



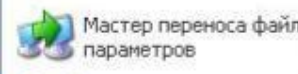
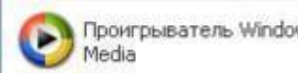
# Алгоритмический язык КуМир

*Мазеева Татьяна Александровна, учитель информатики  
МКОУ «СОШ №3» г. Николаевска Волгоградской обл.  
2011г.*

# Содержание

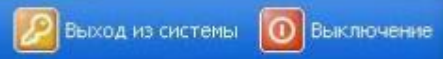
- Интерфейс окна программы
- Линейный алгоритм
- Структуры «Ветвление»
- Одномерный массив. Циклические конструкции
- Обработка двумерных массивов
- Литерные величины



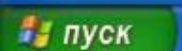


- Program Updates
- Windows Update
- Выбор программ по умолчанию
- Каталог Windows
- Центр решений HP
- 7-Zip
- avast! Free Antivirus
- GIMP
- HP
- McAfee Security Scan Plus
- Mozilla Firefox
- OpenOffice.org 2.4
- WinRAR
- Автозагрузка
- Игры
- Интернет Цензор
- Стандартные
- Adobe Reader X
- Internet Explorer
- Outlook Express
- Windows Messenger
- Windows Movie Maker
- Многопользовательские игры
- Прогрыватель Windows Media
- Удаленный помощник

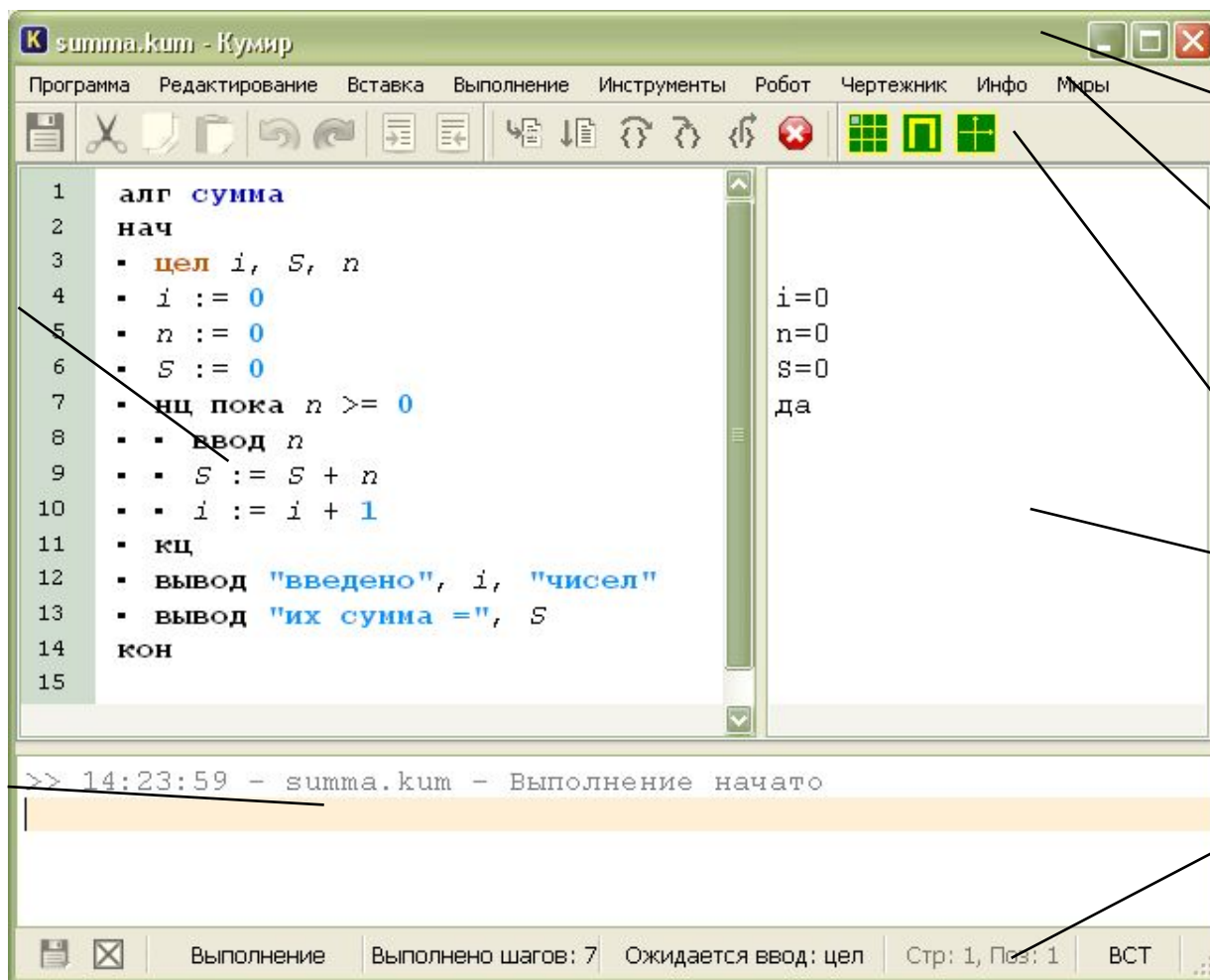
Все программы



- Водолей
- Кузнечик
- КуМир**
- КуМир (режим учителя)
- КуМир ЕГЭ
- Удалить КуМир
- Черепаха



# Внешний вид окна



Область  
программы

Область  
ввода-  
вывода

Заголовок

Главное  
меню

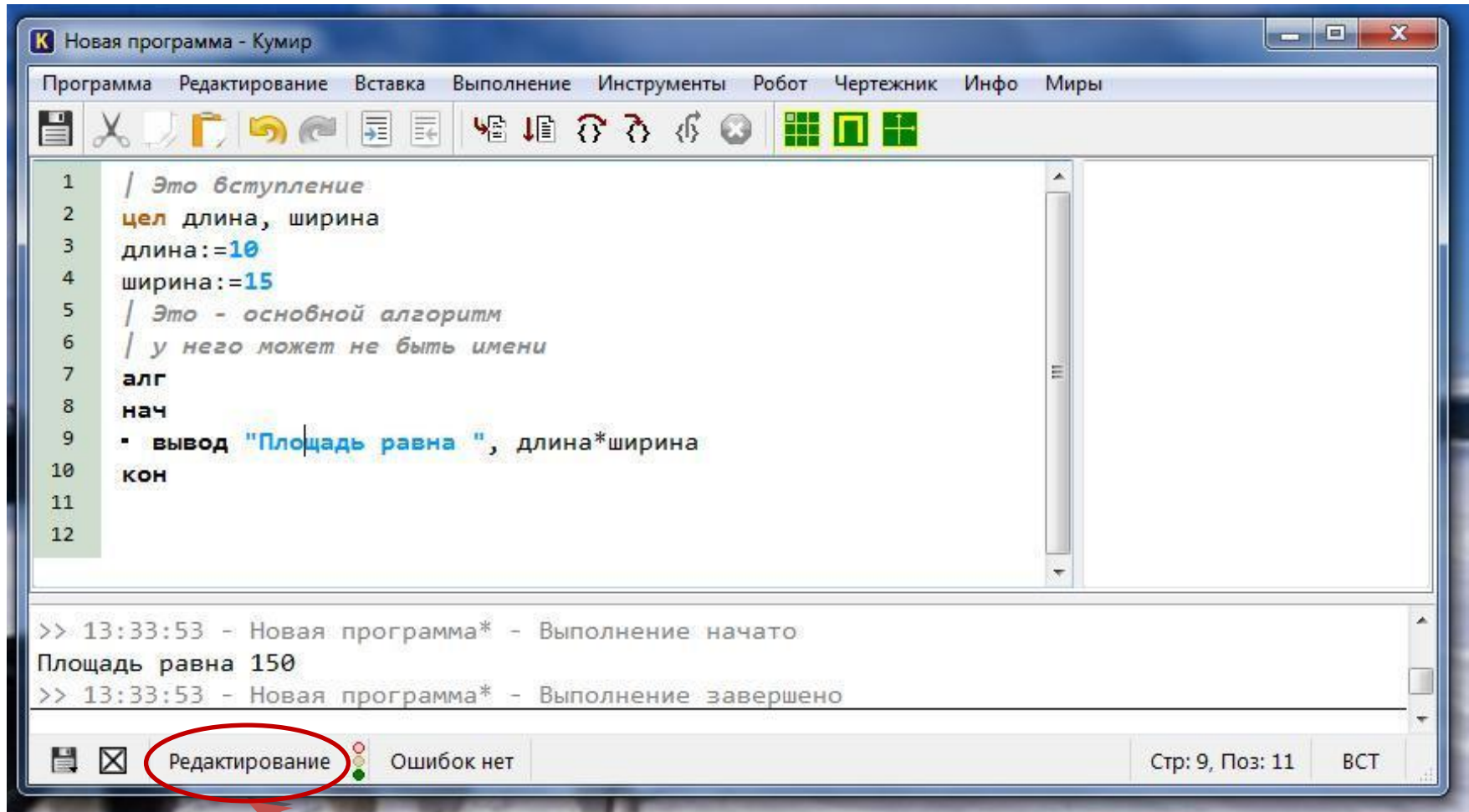
Панель  
инструментов

Область  
сообщений

Строка  
состояния



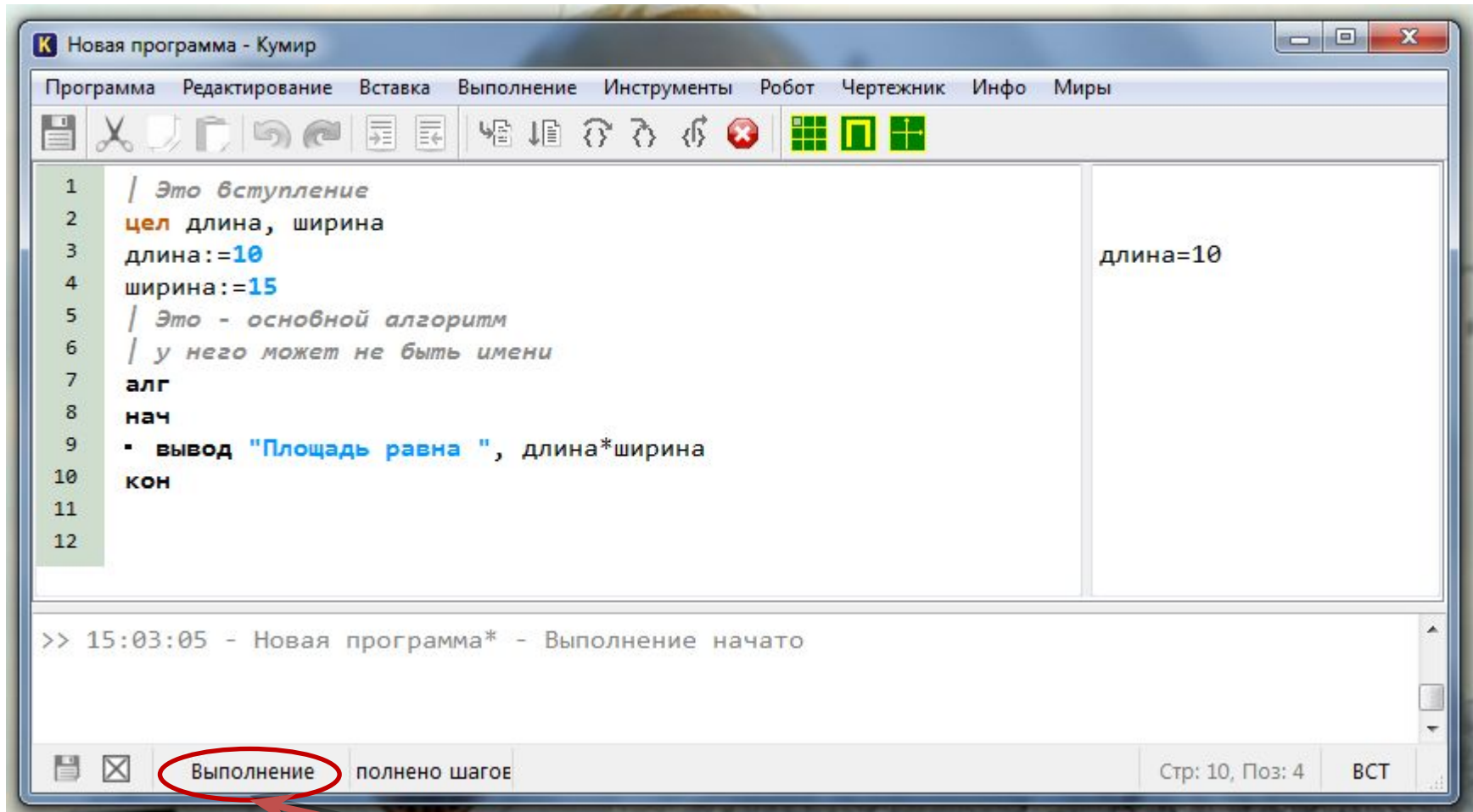
# Рабочие состояния системы Кумир



Редактирован  
ия

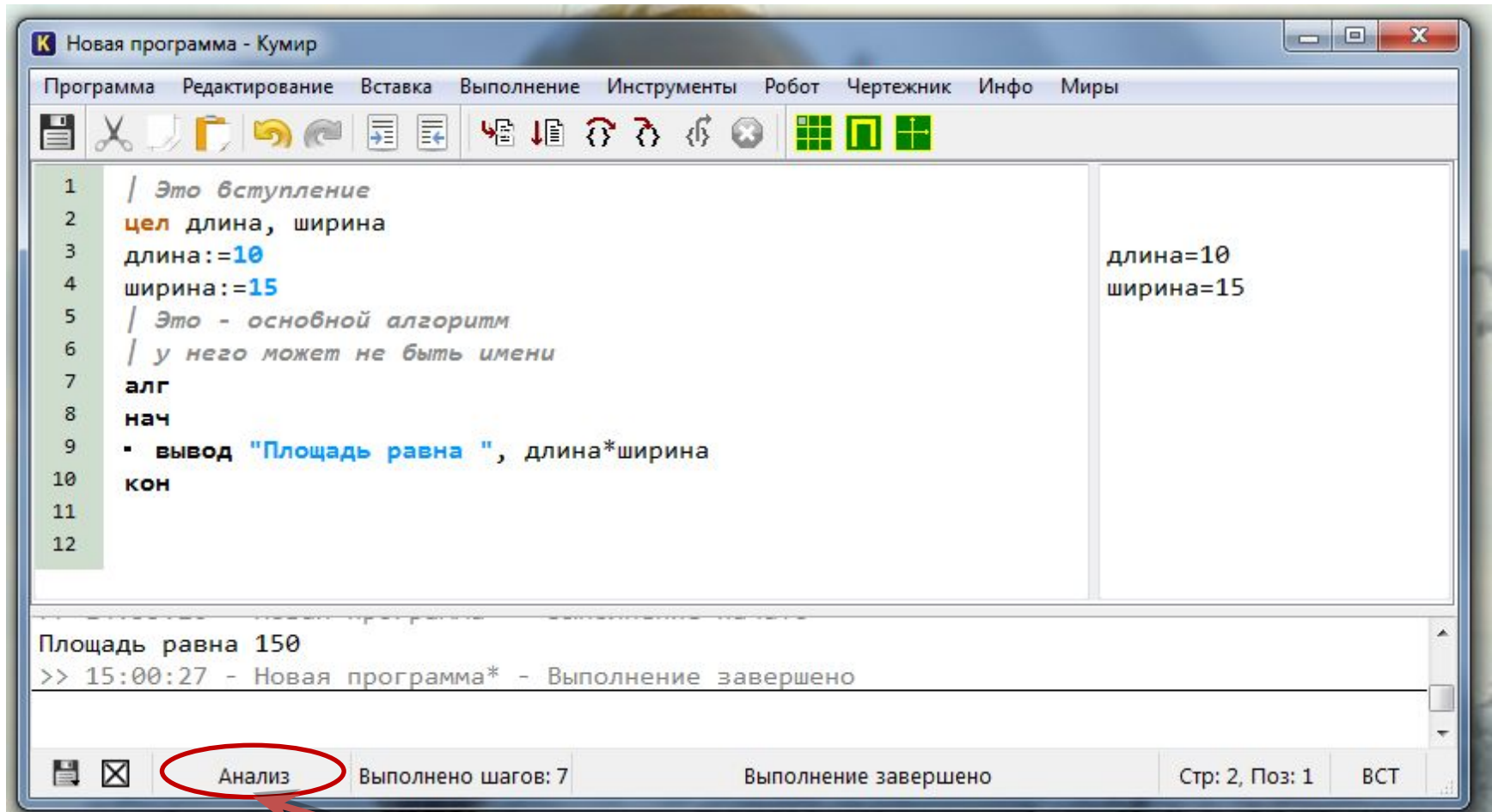


# Рабочие состояния системы Кумир



Выполнения

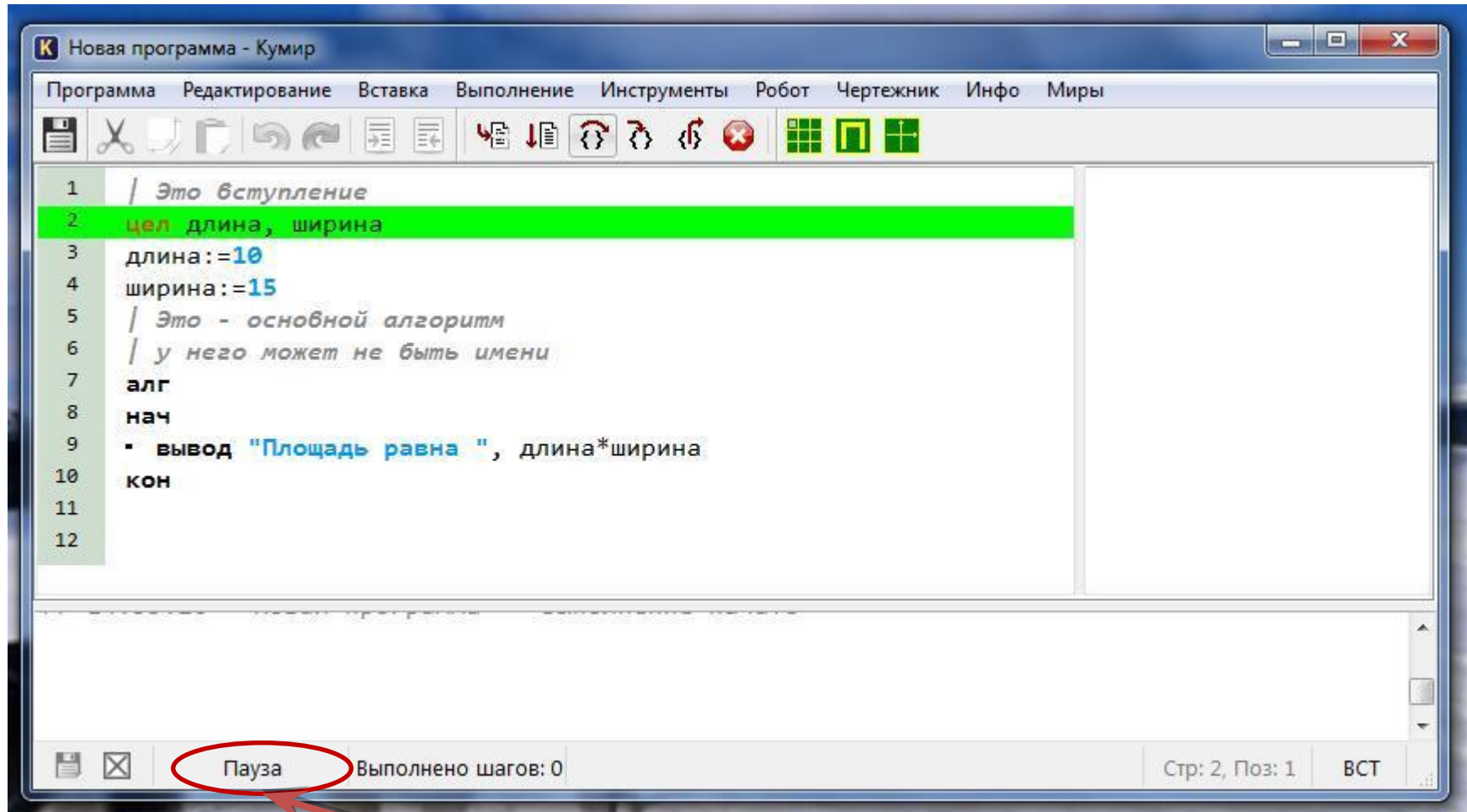
# Рабочие состояния системы Кумир



Анализ

а

# Рабочие состояния системы Кумир



Паузы



# Главное меню системы КуМир

Команды меню

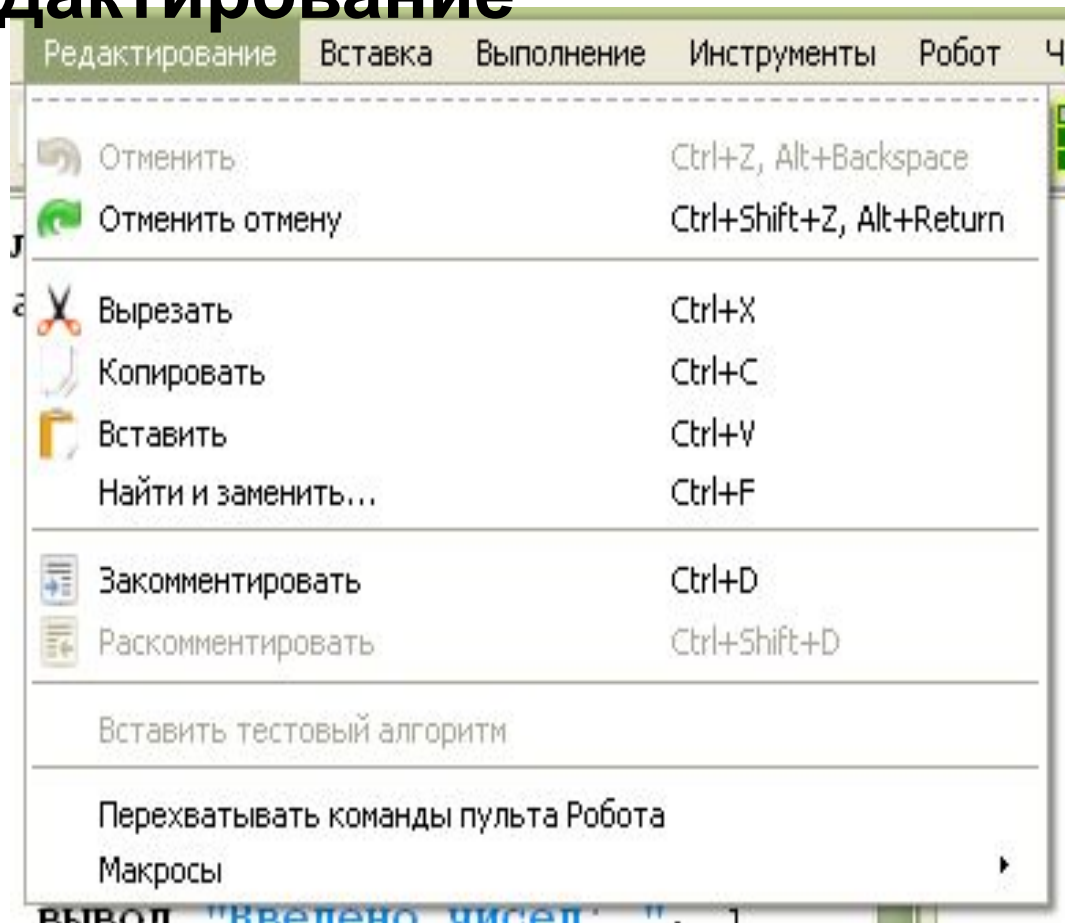
**Программа**



# Главное меню системы КуМир

Команды меню

**Редактирование**



# Главное меню системы КуМир

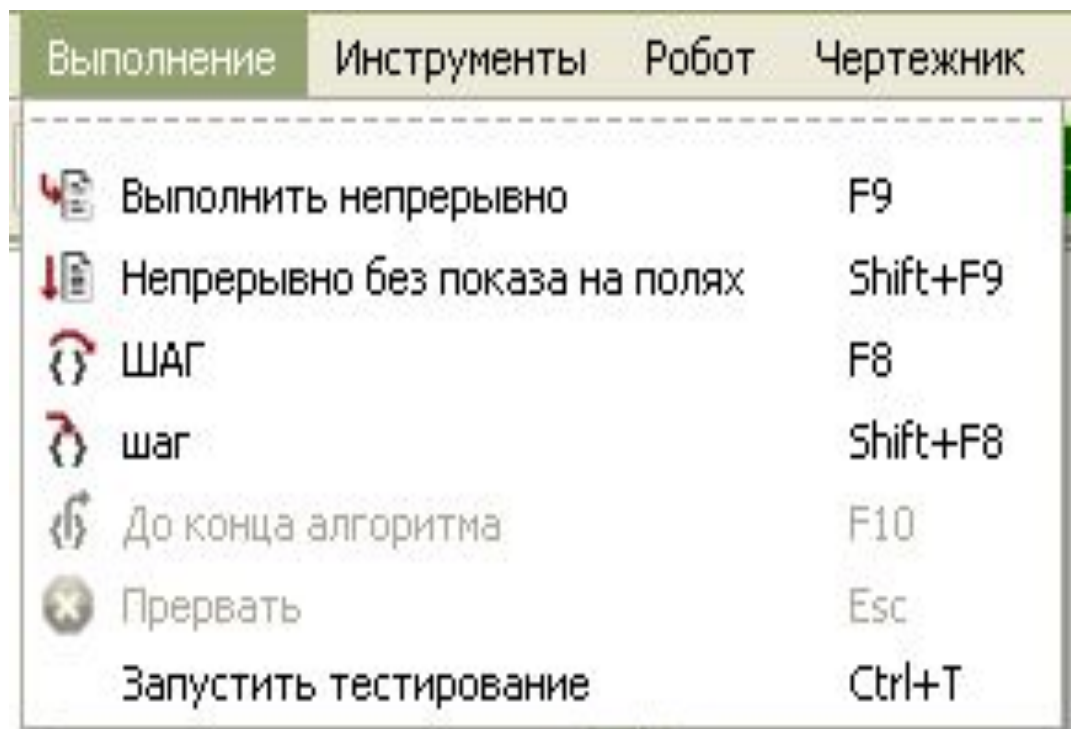
## Команды меню Вставка

Вставка	Выполнение	Инструменты	Робо
алг-нач-кон	(ESC, A)	Esc, F	
если-то-все	(ESC, E)	Esc, T	
выбор-при-все	(ESC, B)	Esc, D	
иначе	(ESC, И)	Esc, B	
нц-раз-кц	(ESC, P)	Esc, H	
нц-для-кц	(ESC, Д)	Esc, L	
нц-пока-кц	(ESC, П)	Esc, G	
нц-кц	(ESC, Ц)	Esc, W	
исп-кон_исп	(ESC, C)	Esc, C	
вверх		Esc, Up	
вправо		Esc, Right	
вниз		Esc, Down	
влево		Esc, Left	
закрасить		Esc, Space	
использовать Робот		Esc, 1	
использовать Чертежник		Esc, 2	
использовать Файлы		Esc, 3	
использовать Кузнечик		Esc, 4	
использовать Черепаха		Esc, 5	
использовать Водолей		Esc, 6	

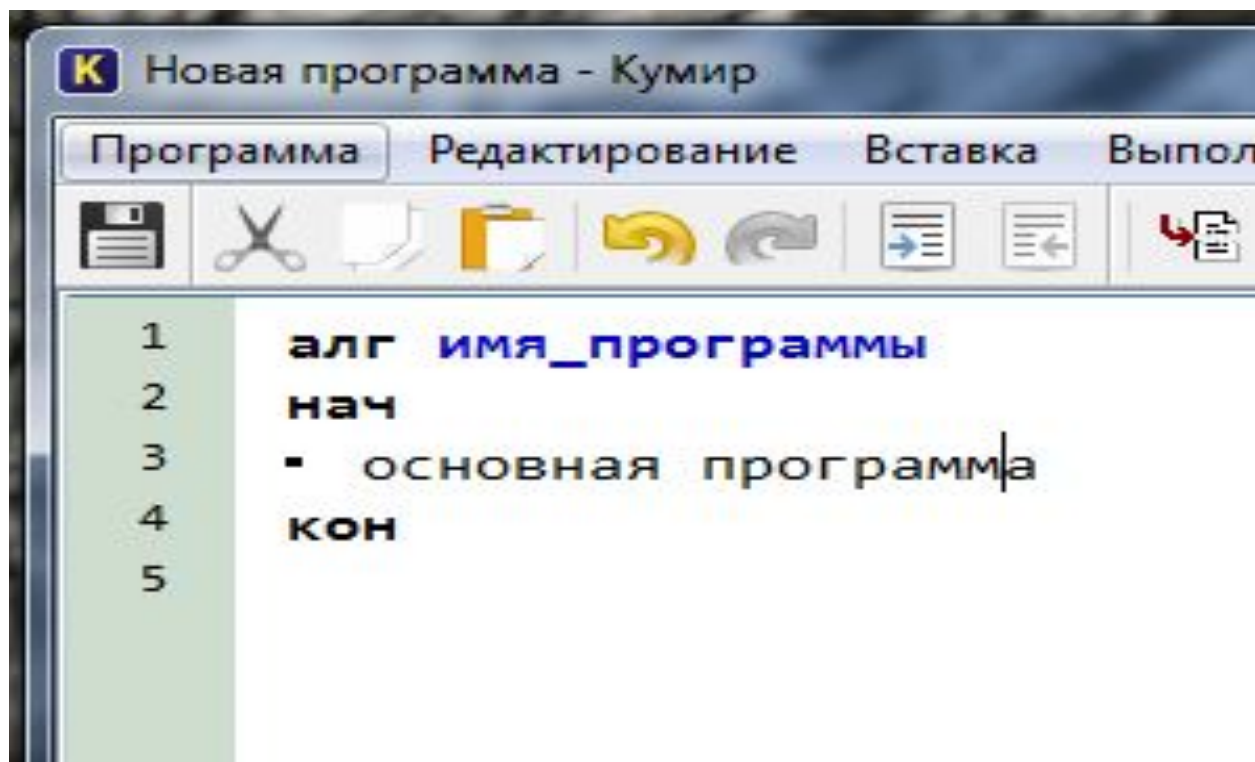


# Главное меню системы КуМир

Команды меню **Выполнение**



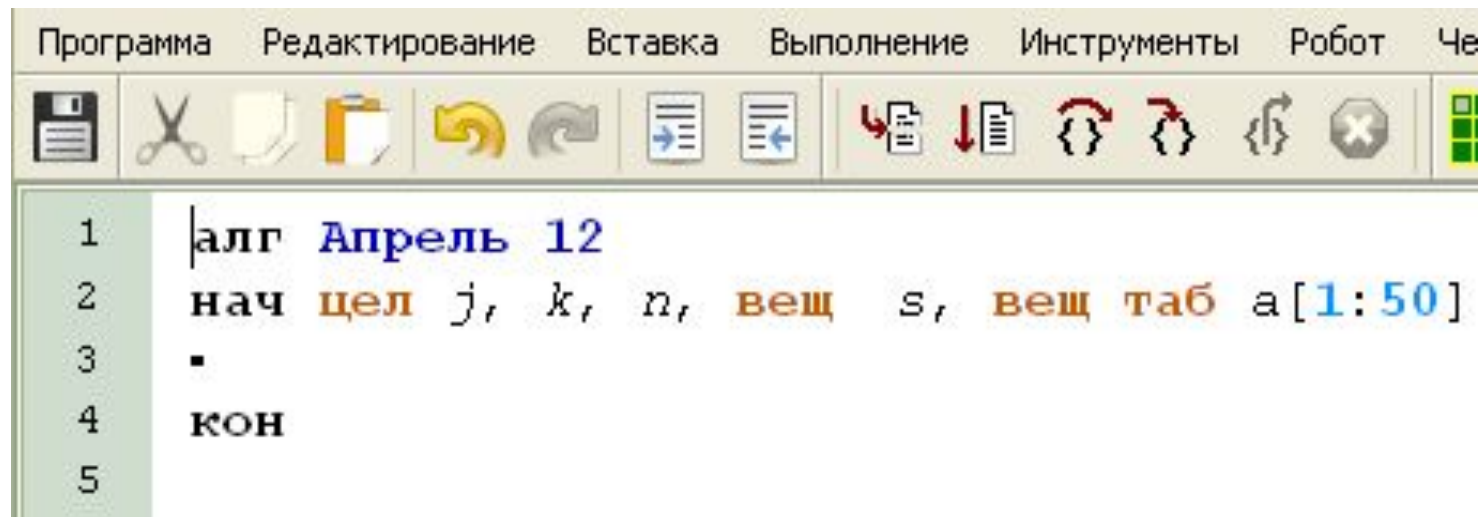
# Общая структура программы (без параметров) в системе КуМир





# Типы величин

Базовые типы	Типы для таблиц
целые числа <b>цел</b>	целые числа <b>цел таб</b>
вещественные числа <b>вещ</b>	вещественные числа <b>вещ таб</b>
логические <b>лог</b>	логические <b>лог таб</b>
символьные <b>сим</b>	символьные <b>сим таб</b>
литерные <b>лит</b>	литерные <b>лит таб</b>



```
1 | алг Апрель 12
2 | нач цел j, k, n, вещ s, вещ таб a[1:50]
3 | .
4 | кон
5 |
```



# Арифметические действия и стандартные функции

Название операции или функции	Название операции или функции
сложение	$x + y$
вычитание	$x - y$
умножение	$x * y$
деление	$x / y$

Название операции или функции	Название операции или функции
возведение в степень	$x ** y$
корень квадратный	$\text{sqrt}(x)$
абсолютная величина	$\text{abs}(x)$ и $\text{iabs}(x)$
знак числа	$(-1, 0 \text{ или } 1) \text{ sign}(x)$



# Арифметические действия и стандартные функции

Название операции или функции	Название операции или функции
синус	$\sin(x)$
косинус	$\cos(x)$
тангенс	$\operatorname{tg}(x)$
котангенс	$\operatorname{ctg}(x)$
арксинус	$\arcsin(x)$
арккосинус	$\arccos(x)$
арктангенс	$\operatorname{arctg}(x)$
арккотангенс	$\operatorname{arcctg}(x)$





# Операции сравнения

Операция	Форма записи
Равно	$a = b$
не равно	$a < > b$
меньше	$c < b+3$
больше	$c > 1$
меньше или равно	$k \leq f$
больше или равно	$k \geq f$





# Логические операции

Операция	Форма записи
конъюнкция	$a \text{ и } b$
дизъюнкция	$a \text{ или } b$
отрицание	$\text{не } a$



# Приоритеты логических операций

- 1) выражения в скобках
- 2)  $=$ ,  $<$ ,  $<=$ ,  $>$ ,  $>=$ ,  $<>$
- 3) не
- 4) и
- 5) или



# Операции ввода-вывода

```
1  алг Апрель 12
2  нач цел j, k, n, вещ s, вещ таб a[1:50]
3  ▪ ввод k
4  ▪ s := arctg(k)
5  ▪ вывод s
```



# Этапы решения задач

- Условие задачи
- Постановка задачи
- Математическая формализация
- Алгоритм
- Блок-схема
- Программа
- Компьютерный эксперимент

# **Простейшая программа (линейный алгоритм)**



# (ГИА задание №10 и №12)

В алгоритме, записанном ниже, используются действительные переменные  $a$  и  $b$ .

Определите значение переменной  $b$  после выполнения следующего фрагмента алгоритма:

$a := -5;$

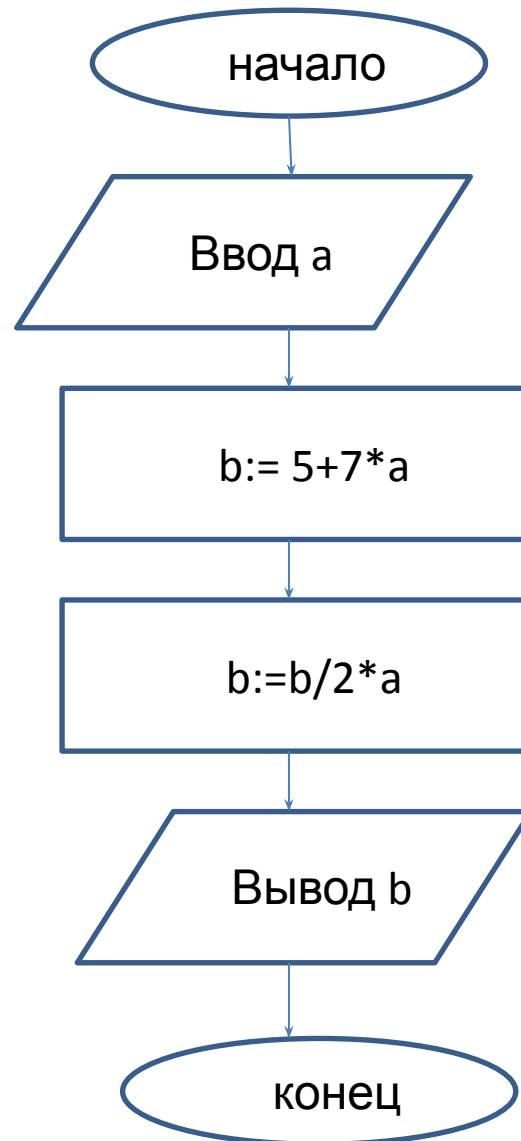
$b := 5 + 7 * a;$

$b := b / 2 * a;$

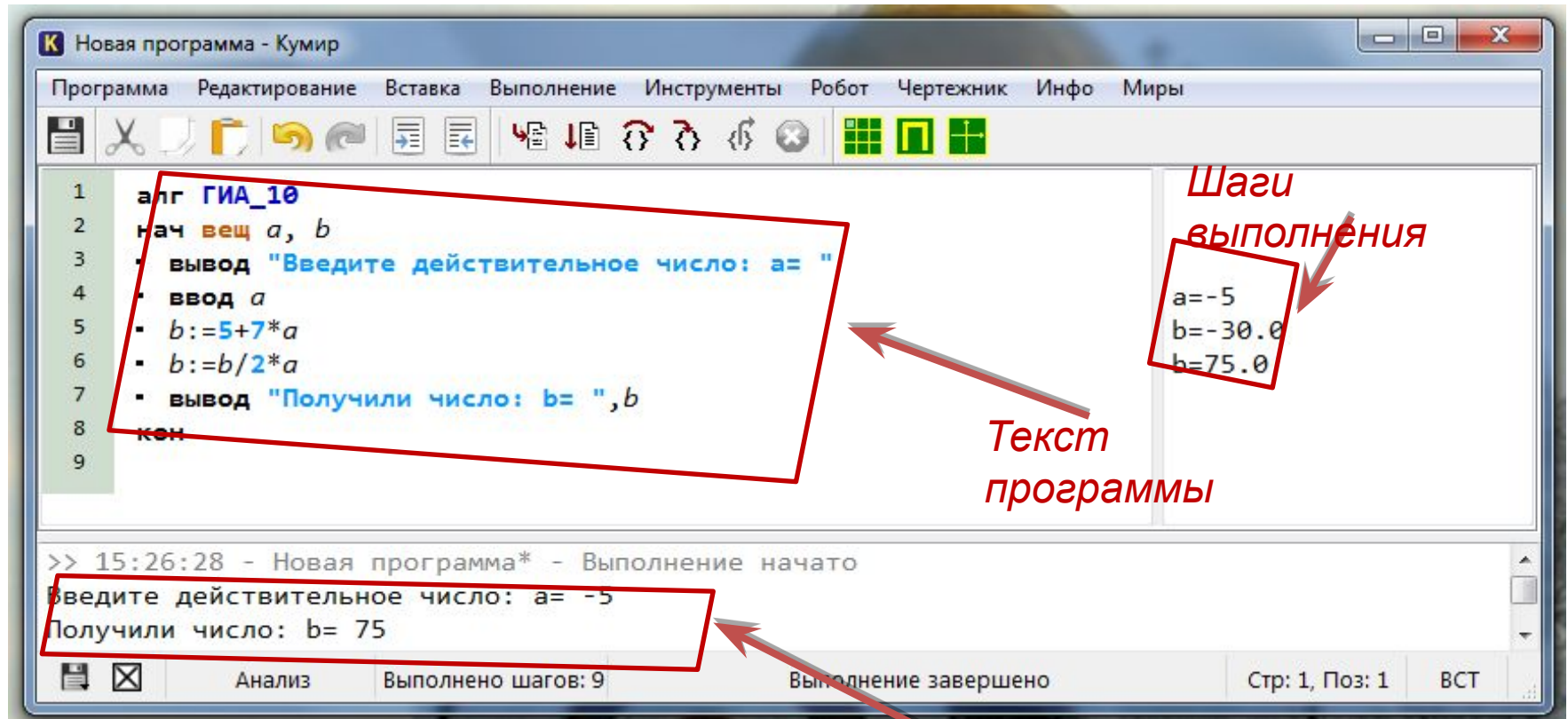
Порядок действий соответствует правилам арифметики. В ответе укажите одно число – значение переменной  $b$ .



# Линейный алгоритм (блок-схема)



# Простейшая программа (линейный алгоритм) (ГИА задание №10)



Результат  
выполнения  
программы



# Структура «ветвление»



# Команда если – то - все или неполное ветвление

Общий вид команды:

***если*** условие  
***то*** действия  
***все***

```
1  алг ветвление
2  нач вещ x, y
3  ▪ вывод "Введите значение x= "
4  ▪ ввод x
5  ▪ вывод "Введите значение y= "
6  ▪ ввод y
7  ▪ если x>0
8  ▪   ▪ то y:=y+5
9  ▪   ▪ все
10 ▪ вывод "Новое значение y= ", y
11 кон
12
```

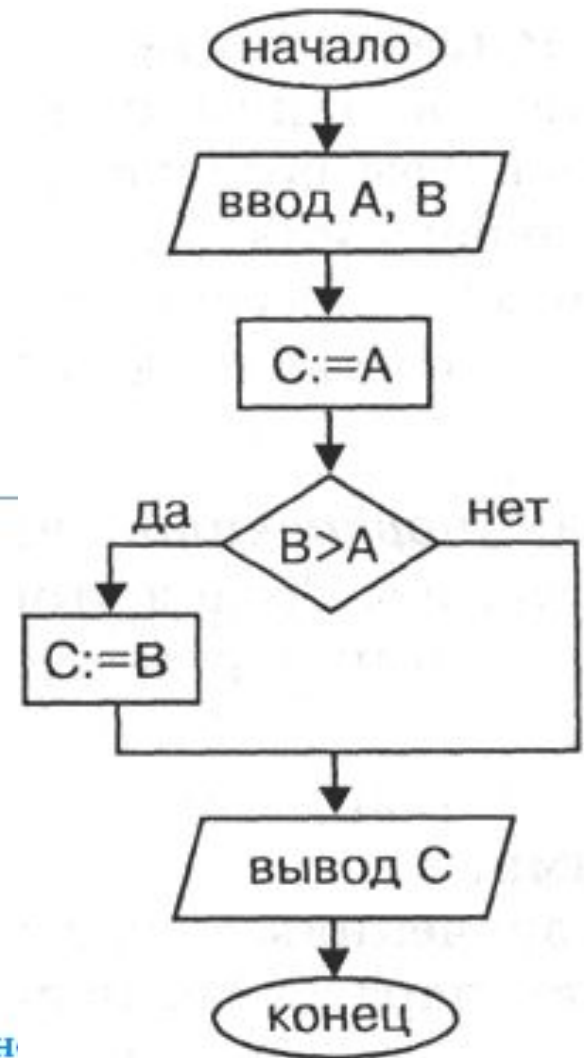


# Пример 1

(неполное ветвление)

Даны два числа; выбрать  
большее из них.

```
1  алг неполное ветвление
2  нач вещ a, b, c
3  ▪ вывод "Введите значение числа a= "
4  ▪ ввод a
5  ▪ вывод "Введите значение числа b= "
6  ▪ ввод b
7  ▪ если  $a > b$ 
8  ▪   то
9  ▪      $c := a$ 
10 ▪   все
11 ▪   вывод "Наибольшее из двух данных чисел равно c"
12 кон
13
```



# Команда если – то – иначе - все или полное ветвление

Общий вид команды:

**если** условие

**то** действия 1

**иначе** действия 2

**все**

```
1  алг ветвление
2  нач вещ x, y
3  ▪ вывод "Введите значение x= "
4  ▪ ввод x
5  ▪ вывод "Введите значение y= "
6  ▪ ввод y
7  ▪ если x>0
8  ▪ ▪ то y:=y+5
9  ▪ ▪ иначе y:=y-2
10 ▪ все|
11 ▪ вывод "Новое значение y= ", y
12 кон
13
```

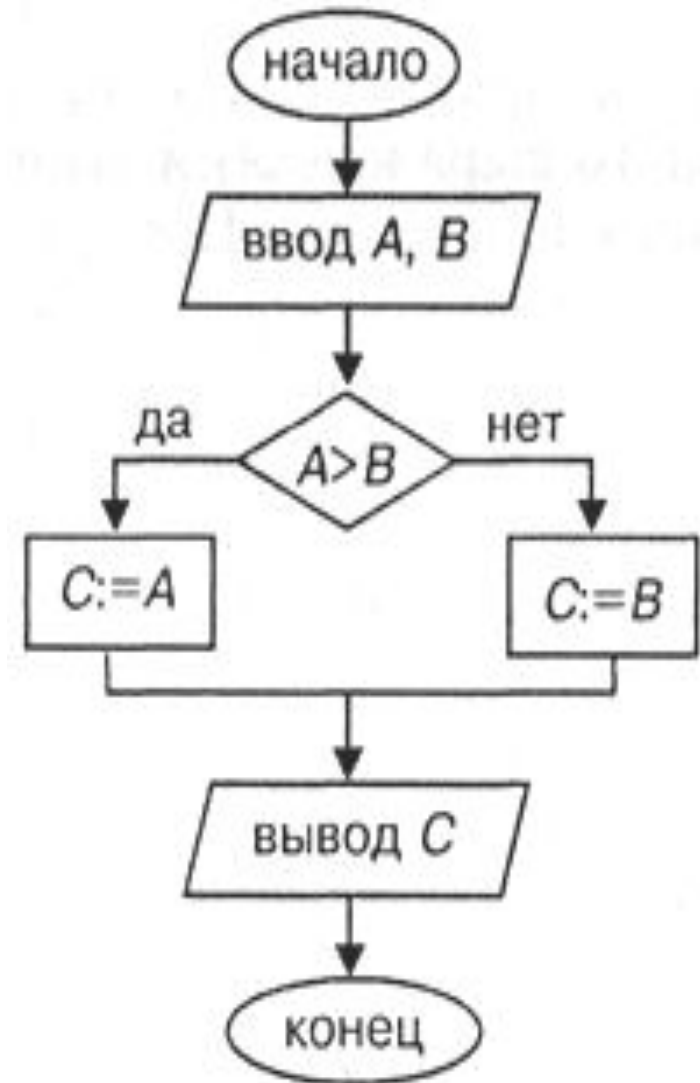


# Пример 2

(полное ветвление)

Даны два числа; выбрать  
большее из них.

```
1  алг неполное ветвление  
2  нач вещ  $a, b, c$   
3  ▪ вывод "Введите значение числа  $a =$  "  
4  ▪ ввод  $a$   
5  ▪ вывод "Введите значение числа  $b =$  "  
6  ▪ ввод  $b$   
7  ▪ если  $a > b$   
8  ▪   ▪ то  
9  ▪     ▪  $c := a$   
10 ▪   ▪ иначе  
11 ▪     ▪  $c := b$   
12 ▪ все  
13 ▪ вывод "Наибольшее из двух данных чисел равно ",  $c$   
14 кон
```





Условные  
выражени  
я

```
graph TD; A[Условные выражения] --> B[Простые]; A --> C[Сложные];
```

Простые

Сложные



# Условные выражения

- Простое условие включает в себя два числа, две переменных или два арифметических выражения, которые сравниваются между собой с использованием операций сравнения ( $=$ ,  $>$ ,  $<$ ).

*Например:  $7 > 5$ ,  $2 * 8 = 4 * 4$  и т.д.*

# Условные выражения

- **Сложное условие** — это последовательность простых условий, объединенных между собой знаками логических операций (и, или и др.).

*Например:  $(7 > 5)$  и  $(2 * 8 = 4 * 4)$*



# Сложные ветвления

```
если условие 1  
    то действие 1  
    иначе  
        если условие 2  
            то действие 2  
            иначе действие 3  
        все  
    все
```

```
если условие 1  
    то  
        если условие 2  
            то действие 2  
            иначе действие 3  
        все  
    все
```

Оператор **ИНАЧЕ** всегда относится к ближайшему **ЕСЛИ**

# Команда выбор (неполная форма)

## Общий вид команды:

### *выбор*

- *при* условие 1: действия 1
  - *при* условие 2: действия 2
  - ...
  - *при* условие n: действия n
- все**

```
1  алг ветвление
2  нач вещ x, y
3  ▪ вывод "Введите значение x= "
4  ▪ ввод x
5  ▪ выбор
6  ▪ ▪ при x>0: y:=x+1
7  ▪ ▪ при x<0: y:=x**2
8  ▪ все
9  ▪ вывод "Значение функции y= ", y
10 кон
11
```



# Команда выбор (полная форма)

## Общий вид

### команды:

#### *выбор*

- *при* условие 1: действия 1
  - *при* условие 2: действия 2
  - ...
  - *при* условие n: действия n
  - *иначе* действия n+1
- все**

```
1  алг ветвление
2  нач вещ x, y
3  ▪ вывод "Введите значение x= "
4  ▪ ввод x
5  ▪ выбор
6  ▪ ▪ при x>0: y:=x+1
7  ▪ ▪ при x<0: y:=x**2
8  ▪ ▪ иначе
9  ▪ ▪ ▪ y:=0
10 ▪ все
11 ▪ вывод "Значение функции y= ", y
12 кон
13
```



# **Одномерный массив. Циклические конструкции**



# Описание одномерного массива

**цел таб а[1:50]** — массив целых чисел из 50 элементов;

**вещ таб а[1:50]** — массив вещественных чисел из 50 элементов.

**нач**

. **цел** N=**20**

. **цел таб** а [**1**:N] , b [**1**:N]

. . .





```

1  | Дан целочисленный массив из N элементов.
2  | Увеличить все элементы в 2 раза.
3  алг пример_4_1
4  нач целтаб а[1:100], цел т, н
5  ▪ вывод "Введите количество элементов массива"
6  ▪ ввод н
7  ▪ вывод "введите элементы:"
8  ▪ нц для т от 1 до н
9  ▪ ▪ ввод а[т]
10 ▪ кц
11 ▪ вывод "полученный массив:"
12 ▪ нц для т от 1 до н
13 ▪ ▪ вывод а[т]*2
14 ▪ ▪ вывод нс
15 ▪ кц

```

>> 19:49:02 - Новая программа\* - Выполнение начато

Введите количество элементов массива2

введите элементы:23

43

полученный массив:46

86

# Циклические конструкции



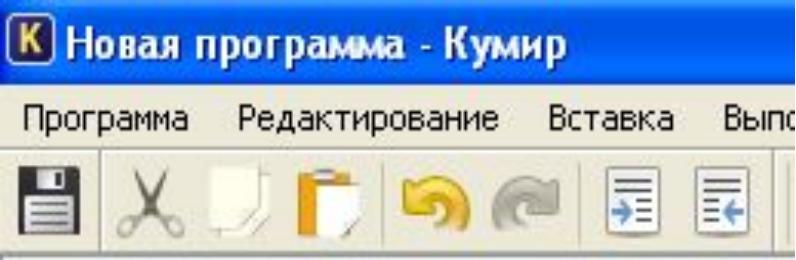
# Цикл «для»

## Общий вид:

нц для  $i$  от  $i1$  до  $i2$

тело цикла  
(последовательность  
команд)

кц



```
1  алг цикл
2  нач цел а, b, i
3  ▪ цел таб t1[1:5]
4  ▪ а:=1
5  ▪ b:=5
6  ▪ нц для i от а до b
7  ▪ ▪ t1[i]:=a*i+b
8  ▪ кц
9  кон
10
```

Где  $i$ ,  $i1$ ,  $i2$  — переменные целого типа.

Параметр цикла  $i$  изменяется от  $i1$  до  $i2$  с шагом 1. Должно выполняться неравенство  $i1 < i2$ , если же  $i1 = i2$ , то тело цикла выполнится один раз, если  $i1 > i2$ , то тело цикла не выполнится ни разу.

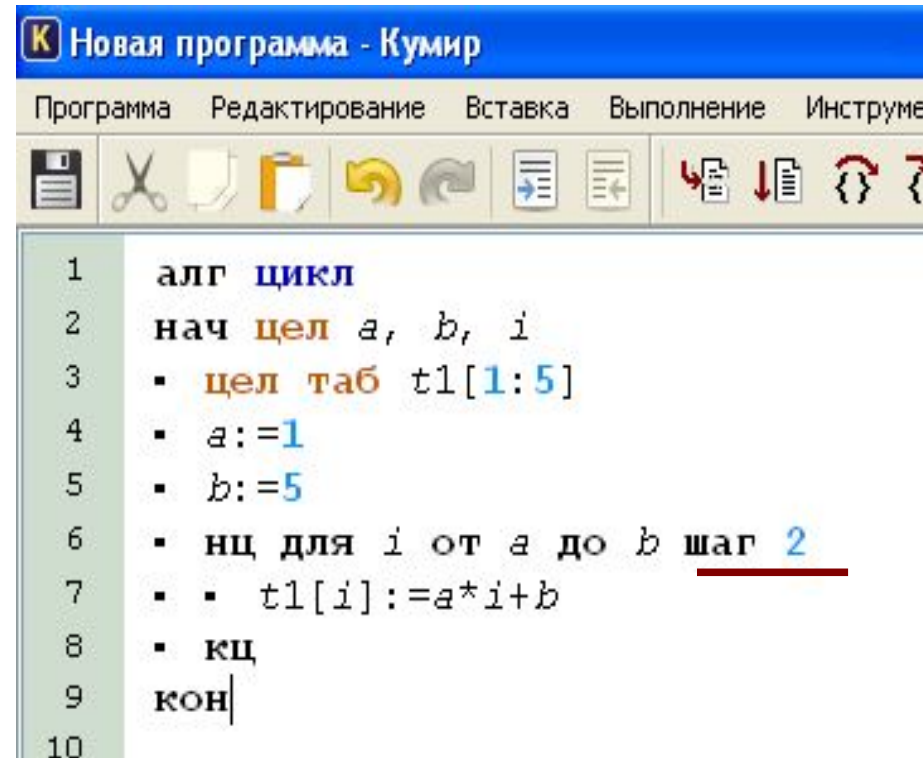


# Цикл «для» (общий вид с шагом)

нц для  $i$  от  $i_1$  до  $i_2$  шаг  $i_3$

тело цикла  
(последовательность  
команд)

кц



```
1  алг цикл
2  нач цел а, b, i
3  ▪ цел таб t1[1:5]
4  ▪ а:=1
5  ▪ b:=5
6  ▪ нц для i от а до b шаг 2
7  ▪ ▪ t1[i]:=a*i+b
8  ▪ кц
9  кон
10
```

- Если шаг  $i_3 > 0$ , то тело цикла выполняется до тех пор, пока  $i \leq i_2$  с условием, что  $i_1 \leq i_2$ .
- Если шаг  $i_3 < 0$ , то тело цикла выполняется до тех пор, пока  $i \geq i_1$ , с условием, что  $i_1 \geq i_2$ .



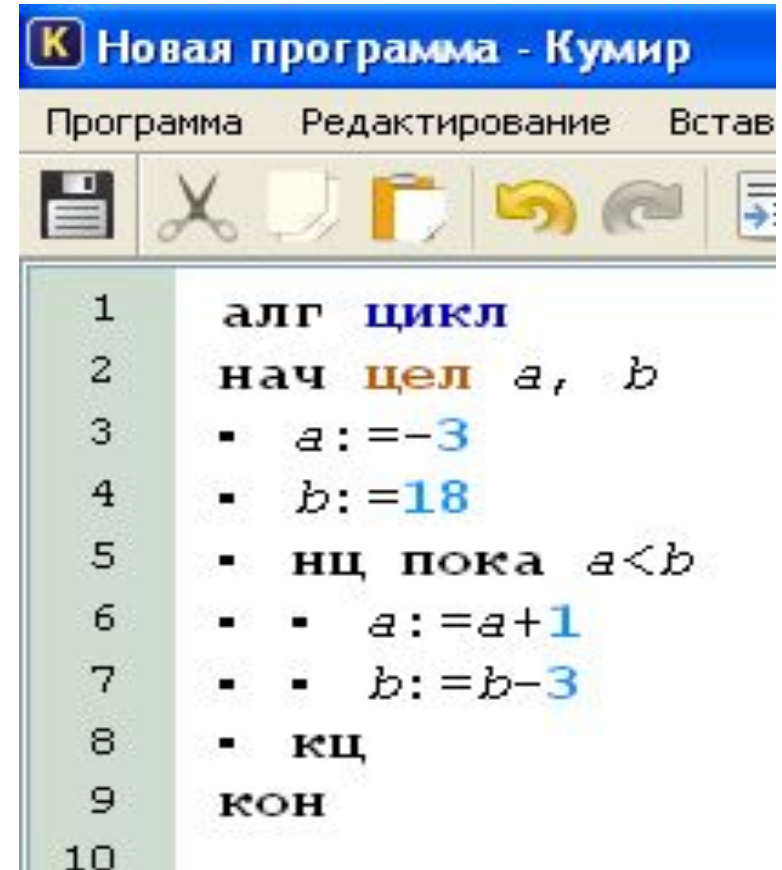
# Цикл «пока»

## Общий вид:

**нц пока условие**

тело цикла  
(последовательность  
команд)

**кц**

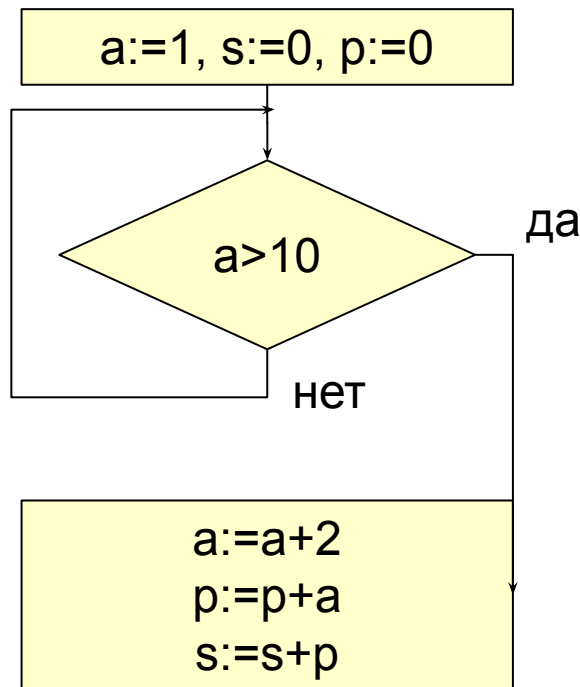


```
1  алг цикл
2  нач цел a, b
3  ▪ a := -3
4  ▪ b := 18
5  ▪ нц пока a < b
6  ▪ ▪ a := a + 1
7  ▪ ▪ b := b - 3
8  ▪ кц
9  кон
10
```

Тело цикла выполняется до тех пор, *пока* выполняется условие



# Блок-схема для цикла «пока»



Определите значение  
переменной  $s$  после  
выполнения  
фрагмента  
алгоритма,  
записанного в виде  
блок-схемы

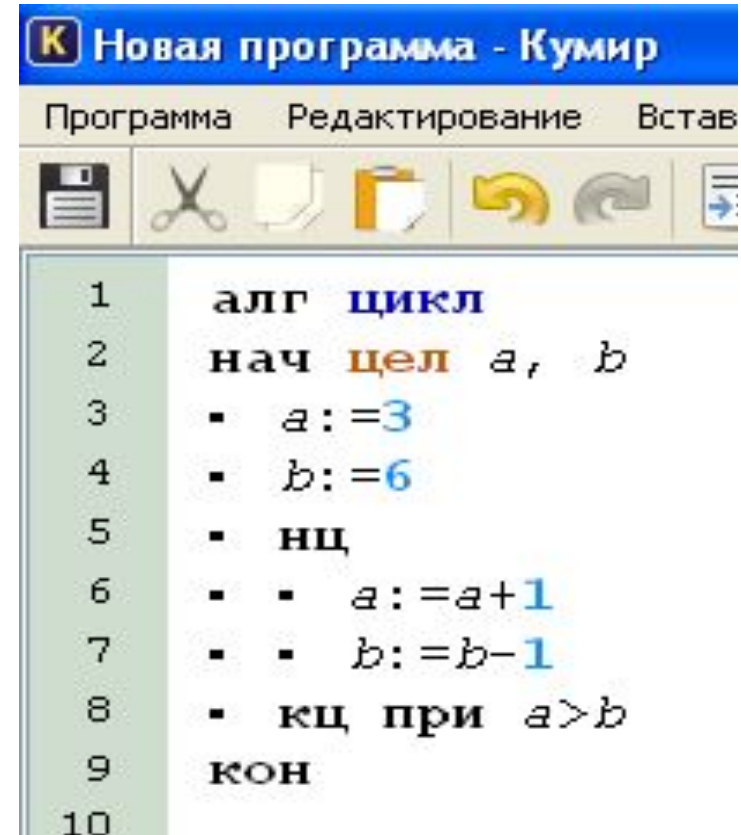
# Цикл «кц\_при»

## Общий вид:

**нц**

тело цикла  
(последовательность  
команд)

**кц при условие**

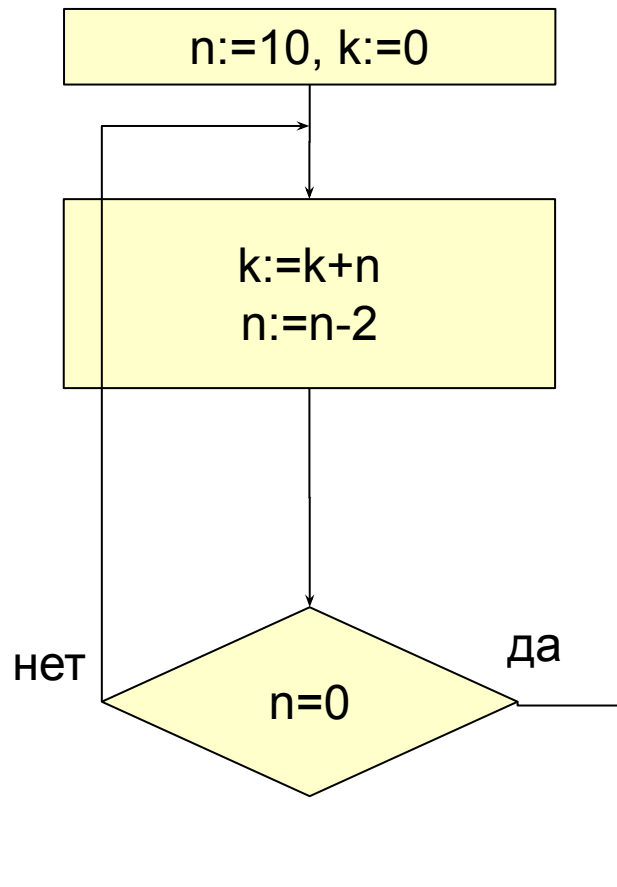


```
1  алг цикл
2  нач цел a, b
3  ▪ a:=3
4  ▪ b:=6
5  ▪ нц
6  ▪ ▪ a:=a+1
7  ▪ ▪ b:=b-1
8  ▪ кц при a>b
9  кон
10
```

Тело цикла выполняется при  
выполнении условия



# Блок-схема для цикла «кц\_при»



Определите значение  
переменной  $k$  после  
выполнения  
фрагмента  
алгоритма,  
записанного в виде  
блок-схемы



# Обработка двумерных массивов

цел таб  $a[1:n1, 1:n2]$ ,  
вещ таб  $b['0': '9', '0': '9']$



# Литерные величины



# Типы литерных величин

- **сим** – один любой символ (кроме специальных символов КуМира);
- **лит** – строка литеральных символов



# Команды обработки литерных величин

Команда	Форма записи
Длина строки	<b>длин</b> (стр)
Код символа в таблице КОИ-8	<b>код</b> (с)
Символ таблицы КОИ-8	<b>символ</b> (х)
Код символа в таблице Юникод	<b>юникод</b> (с)
Символ таблицы Юникод	<b>символ2</b> (х)
Строковое представление целого числа	<b>цел_в_лит</b> (х)



# Команды обработки литерных величин

Команда	Форма записи
Перевод строки в целое число	<b>лит_в_цел</b> (с, успех) – если перевод прошел успешно, то в переменную <b>успех</b> записывается «да», иначе – записывается «нет»
Перевод строки в вещественное число	<b>лит_в_вещ</b> (с, успех) – если перевод прошел успешно, то в переменную <b>успех</b> записывается «да», иначе – записывается «нет»
Вырезка подстроки (части строки)	<b>a[n:m]</b> – вырезает из строки <b>a</b> символы с <b>n</b> -го по <b>m</b> -ный включительно
Слияние	<b>a+b</b>
Извлечение символа	<b>a[3]</b>



# Исполнитель Строки

Команда	Форма записи
Удаление части строки	<b>удалить</b> (a, k,n) a - откуда удаляем (строка), k – начальная позиция удаляемой части подстроки, n – количество удаляемых символов, начиная с k-го
Вставка в строку	<b>вставить</b> (с, a, k) с – вставляемая подстрока, a – строка, в которую необходимо вставить, k – позиция первого символа вставляемой строки
Поиск символа (или подстроки)	<b>найти</b> (x, a) x – строка поиска, a – где ищем Находит позицию <b>только первого</b> символа искомой подстроки. Если подстроки нет, то возвращает значение <b>-1</b>

