

26.09.2020

Что такое программирование. Язык программирования java

Этапы решения задач на компьютере

Этап

Результат

1..Постановка задачи

Словесная информационная модель

2. Формализация задачи

Математическая модель

3. Построение алгоритма
(Алгоритмизация)

Алгоритм

4. Программирование

Программа

5. Отладка, тестирование программы

Уточнение модели

6. Выполнение расчётов,
анализ результата

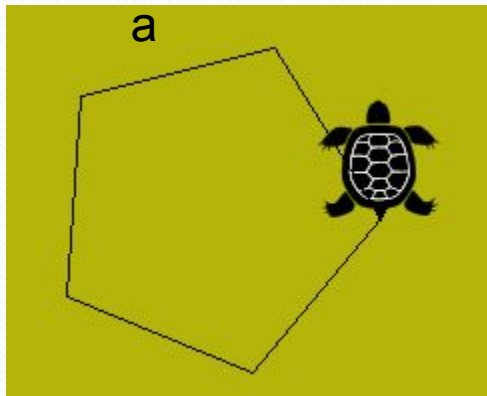
Получение результата

Работа без
компьютера

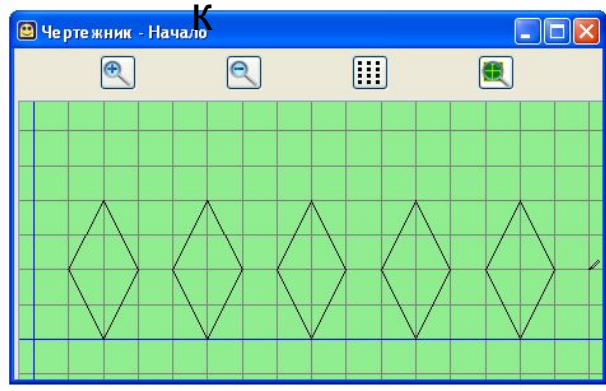
Понятие алгоритма

Алгоритм – это последовательность команд управления каким-либо исполнителем.

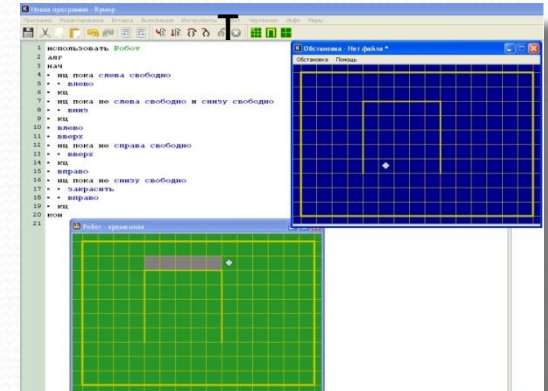
Черепашк



Чертежни



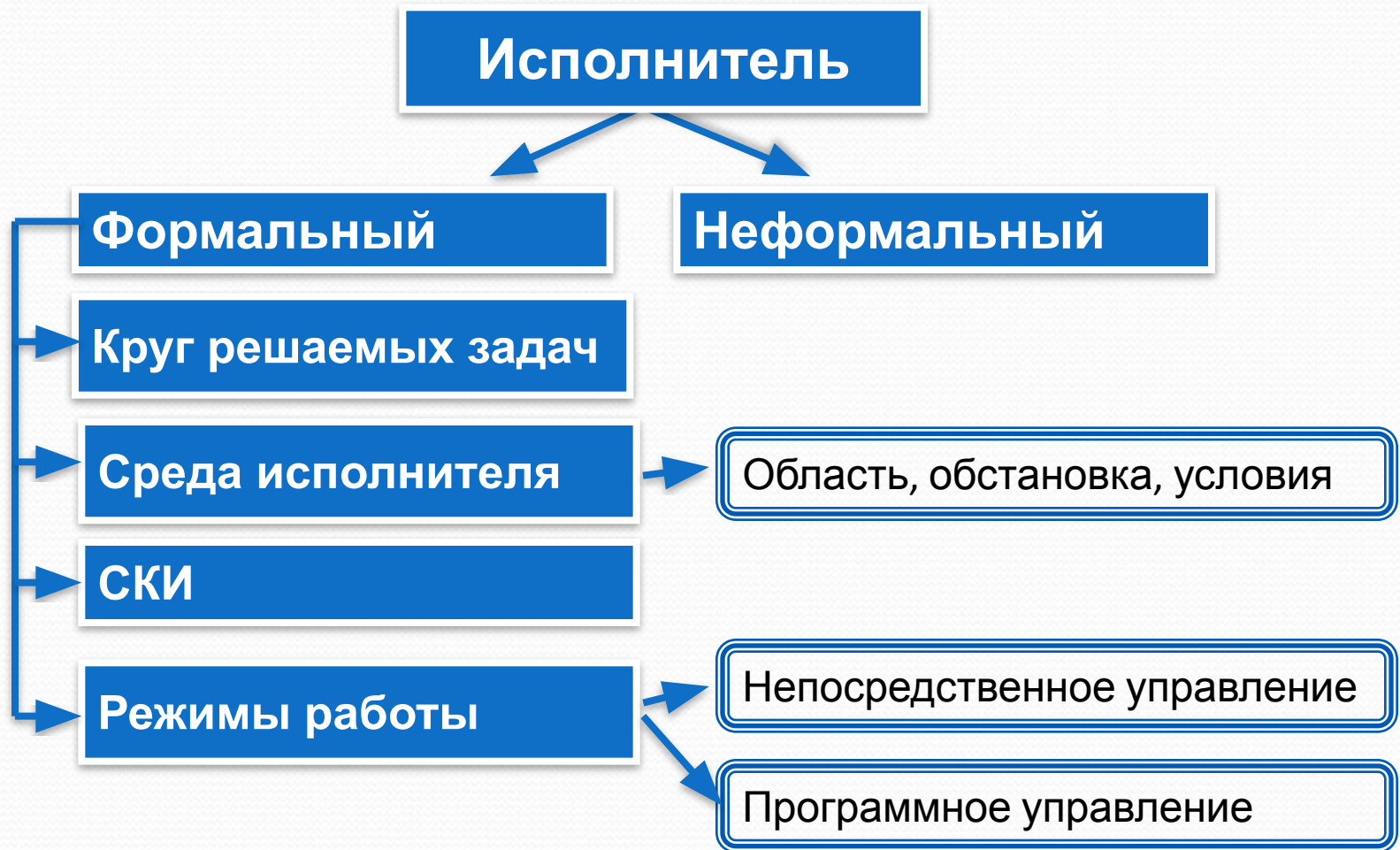
Робо



Эти исполнители ничего не вычисляют, они создают рисунки на экране, перемещаются в лабиринтах, перетаскивают предметы с места на место. (**исполнители, работающие в обстановке**)

Исполнитель алгоритма

Исполнитель - это некоторый объект (человек, животное, техническое устройство), способный выполнять определённый набор команд.



Разработка алгоритма



Алгоритм – модель деятельности исполнителя алгоритмов

Для записи алгоритмов используются специальные языки:

1 Естественный язык (словесная запись)

2 Формулы

3 Псевдокод

4 Структурограммы

5 Синтаксические диаграммы

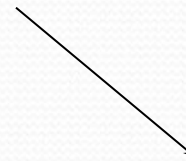
6 Графический (язык блок-схем)

Назначение *программирования* – разработка программ управления компьютером с целью решения различных информационных задач.

Программирование



Системное



Прикладное

Системное программирование – разработка системного программного обеспечения: операционных систем, утилит и т.д.



Прикладное программирование – создание прикладных программ: редакторы, табличные процессоры, игры, обучающие программы и т.д.



Для составления программ существуют разнообразные языки программирования.



Язык программирования – это фиксированная система обозначений для описания алгоритмов и структур данных.

- За годы существования ЭВМ было создано много языков программирования. Наиболее известные среди них: Фортран, Паскаль, Бейсик, С (Си) и др.
- Распространенными языками программирования сегодня являются C++, Delphi, Java, Pascal, Visual Basic, Python.

Языки программирования

Название	Название	Особенности
Лого	Logo	Язык, рассчитанный на детей, позволяющий просто и занимательно рисовать картинки и программировать простейшие игры.
Бейсик	Basic	Язык как для начинающих, так и для профессиональных программистов.
Паскаль	Pascal	Универсальный язык, позволяющий прекрасно программировать самые разные задачи.
Си	C	Сложный, мощный язык для профессиональных программистов.
Ассемблер	Assembler	Сложный, мощный язык с очень мелкими командами, близкими к командам машинного языка.
Ява (Джава)	Java	Мощный язык. Изначально применялся в основном в Интернете

Система программирования – это программное обеспечение компьютера, предназначенное для разработки, отладки и исполнения программ, записанных на определенном языке программирования.

Составляющие СП:

→ **текстовый редактор** (редактор кода)

Предназначен для набора текста программы.

→ **транслятор** – программа, которая служит для перевода исходного текста с языка высокого уровня (Pascal, BASIC, C) на машинный язык.

→ **компоновщик**

Осуществляет компоновку ваших программ и модулей и стандартных программ и модулей.

→ **библиотеки стандартных подпрограмм**

→ **отладчик** – позволяет выполнять программу по шагам и после каждого шага следить за состоянием всех переменных.

Если все эти программы представляются в виде одной, то такая программа называется интегрированной средой разработки (IDE). Большинство популярных языков снабжены IDE.

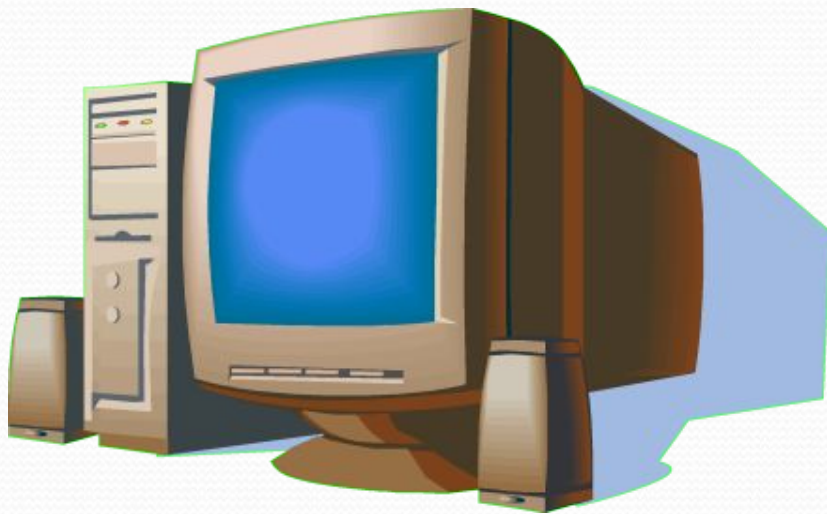
Преимущества *программирования на Java*

код, написанный на java, переводится с помощью соответствующей программы-транслятора в т.н. байт-код, а он, в свою очередь, — в машинный код с помощью виртуальной машины Java (Java Virtual Machine, или просто JVM).

Поэтому написанная на Java программа выполняется на любом компьютере с любой системой, лишь бы на нем была установлена виртуальная машина Java.

Разработка любой программы начинается с построения алгоритма решения задач. Такие алгоритмы называют алгоритмами работы с величинами.

В качестве исполнителя рассматривается – компьютер, оснащенный системой программирования на определенном языке.



Компьютер-исполнитель работает с определенными данными по определенной программе.

Язык программирования Паскаль

создан **Никлаусом Виртом** в **1968-1971**, как язык для обучения процедурному программированию.



Название языку дано в честь выдающегося французского математика, физика, литератора и философа **Блеза Паскаля (1623 – 1662)**.

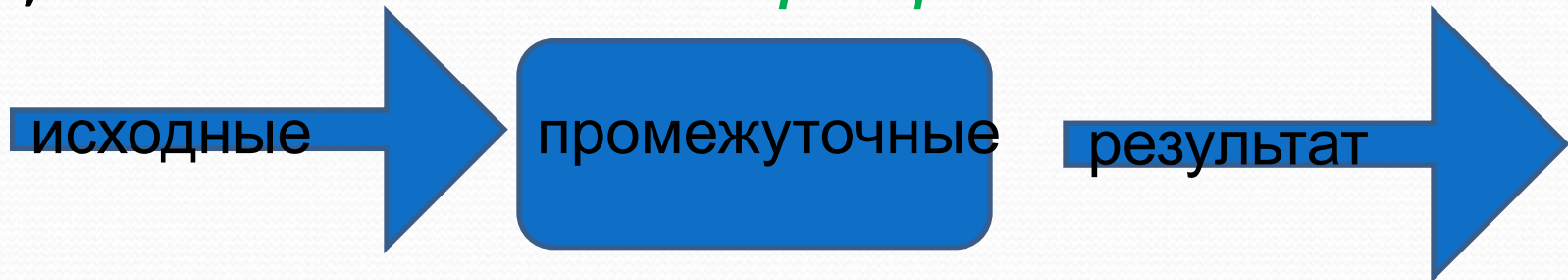


Особенностями языка являются строгая типизация и наличие средств структурного (процедурного) программирования.

Данные и величины

Данные – совокупность величин, с которыми работает компьютер.

1) *по отношению к программе:*

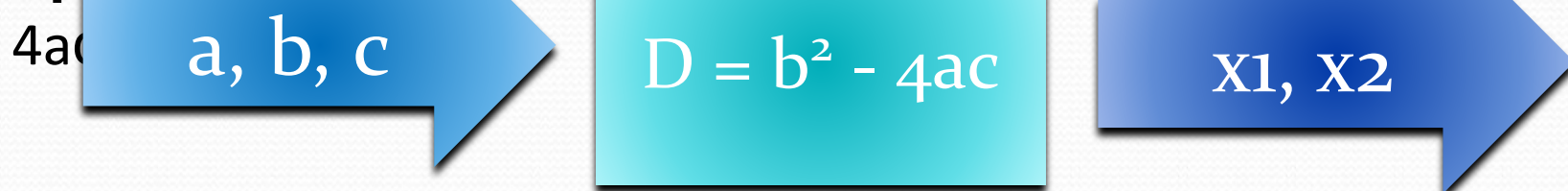


Пример: решение квадратного уравнения $ax^2 + bx + c = 0$

исходные данные: коэффициенты a, b, c ,

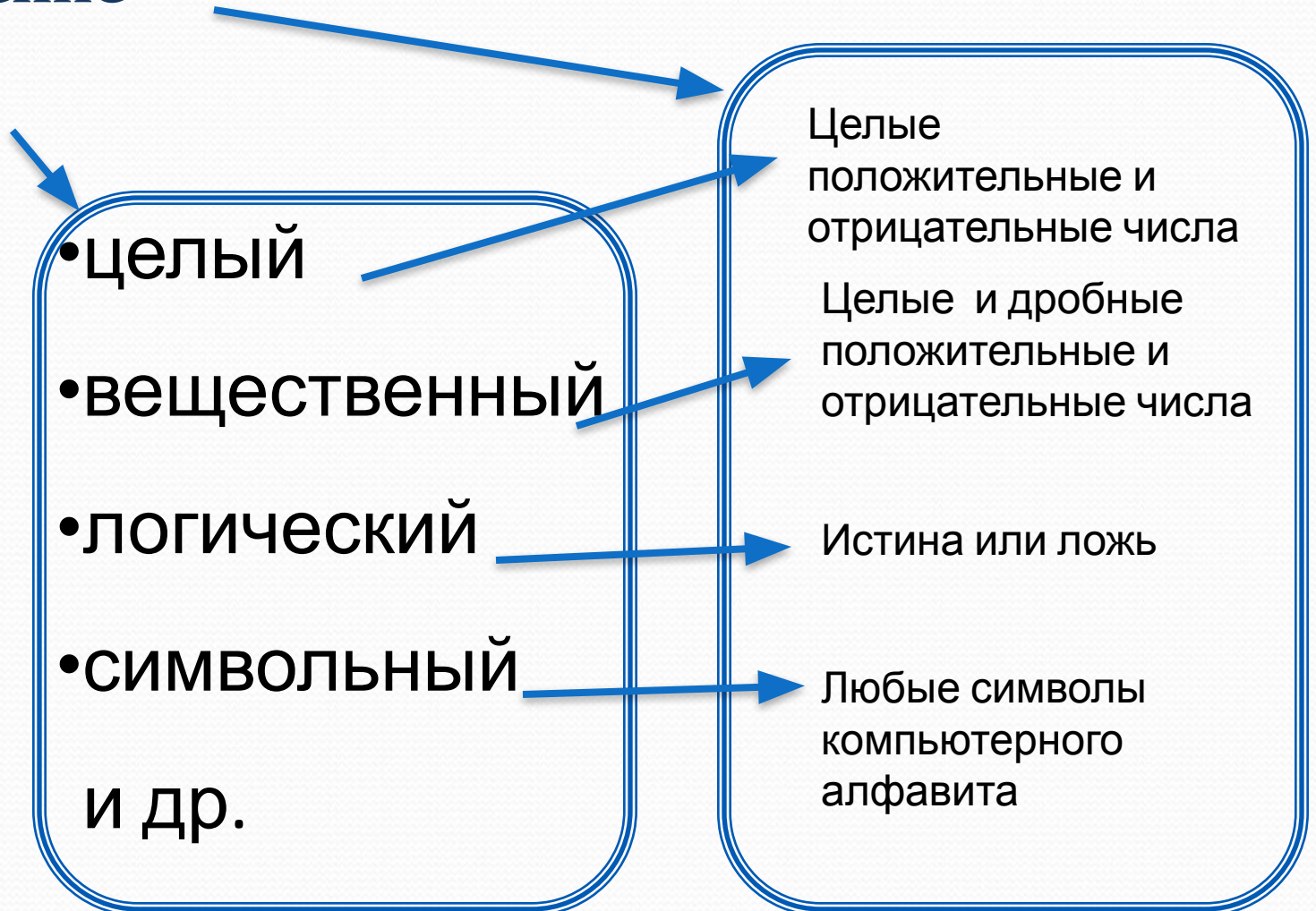
результат: корни уравнения x_1, x_2 ,

промежуточные данные: дискриминант $D = b^2 - 4ac$



Свойства данных

- Имя
- Значение
- Тип



Типы данных

Тип	Значения	Операции	Внутреннее представление
Целый	Целые положительные и отрицательные числа в некотором диапазоне. Примеры: 23, -12, 387	Арифметические операции с целыми числами: +, *, целочисленное деление и остаток от деления. Операции отношений (<, >, = и др.)	Формат с фиксированной запятой
Вещественный	Любые (целые и дробные) числа в некотором диапазоне. Примеры: 2.5, -0.01, 45.0, 3.6 · 10 ⁹	Арифметические операции: +, *, /. Операции отношений	Формат с плавающей запятой
Логический	true (истина) false (ложь)	Логические операции: И (and), ИЛИ (or), НЕ (not). Операции отношений	1 бит: 1 — true; 0 — false
Символьный	Любые символы компьютерного алфавита. Примеры: 'a', '*5',	Операции отношений	Коды таблицы символьной кодировки. 1 символ — 1 байт

Классификация данных

2) по значениям:

- **константы** (неизменная величина, в алгоритме представляется собственным значением)
- **переменные** (изменяют свои значения в ходе выполнения программы и представляются символическими именами — идентификаторами)

3) по типам:

- **целый**
- **вещественный**
- **логический**
- **символьный**

4) по структуре:

- **простые** (одна величина – одно значение)
- **структурированные** (одна величина – множество значений)

Независимо от того , на каком языке программирования будет написана программа , **алгоритм решения любой задачи на компьютере**

может быть составлен из команд:

- присваивания (переменная:=выражение);
- ввода;
- вывода;
- обращения в вспомогательному алгоритму (подпрограмме);
- цикла;
- ветвления.

Команда присваивания

Формат:

переменная=выражение

(Знак «=» нужно читать как «присвоить».)

Действия, выполняемые компьютером:

1. Вычисляется выражение.
2. Полученное значение присваивается переменной.

Пример

а)

Шаг	Команда	S
1	$S=5$	5
2	$S=57$	57

Ответ: $S=57$.

Пример

б)

Шаг	Команда	S
1	$S=6$	6
2	$S=-5.2*S$	-31.2
3	$S=0$	0

Ответ: $S=0$.

Пример

в)

Шаг	Команда	S
1	$S = -7.5$	-7.5
2	$S = 2 * S$	-15

Ответ: $S = -15$.

Пример

а)

Шаг	Команда	S	K
1	$S=45$	45	---
2	$K=-25$	45	-25
3	$S=S+K$	20	-25

Ответ: $S=20$

Базовые алгоритмические конструкции

Теорема Э. Дейкстры: Для записи любого алгоритма достаточно трёх основных алгоритмических конструкций:

- следования,
- ветвления,
- повторения.



При этом структуры могут располагаться последовательно друг за другом или вкладываться друг в друга.







Методика программирования, основанная на этой теореме, называется **структурным программированием**.

Эдсгер Вибс Дейкстра (1930–2002). Выдающийся нидерландский учёный, идеи которого оказали огромное влияние на развитие компьютерной индустрии.

Элементы блок-схемы

 <p>Начало</p>	Начало алгоритма
 <p>Конец</p>	Конец алгоритма
	Выполняемое действие
	Условие выполнения действий
	Счетчик количества повторов
	Последовательность выполнения действий

Элементы блок-схемы

Обозначение	Описание	Примечания
	Начало и конец алгоритма	
	Ввод и вывод данных.	Вывод данных иногда обозначают иначе: 
	Действие	В вычислительных алгоритмах так обозначают присваивание
	Развилка	Развилка - компонент, необходимый для реализации ветвлений и циклов
	Начало цикла с параметром	
	Типовой процесс	В программировании - процедуры или подпрограммы
	Переходы между блоками	

Запиши в тетрадь:

Графические формы записи алгоритмов:



Следование

Следование – линейная последовательность действий.

Алгоритмы, в которых используется только структура «следование», называются **линейными**.

Действие 1

Действие 2



Задача №1.

Найдите площадь треугольника с основанием A , высотой H .

алг

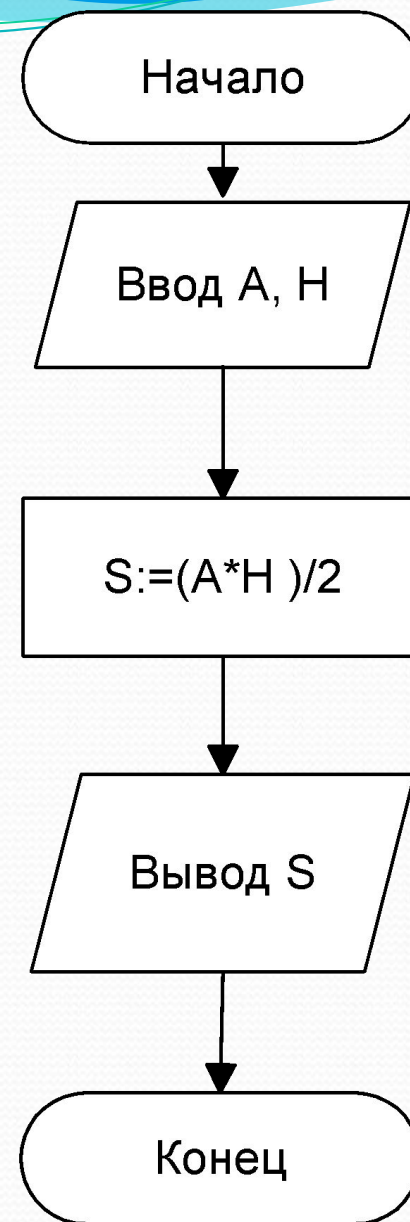
Нач

Ввод A, H

$S := (A * H) / 2$

Вывод S

кон



Задача №2.

В схематическом виде отразите изменения в ячейках, соответствующих переменным A и B, в ходе последовательного выполнения команд присваивания:

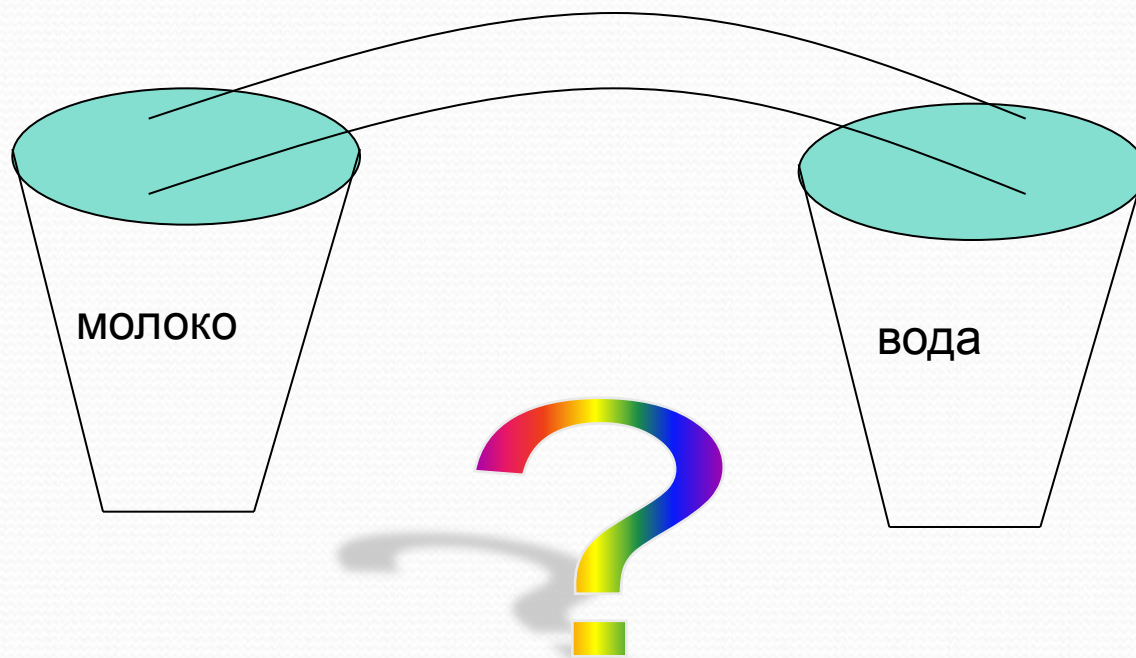
- 1) $A=1; B=2; A=A+B; B=2*A.$
- 2) $A=1; B=2; C=A; A=B; B=C.$
- 3) $A=1; B=2; A=A+B; B=A-B; A=A-B.$

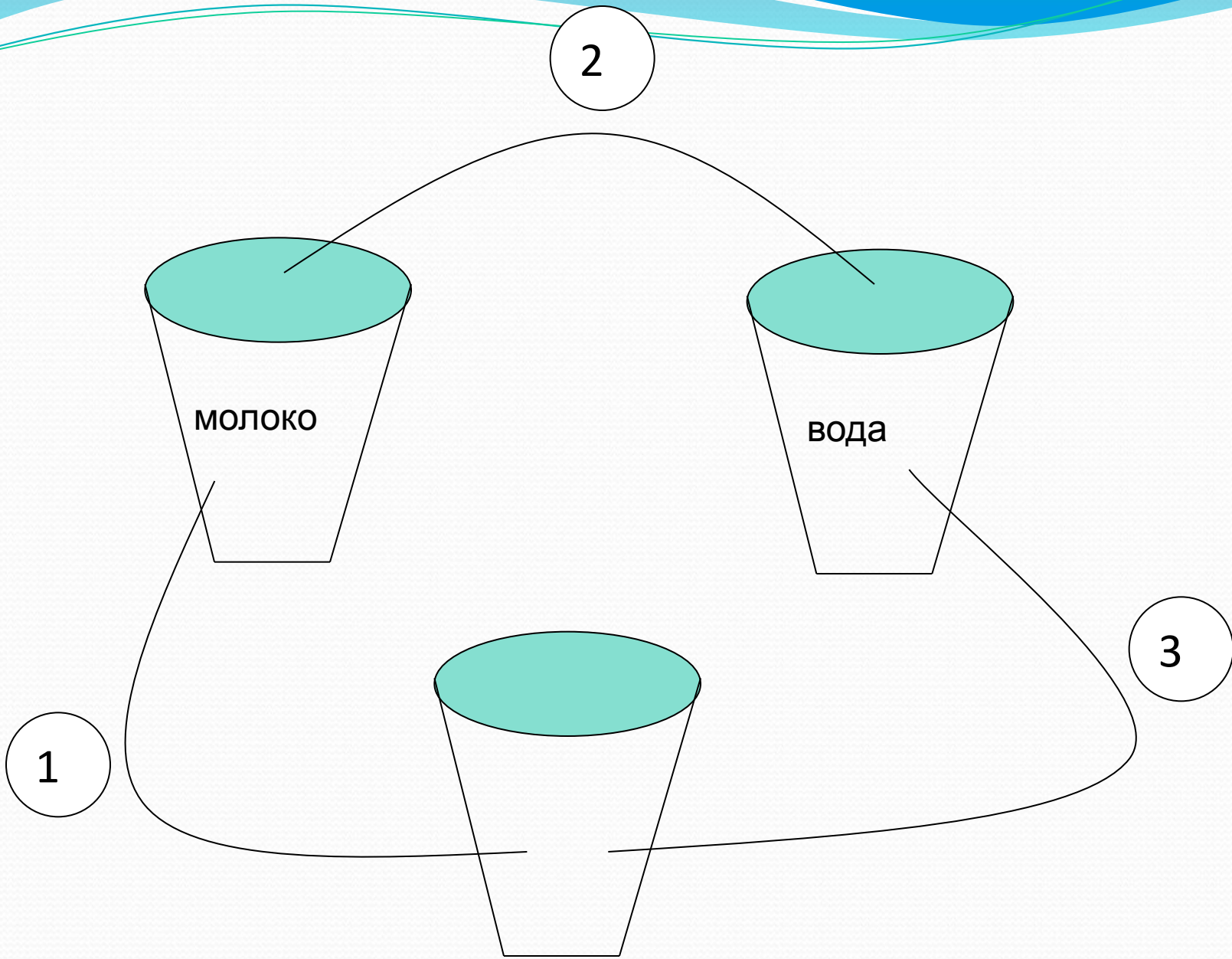
Задача 3.

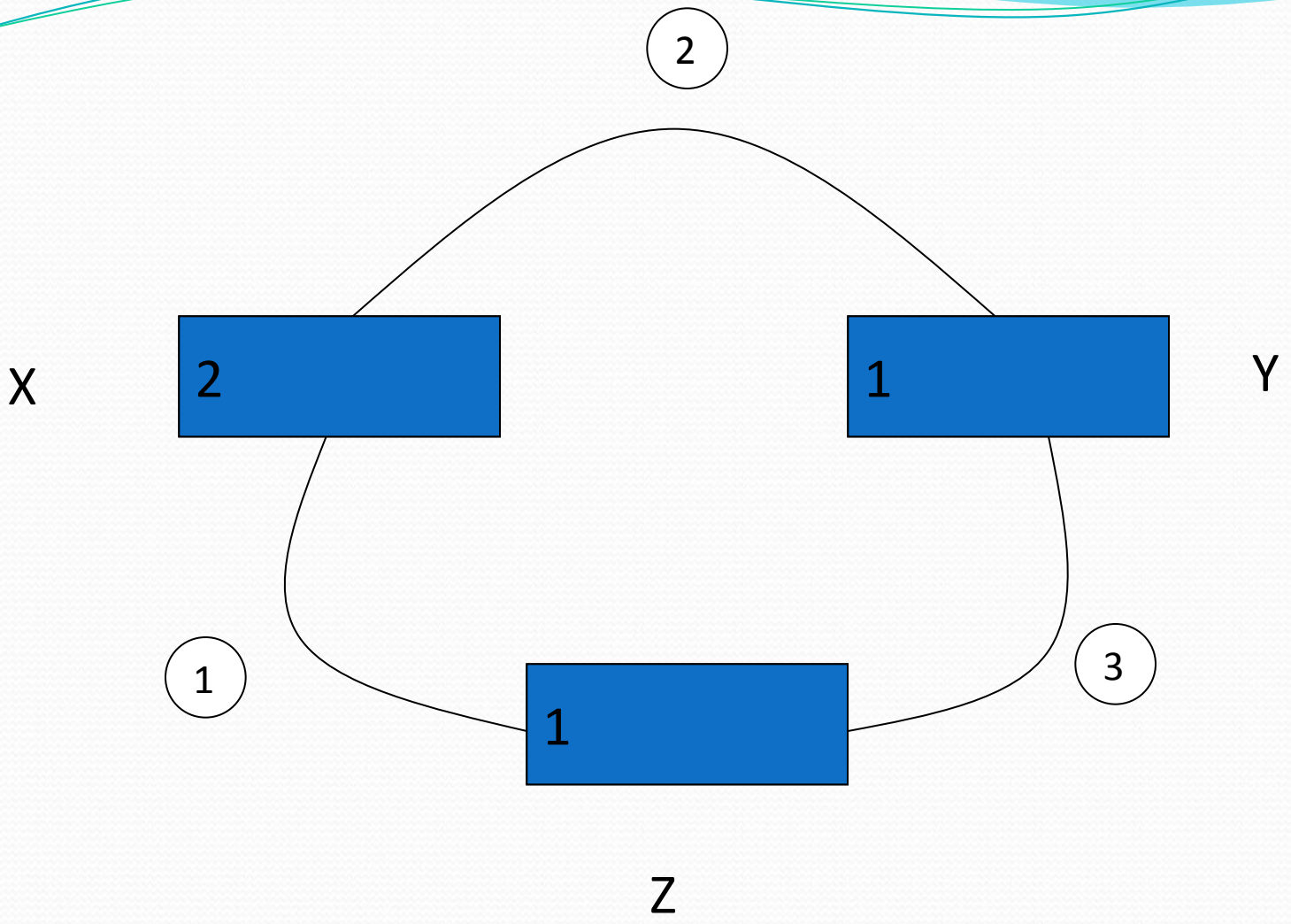
Вместо многоточия впишите в алгоритм несколько команд присваивания, в результате чего должен получиться алгоритм возведения в четвертую степень введенного числа:

ввод A ... вывод A .

Задача №3.







Алгоритм решения задачи:

ввод X, Y

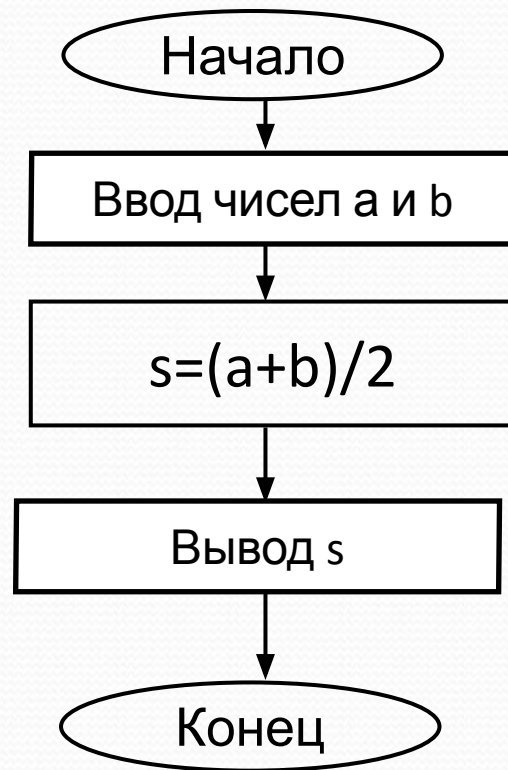
$Z=X$

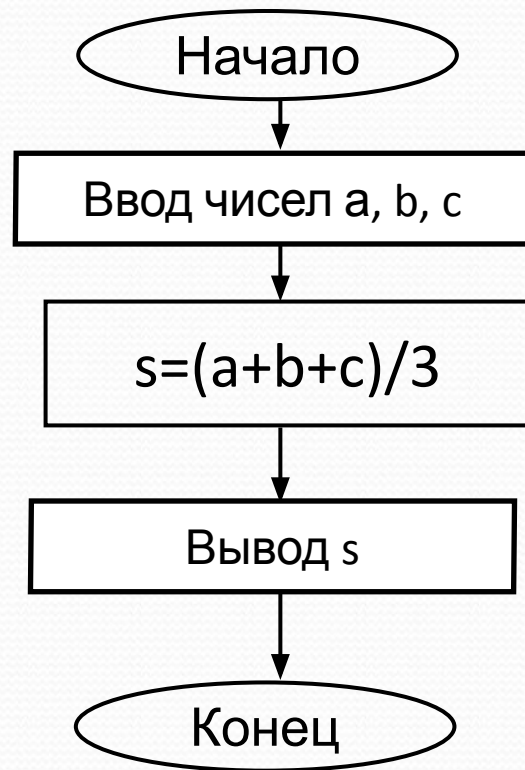
$X=Y$

$Y=Z$

вывод X, Y

Задача №4. Чему будет равно значение переменной s , если $a=5$, $b=7$?

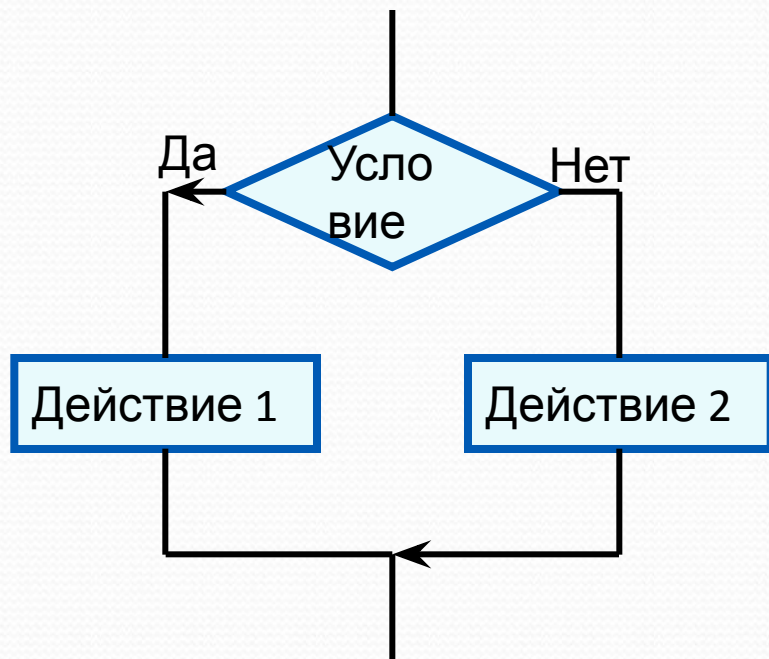




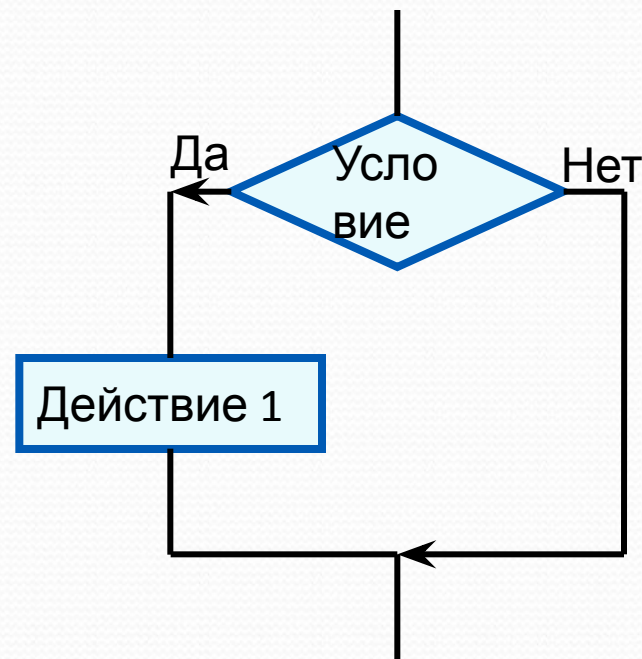
Ветвление

Ветвление - алгоритмическая конструкция, в которой в зависимости от результата проверки условия (да или нет) предусмотрен выбор одной из двух последовательностей действий (ветвей).

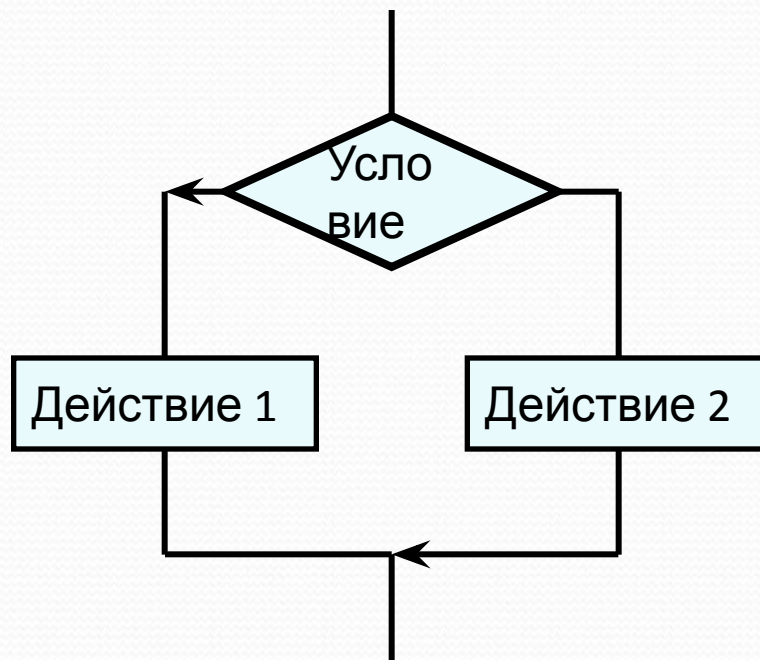
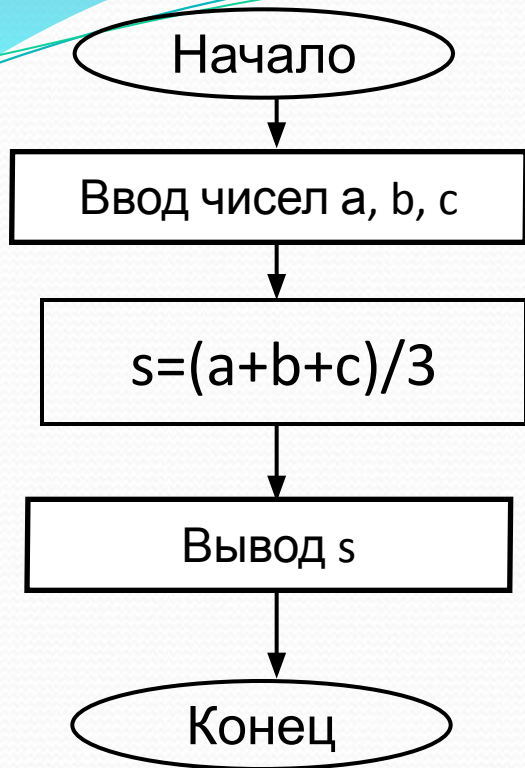
Алгоритмы, в основе которых лежит структура «ветвление», называют **разветвляющимися**.



Полная форма ветвления



Неполная форма ветвления



Задача №5. Найдите значение функции

$$y = \begin{cases} 3 * X, & \text{если } X \leq 0 \\ \frac{1}{X}, & \text{если } X > 0 \end{cases}$$

АЛГ функция (вещ X, Y)

АРГ X

РЕЗ Y

НАЧ если $x \leq 0$

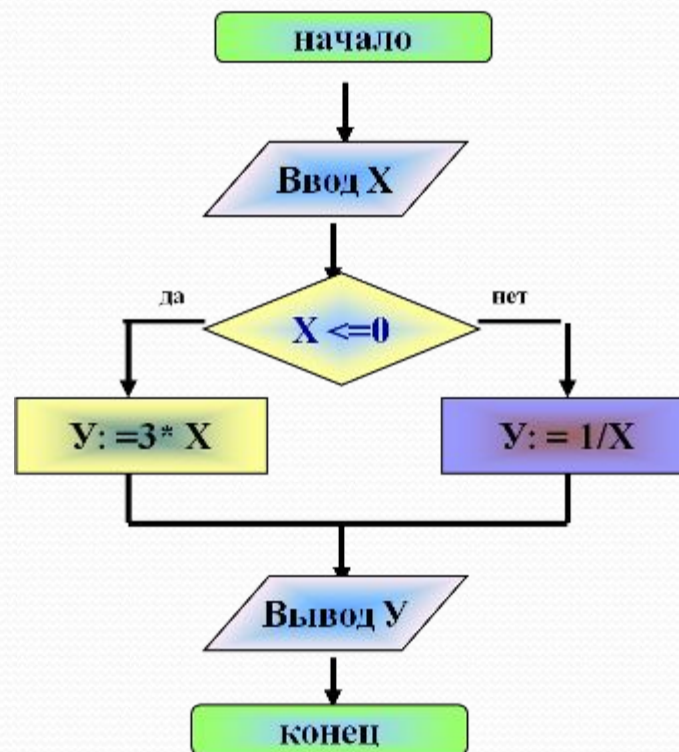
то $y = 3 * x$

иначе $y = 1 /$

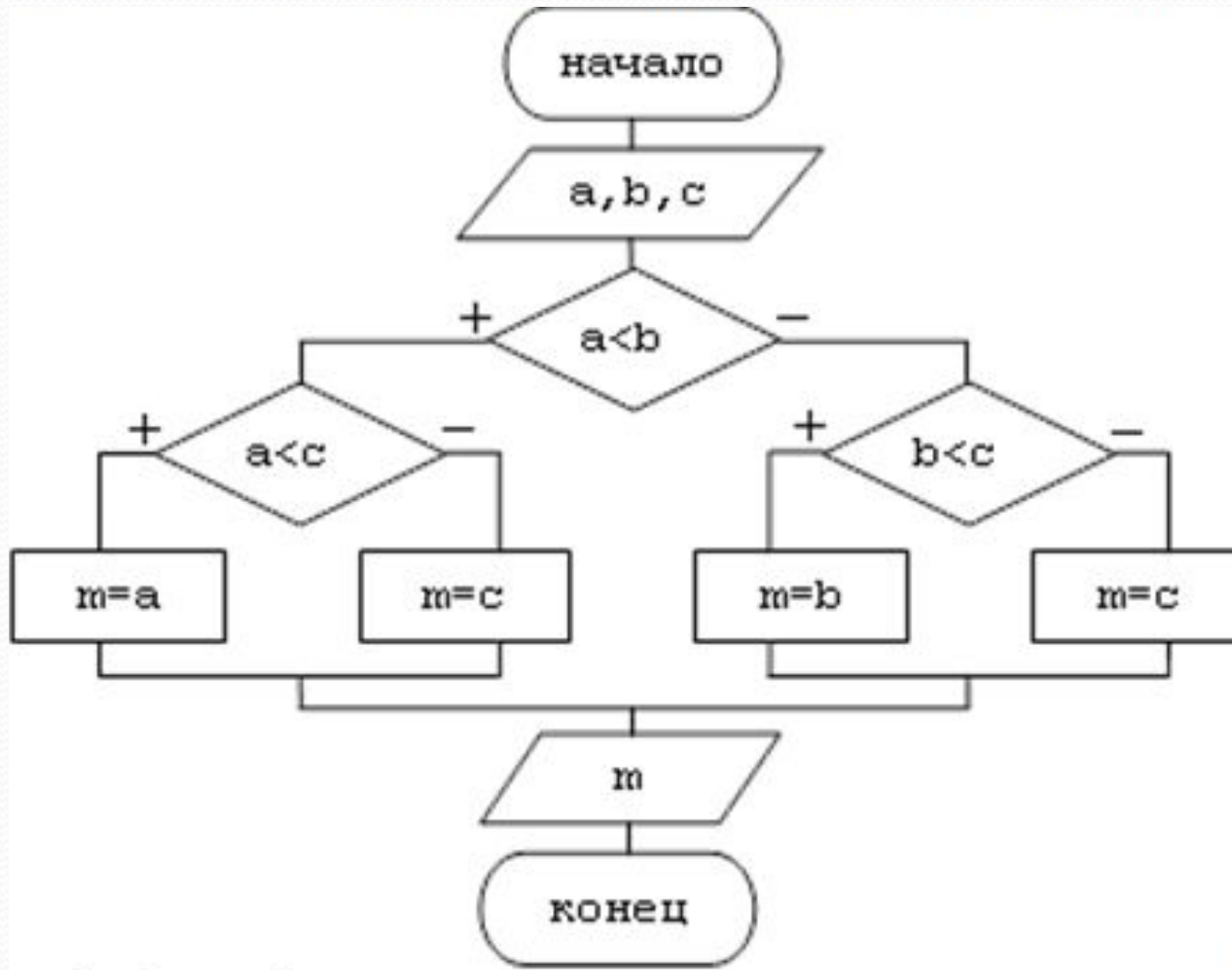
x

все

КОН



Задача №6. Что будет получено в результате выполнения алгоритма, если $A=7$, $B=9$, $C=3$?



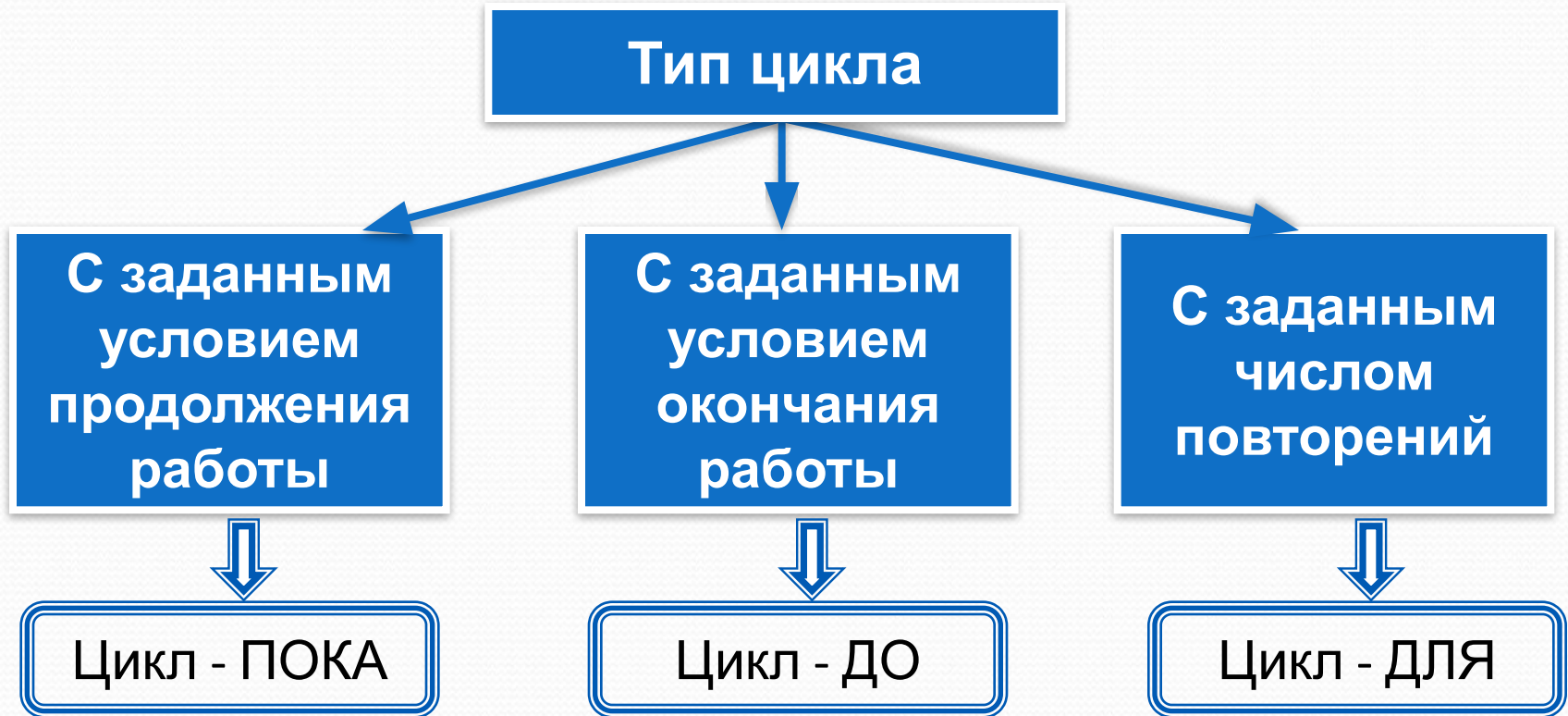
Цикл (повторение)

Цикл – повторение некоторой группы действий по

условию - алгоритмическая конструкция, представляющая собой последовательность действий, выполняемых многократно.

Алгоритмы, содержащие конструкцию «повторение», называют **циклическими** или **циклами**.

Последовательность действий, многократно повторяющаяся в процессе выполнения цикла, называется **телом цикла**.



Цикл – ПОКА

Пока условие истинно, выполняется серия, образующая тело цикла.

пока условие

нц

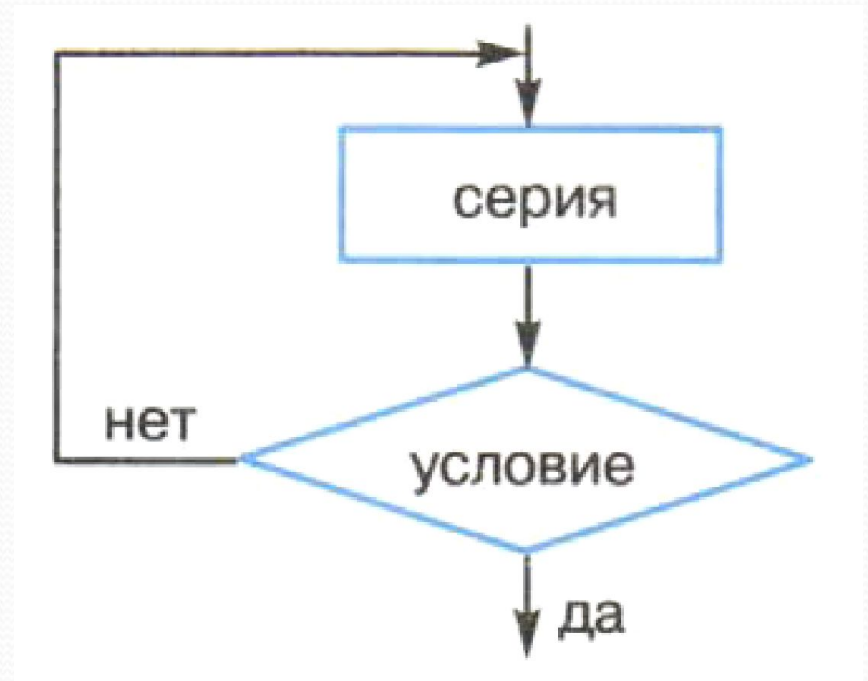
Тело цикла S 2

кц



Цикл - ДО

- Тело цикла предшествует условию цикла. Тело цикла повторяет свое выполнение, если условие ложно.
- Повторение прекращается, когда условие становится истинным.



Цикл – ДЛЯ

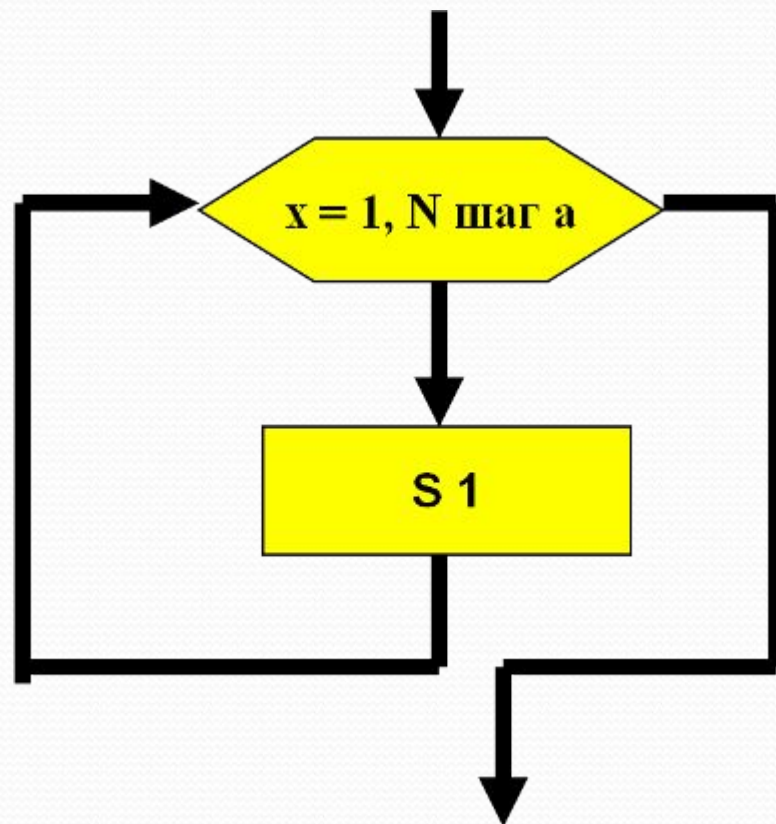
цикл, выполнение которого определяется значениями параметра

для X от 1 до N шаг a

нц

Тело цикла S1

кц



Пример Найти сумму натуральных нечетных

чисел от 1 до N

АЛГ сумма (цел N , S)

АРГ N

РЕЗ S

НАЧ

S := 0

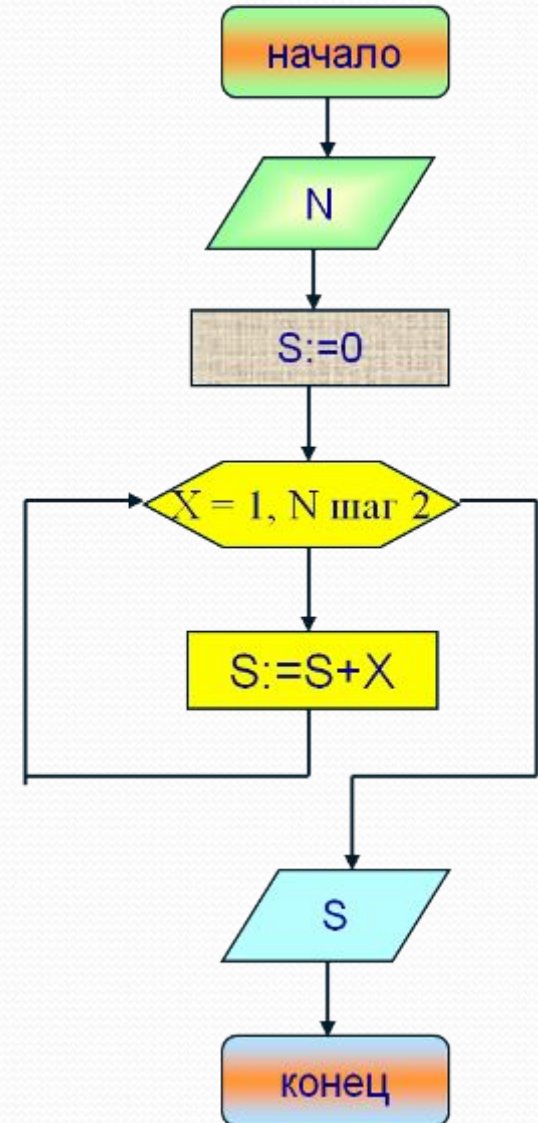
ДЛЯ X ОТ 1 ДО N ШАГ 2

НЦ

S := S + X

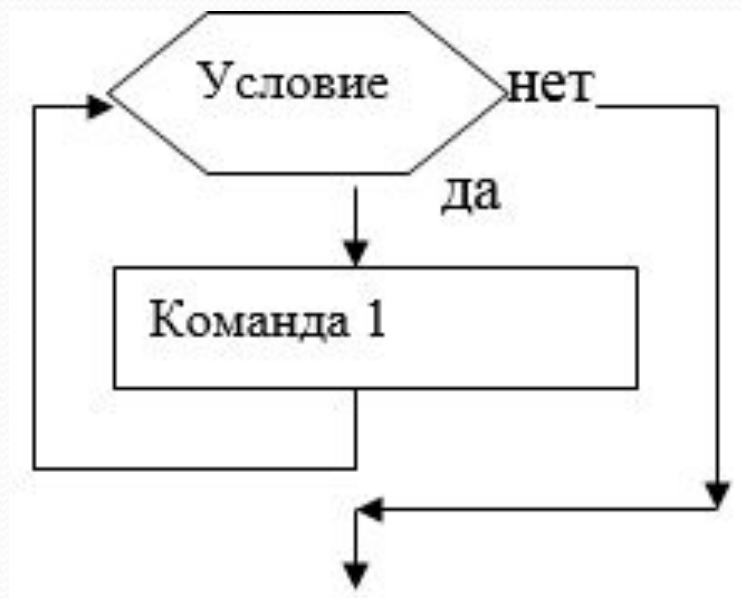
КЦ

КОН



Запиши в тетрадь:

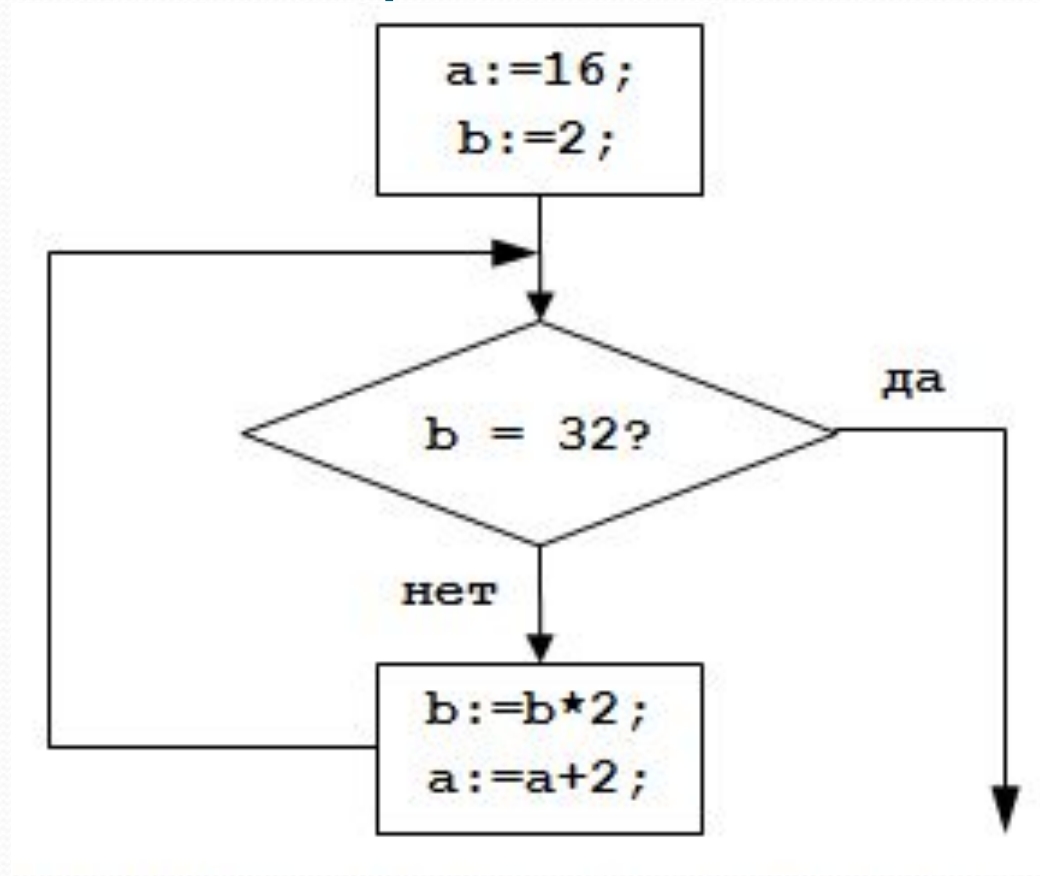
Цикл – многократное повторение действий по условию

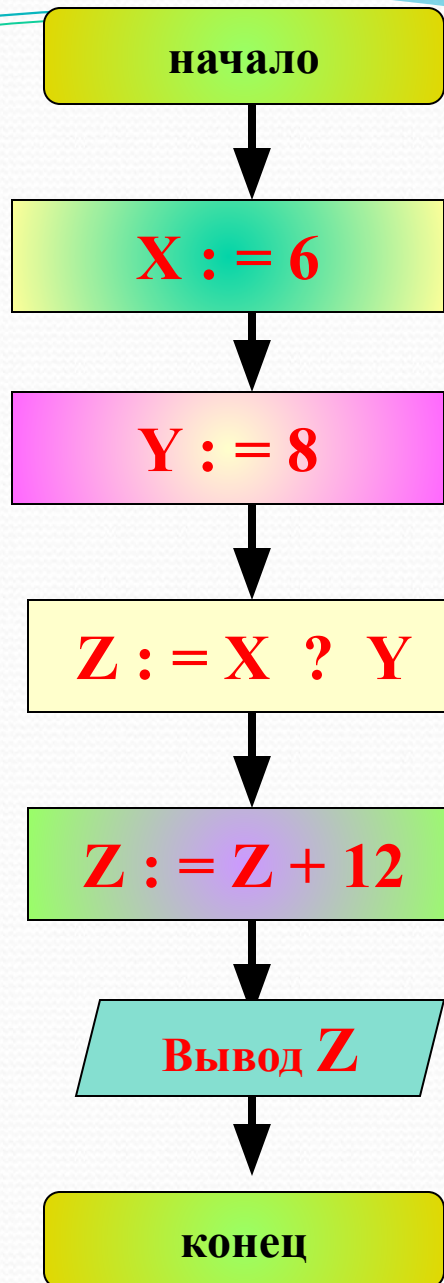


Пример блок-схемы «Переход дороги»



Задача №7. Определите значение
переменной a после выполнения
фрагмента алгоритма



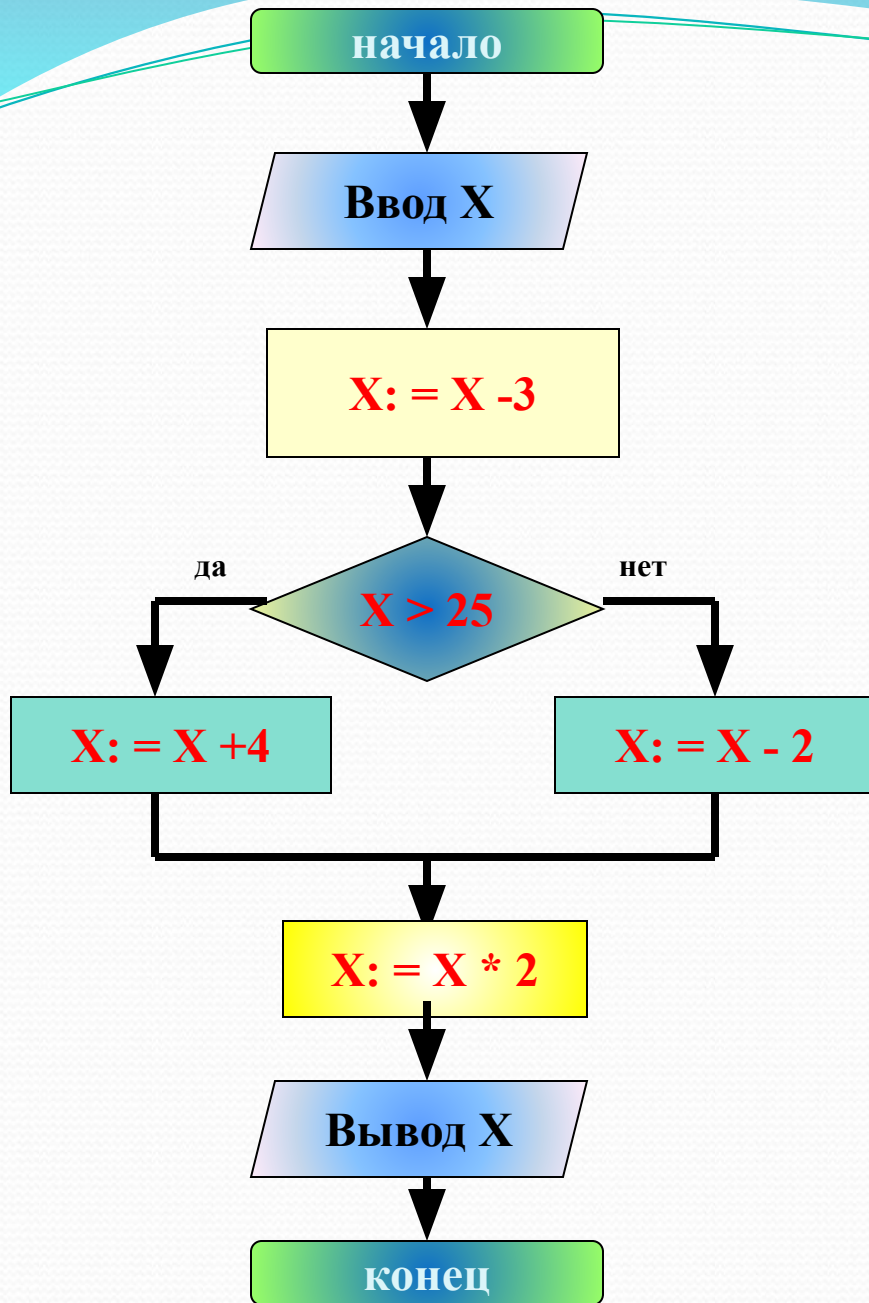


Задача №8.

1. Укажите тип данного алгоритма

2. В результате выполнения алгоритма было получено $z=60$. Укажите пропущенный оператор

Ответ: умножение

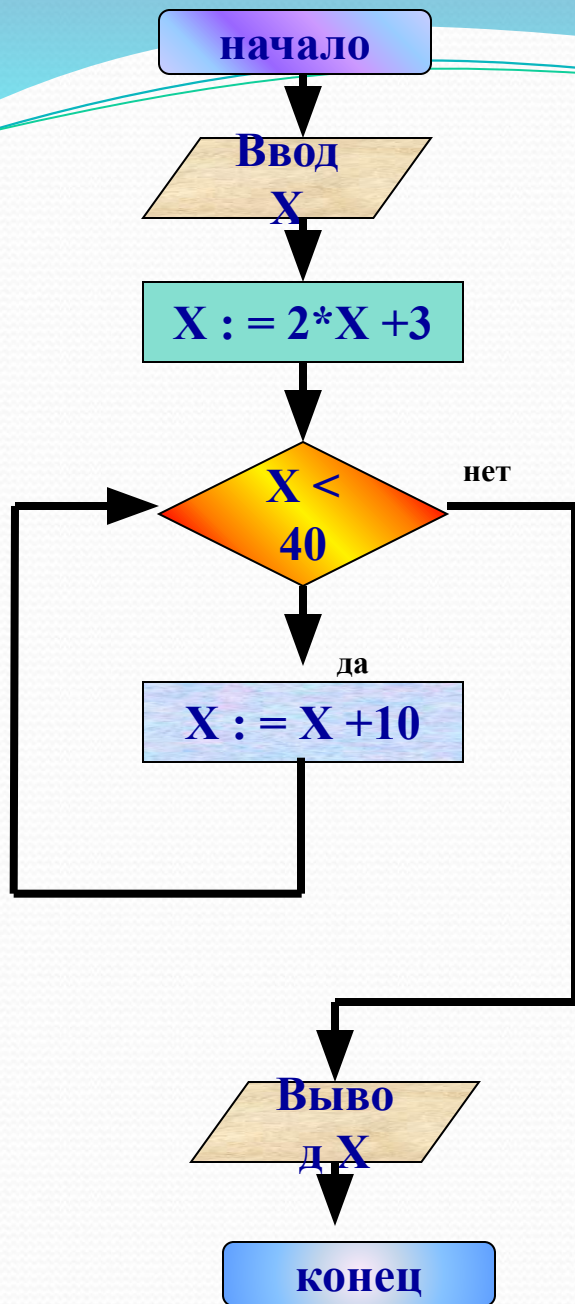


Задача №9.

1. Укажите тип данного алгоритма

2. В результате выполнения алгоритма было получено число **40**. С каким числом начал работать алгоритм

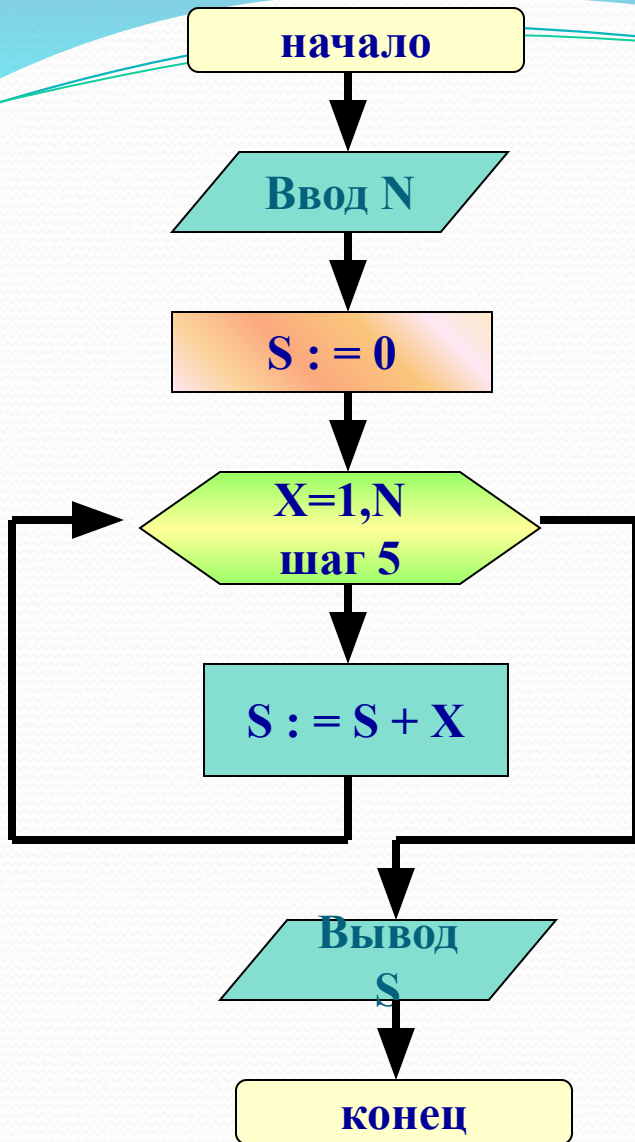
Ответ: X=25



Задача №10.

1. Укажите тип данного алгоритма
2. Введено число **1**. Выполните алгоритм и дайте ответ

Ответ: $X=45$



Задача №11.

1. Укажите тип данного алгоритма

2. Введено число **20**.
Выполните алгоритм и
дайте ответ

Ответ: X=34

Задача №12. Постройте блок-схему.

АЛГ площадь (цел X , Y , вещ S)

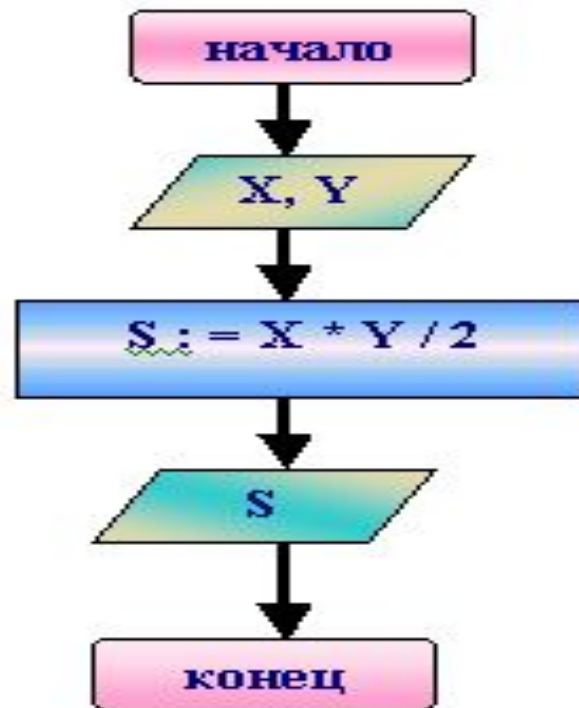
АРГ X , Y

РЕЗ S

НАЧ

$S := X * Y / 2$

КОН



Задача №13. Постройте блок-схему.

АЛГ сумма (цел N , S)

АРГ N

РЕЗ S

НАЧ

S := 0

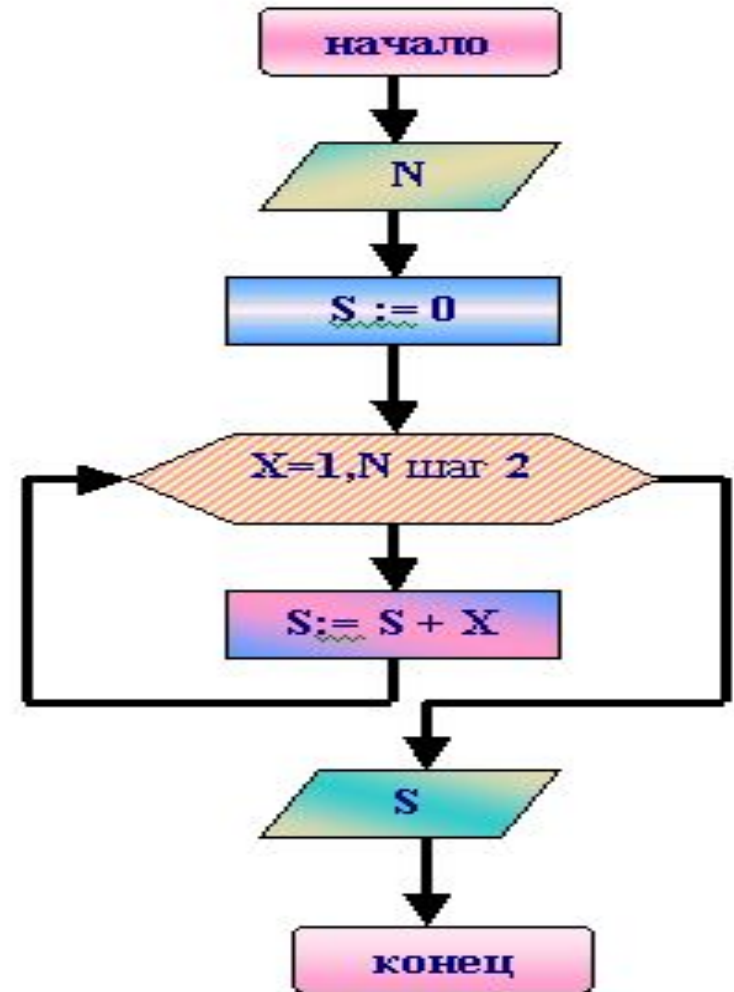
ДЛЯ X ОТ 2 ДО N ШАГ 2

НЦ

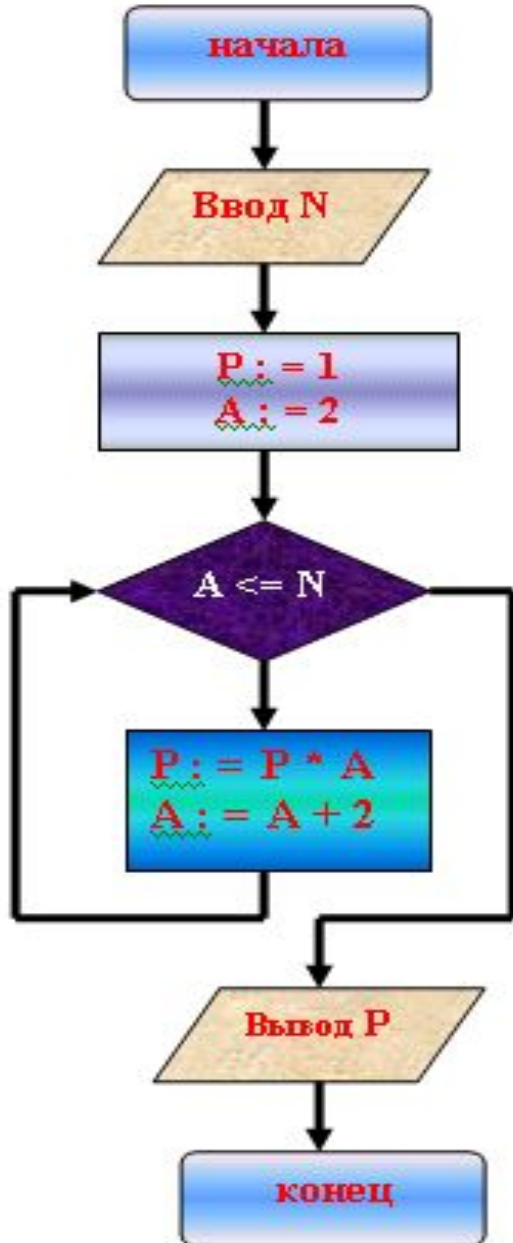
S := S + X

КЦ

КОН



Задача №14. Составьте алгоритм.



АЛГ произведение (цел N ,
 P)

АРГ N

РЕЗ S

НАЧ

$P := 1$

$A := 2$

ПОКА $A \leq N$

НЦ

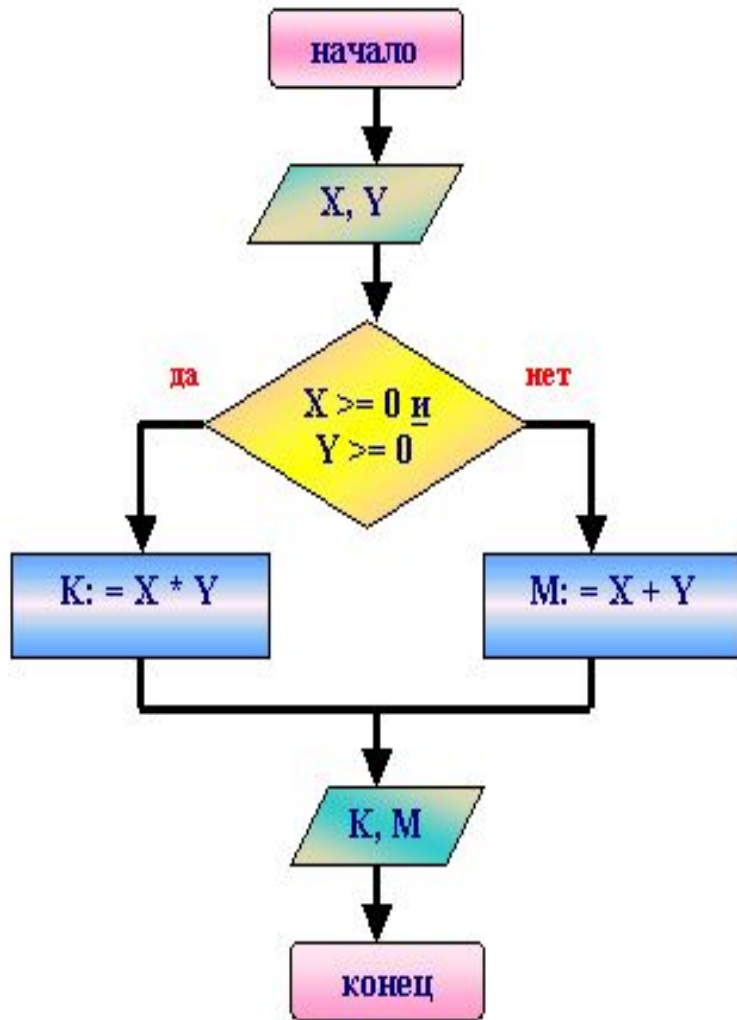
$P := P * A$

$A := A + 2$

КЦ

КОН

Задача №15. Составьте алгоритм.



АЛГ наименьшее (вещ X, Y, K, M)

АРГ X, Y

РЕЗ K, M

НАЧ

ЕСЛИ X >= 0 и Y >= 0

ТО K := X * Y

ИНАЧЕ M := X + Y

ВСЕ

КОН

Любой естественный язык (русский, английский, немецкий, ...)
изучается по следующей цепочке:

алфавит \Rightarrow слова \Rightarrow предложения \Rightarrow $\tau \Rightarrow \Gamma$

Этот же план можно применить и для изучения любого языка программирования.

Алфавит языка программирования – это набор символов, используемых в данном языке.

Алфавит языка JAVA содержит следующие символы:

1. 26 прописных и 26 строчных букв латинского алфавита: **A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, a, b, c, d, e, f, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, x** ;
2. Арабские цифры: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**.
3. Знак подчёркивания (**_**)

4. Специальные символы:

+	плюс	[]	квадратные скобки
-	минус	.	точка
*	умножение	,	запятая
/	деление	;	точка с запятой
>	больше	:	двоеточие
<	меньше	'	апостроф
>=	больше либо равно	{}	фигурные скобки
<=	меньше либо равно	()	круглые скобки
<>	не равно	^	тильда
№	номер	\$	знак денежной единицы

Слова

Слова в java - это **служебные** слова и **стандартные** функции.

Примеры служебных слов: `int`, `for`, `if`

Стандартные функции – это те элементарные функции, которые чаще всего используются в задачах на вычисления. При этом программисту вовсе необязательно знать, как вычисляется данная функция, ему достаточно правильно записать ее вид - `sin(x)`, `cos(x)`

Процедуры вывода

Write и WriteLn

(«пиши» и «пиши строку»)

С помощью данных операторов изображают на экране ту или иную информацию, состоящую из СИМВОЛОВ.

Выводить на экран можно не только числа, но и результаты вычисления арифметических выражений, а также тексты, которые, в отличие от чисел и выражений, нужно брать в одинарные кавычки.

Примеры:

<i>Как пишем</i>	<i>Что видим</i>
<code>Write(-500)</code>	-500
<code>Write(2*2-1)</code>	3
<code>Write('Хорошо!')</code>	Хорошо!

Один оператор Write может выводить сразу несколько элементов. Элементы нужно отделять друг от друга запятыми.

Все элементы выводятся в одну строку вплотную друг к другу.

На экране отображаются только те пробелы, которые встречаются внутри кавычек.

Примеры:

<i>Как пишем</i>	<i>Что видим</i>
<code>Write('Это',4+4,'Кошек')</code>	Это8Кошек
<code>Write('Это ',4+4,' кошек')</code>	Это 8 кошек
<code>Write('16+17=',16+17)</code>	16+17=33
<code>Write(3+2,' ',4)</code>	5 4
<code>Write(3+2,4)</code>	54
<code>Write('125+1',5+1,'=',120+21)</code>	125+16=141

Правила записи и выполнения оператора `WriteLn` те же, что и у `Write`, с одним исключением – после его выполнения следующий оператор `Write` или `WriteLn` печатает свою информацию с начала следующей строки, а после выполнения оператора `Write` продолжает печатать в той же.

Оператор `WriteLn` можно использовать просто для перевода курсора в начало следующей строки.

Программы на Паскале содержат следующие «знаки препинания»:

- Служебные слова BEGIN и END;
- Точка с запятой;
- Точка.

BEGIN (переводится – «начало») – ставят в начале программы, чтобы было видно, откуда она начинается.

END (переводится – «конец») – с точкой ставится в конце программы, чтобы было видно, где она заканчивается.

Точкой с запятой отделяют операторы друг от друга.

Служебные слова **BEGIN** и **END** от операторов точкой с запятой не отделяются.

Программу можно записывать и в строку, и в столбец.

Служебные слова и операторы могут быть записаны любыми буквами (заглавными или строчными, а также любым шрифтом).

Программа на Паскале может содержать комментарии, взятые в фигурные скобки, которые не влияют на выполнение программы.

Пример:

Программа на Паскале.

```
BEGIN
```

```
Write('Начали!'); {Это приказ печатать!}
```

```
Write(8+1);
```

```
Write(5);
```

```
END.
```

Результат выполнения

Начали!95

Примеры:

Программа: Begin Write('АМа'); Write('ЗОНКа'); End.

Результат: АМаЗОНКа

Программа: Begin Write('АМа'); WriteLn('ЗОНКа'); End.

Результат: АМаЗОНКа

Программа: Begin WriteLn('Ама'); Write('Зонка'); End.

Результат:
Ама

Зонка

Программа: Begin WriteLn('Ама'); WriteLn('Зонка'); End.

Результат:
Ама
Зонка

Задача 1

Определить, что напечатает программа:

```
Begin
```

```
Write(1992);
```

```
WriteLn(' Мы начинаем!');
```

```
WriteLn(6*8);
```

```
WriteLn;
```

```
WriteLn('Шестью шесть ',6*6,'.Арифметика:',(6+4)*3);
```

```
End.
```

Оператор присваивания.

При выполнении оператора присваивания компьютер «в уме» вычисляет правую часть и присваивает вычисленное значение переменной, стоящей в левой части.

Обозначение оператора присваивания

=

Пример:

Begin

a=2*3+4;

b=a;

y=a+b+1;

System.out.println(y);

End.

a:=10;

b:=10;

y:=10+10+1;

y=21

Замечание. Если переменная принимает новое значение, то старое значение автоматически стирается

Описание переменных

Описание переменных начинается со служебного слова VAR (переводится – «переменная»), которое записывается выше Begin.

После VAR записываются имена всех переменных, встречающихся в программе с указанием через двоеточие типа значений, которые каждая переменная имеет право принимать.

Типы значений переменных

Тип	Перевод	Диапазон принимаемых значений
Integer	целый	целые числа от - 32 768 до 32 767
LongInt	длинное целое	целые числа от - 2 147 483 648 до 2 147 483 647
Byte		целые числа от 0 до 255
Real	Вещест- венный	целые и дробные числа

Для того, чтобы Паскаль выводил вещественные числа в понятном виде, нужно в оператор вывода `WriteLn` дописывать формат численного значения переменной:

`WriteLn(x:n:m),`

где **n** – натуральное число, показывающее сколько символов, включая целую часть, дробную часть, знак и десятичную точку, должно занимать все изображение числа; **m** – натуральное число, показывающее количество символов после десятичной точки.

Пример:

```
Var a,b:Integer;
```

```
    c:Real;
```

```
Begin
```

```
    a:=6;
```

```
    b:=7;
```

```
    c:=b/a;
```

```
    WriteLn('c=',c:4:2);
```

```
End.
```

Ответ: c=1,17

СПИСОК ИСТОЧНИКОВ

- Информатика и ИКТ: Учебник для 9 класса / И.Г.Семакин Л.А. Залогова, С.В. Русаков, Л.В. Шестакова.
- <http://informaticweb.altervista.org/images/sistemaoperativi.jpg>
- (немного изменен)
<http://seogad.ru/wp-content/uploads/2011/02/12-02-2011-soft.jpg>
- http://farm4.static.flickr.com/3227/2662210806_c7bbc90786.jpg
- <http://www.terageeks.net/images/computer2.gif>
- <http://informatika.mksat.net/wp-content/uploads/2012/01/virt.jpg>