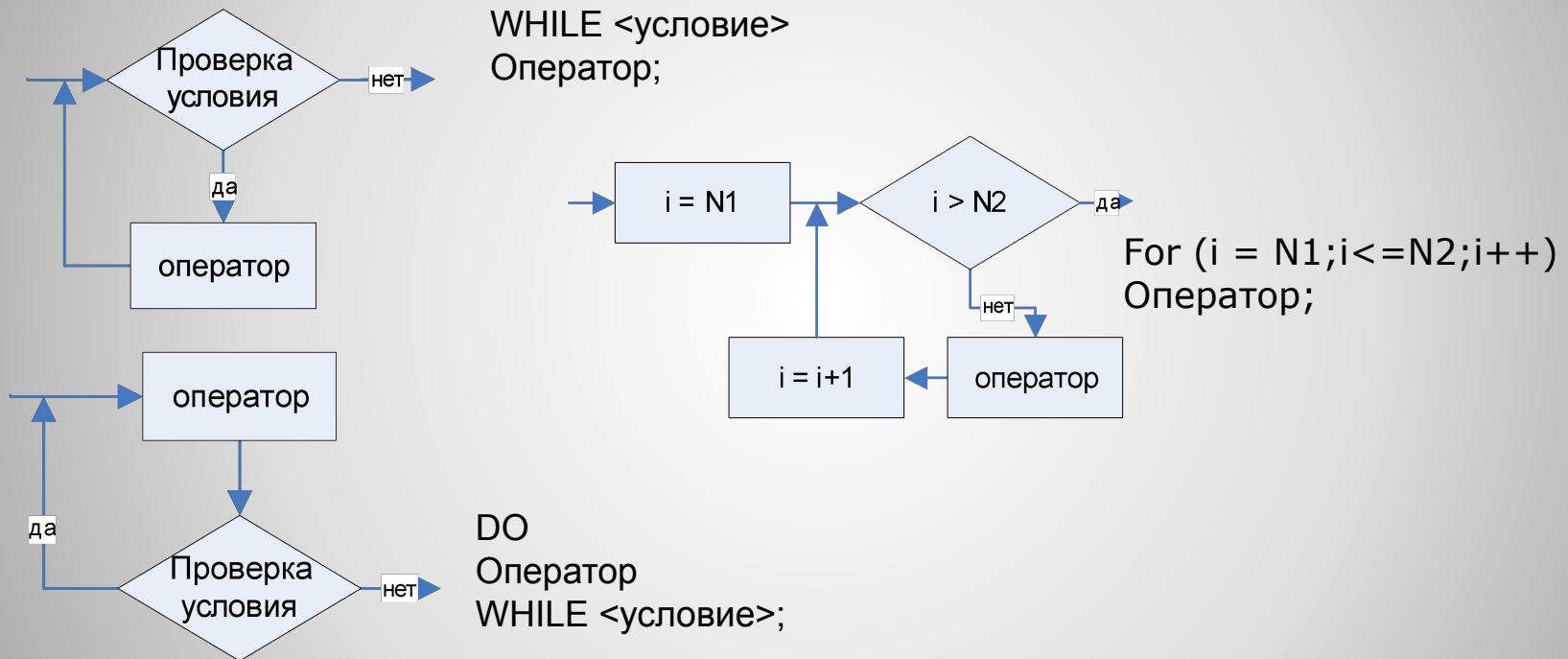
Console.WriteLine("f="+f);

}

- 1) Даны действительные положительные числа x, y, z . Выяснить, существует ли треугольник с длинами сторон x, y, z
- 2) Определить, какая из двух точек - $M_1(x_1, y_1)$ или $M_2(x_2, y_2)$ - расположена ближе к началу координат. Вывести на экран дисплея координаты этой точки
- 3) Определить, какая из двух фигур (круг или квадрат) имеет большую площадь. Известно, что сторона квадрата равна a , радиус круга r . Вывести на экран название и значение площади большей фигуры
- 4) Определить, попадает ли точка $M(x, y)$ в круг радиусом r с центром в точке (x_0, y_0)

Задачи на использование разветвляющихся алгоритмов

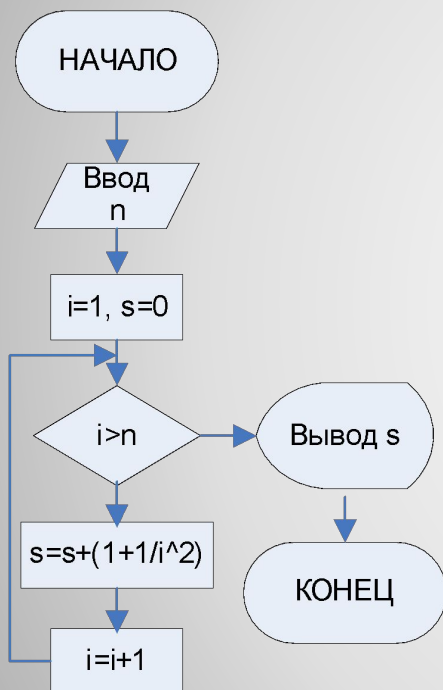
- Алгоритм называется циклическим, если он содержит многократное выполнение одних и тех же операторов при различных значениях промежуточных данных. Число повторений этих операторов может быть задано в явной (цикл с известным заранее числом повторений) или неявной (цикл с неизвестным заранее числом повторений) форме



Программирование циклических алгоритмов

● Пример: Дано натуральное n. Вычислить:

$$(1 + \frac{1}{1^2}) + (1 + \frac{1}{2^2}) + (1 + \frac{1}{3^2}) + \dots + (1 + \frac{1}{n^2})$$



```
Void main()
```

```
{
```

```
    Int i,n; Double s;
```

```
    n = Int32.Parse(Console.ReadLine());
```

```
    for(i=1,s=0;i<=n;i++)
```

```
        s+= 1 + 1.0 / (i * i);
```

```
    Console.WriteLine("s="+s);
```

```
}
```

Пример циклического алгоритма

1) Дано натуральное n . Вычислить: $\frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + \dots + \frac{1}{\sin 1 + \dots \sin n}$

2) Дано действительное число x , натуральное число n .
Вычислить:

$$x(x - n)(x - 2n)(x - 3n) \dots (x - n^2)$$

$$\frac{1}{x} + \frac{1}{x(x + 1)} + \dots + \frac{1}{x(x + 1) \dots (x + n)}$$

$$\frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

Задачи на использование циклических алгоритмов