Алгоритмический язык КуМир

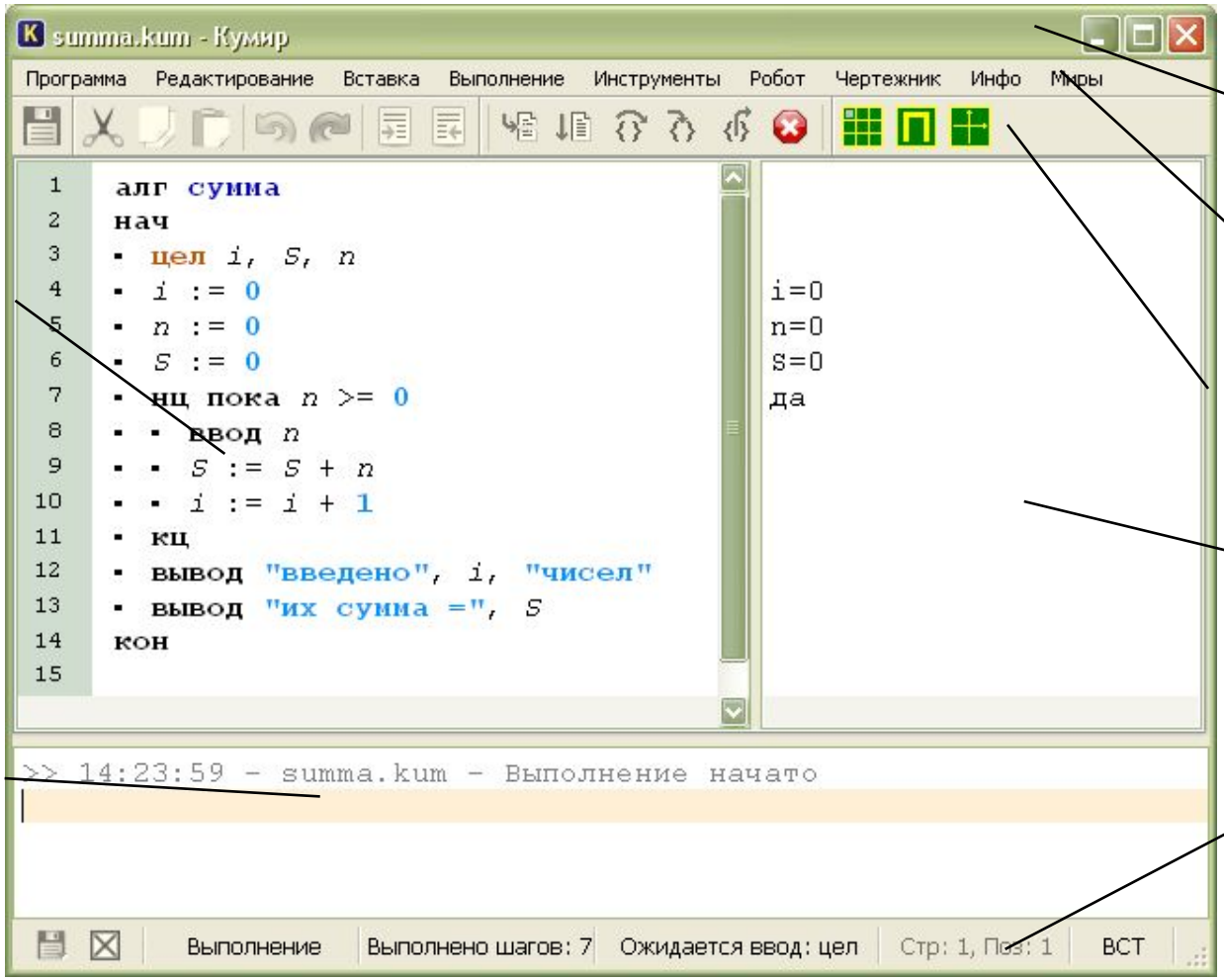
Мазеева Татьяна Александровна, учитель информатики
МКОУ «СОШ №3» г. Николаевска Волгоградской обл.
2011г.

Содержание

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

Корзина
Мои документы
классы
Adobe Reader X
Мой компьютер
avast! Free
user
Интернет
Mozilla Firefox
Электронная почта
Outlook Express
Games
Проигрыватель Windows Media
Windows Messenger
Знакомство с Windows
Мастер переноса файлов и параметров
HP Photosmart Essential
Интернет Цензор
Все программы
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
Проигрыватель Windows Media
Удаленный помощник
НИИСИ РАН
Комплект Учебных МИРов
Водолей
Кузнечик
Кумир
Кумир (режим учителя)
Кумир ЕГЭ
Удалить КуМир
Черепаха
Выход из системы
Выключение
пуск
RU
12:54

Внешний вид окна



Заголовок

Главное меню

Панель инструментов

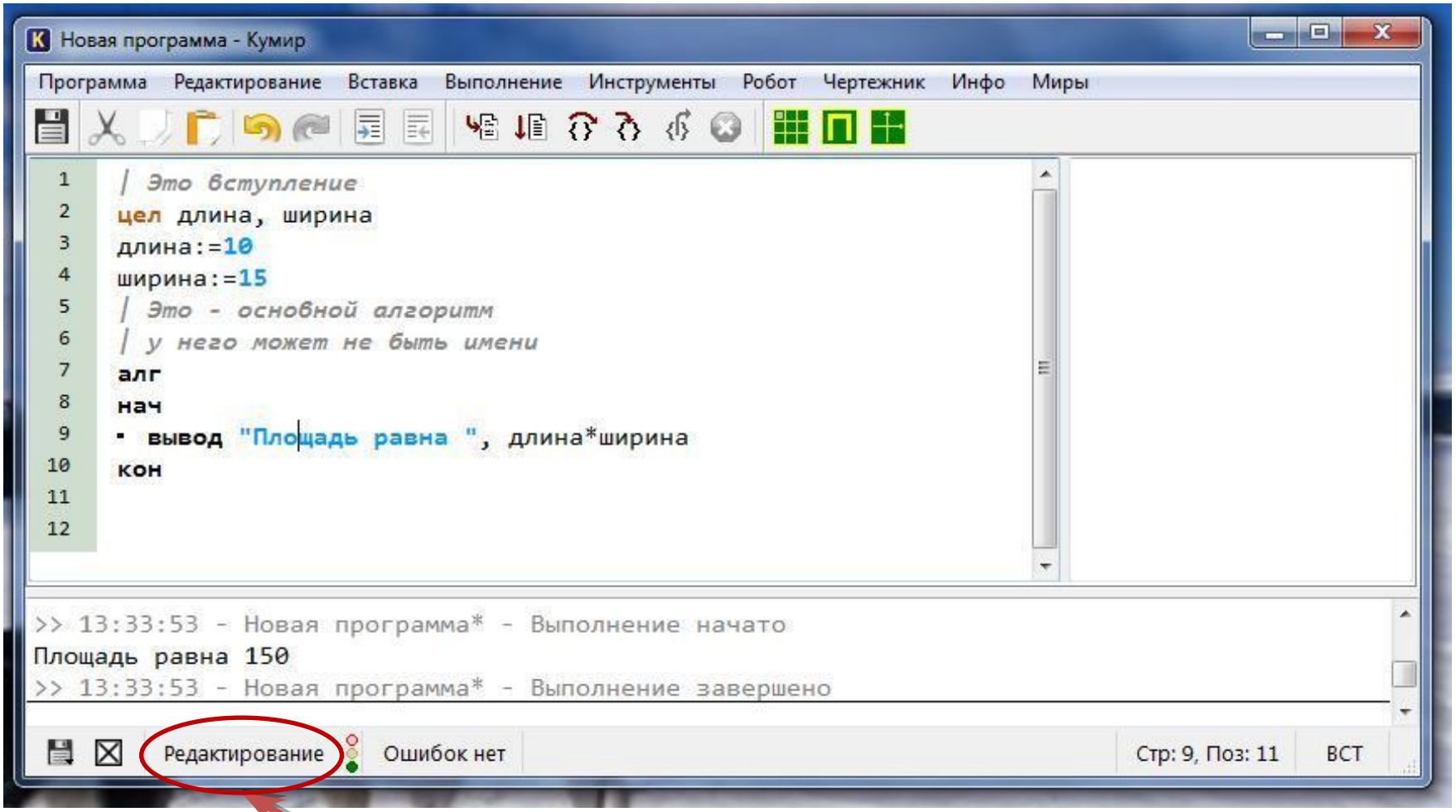
Область сообщений

Строка состояния

Область программы

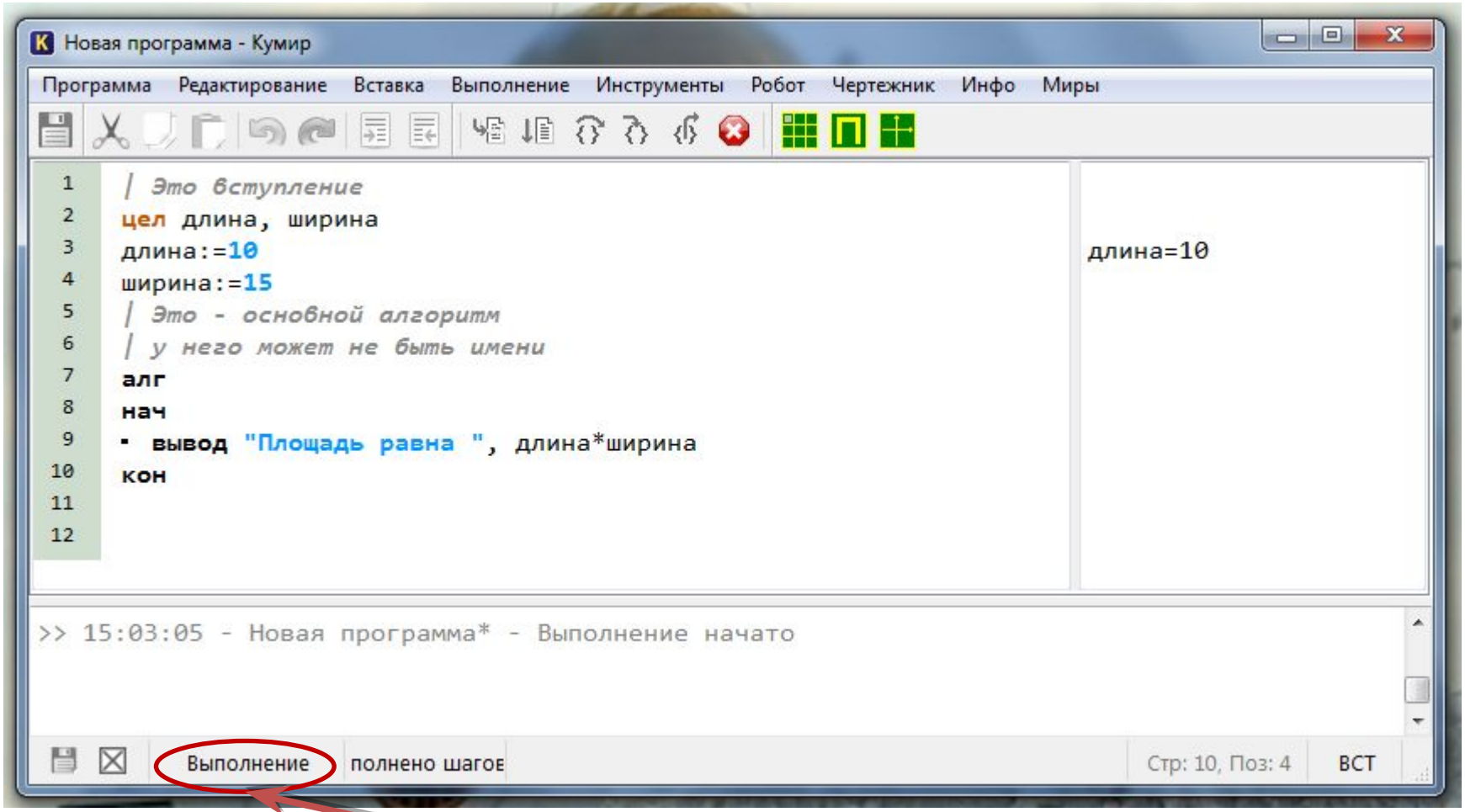
Область ввода-вывода

Рабочие состояния системы Кумир



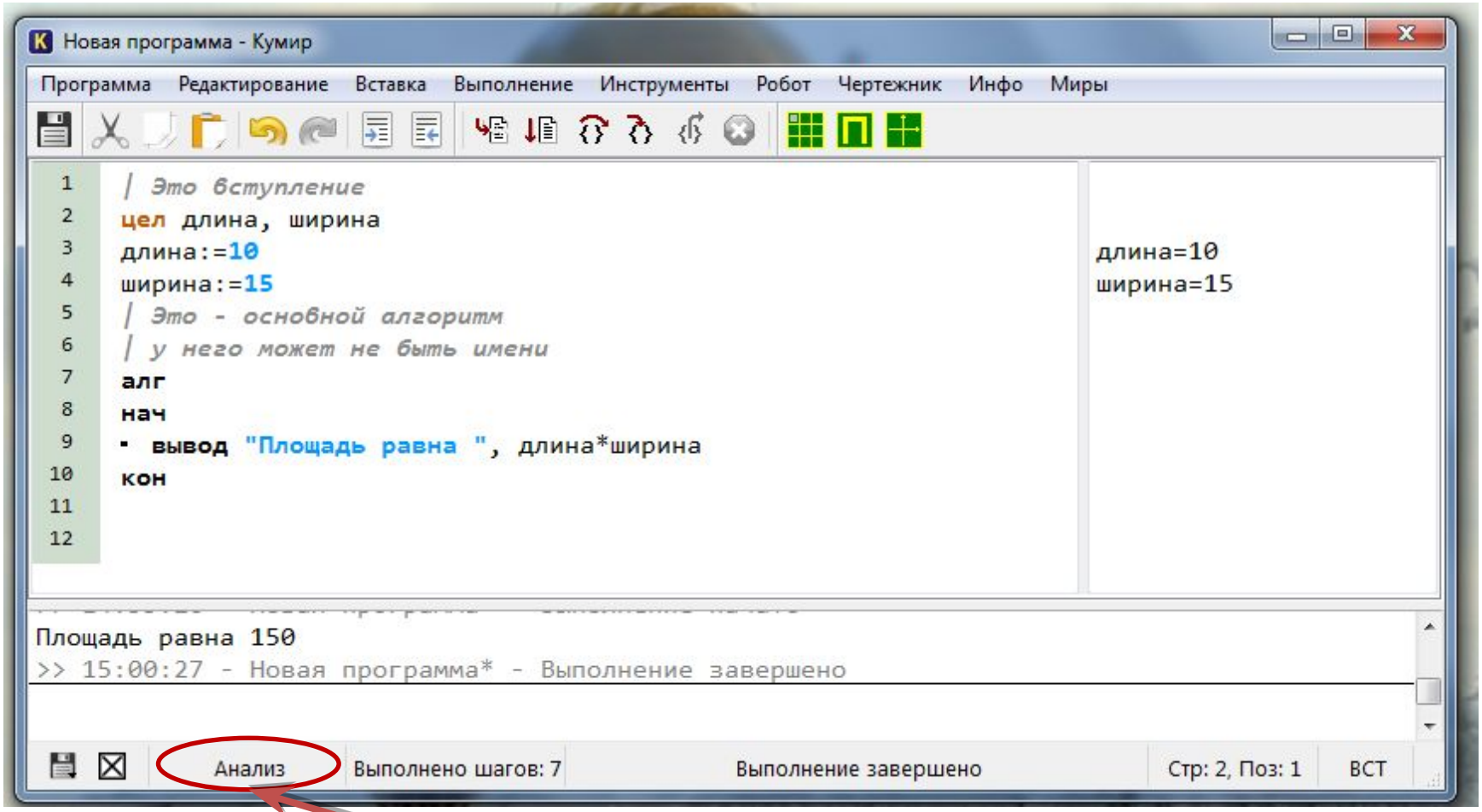
Редактирован
ия

Рабочие состояния системы Кумир



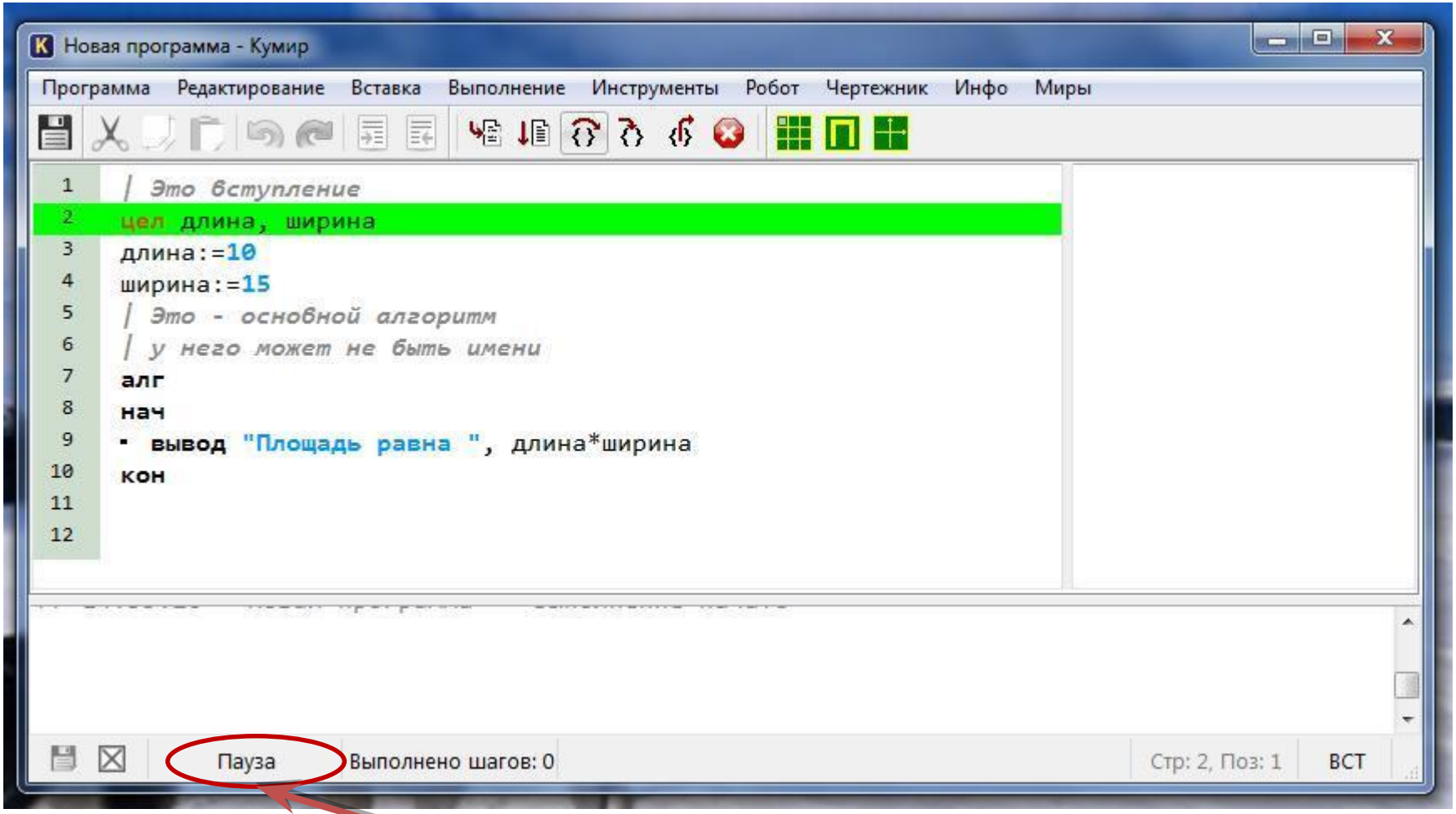
Выполнения

Рабочие состояния системы Кумир



Анализ
а

Рабочие состояния системы Кумир



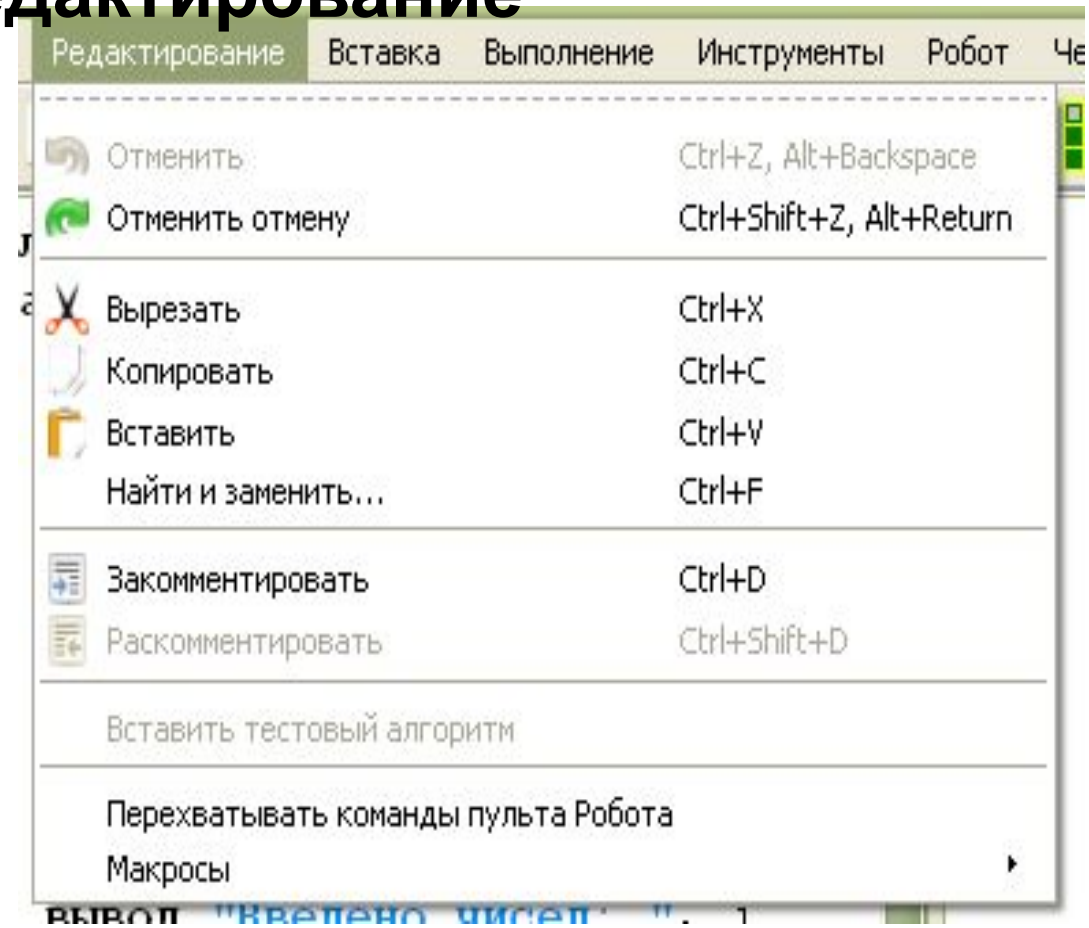
Паузы

Главное меню системы КуМир

Команды меню
Программа

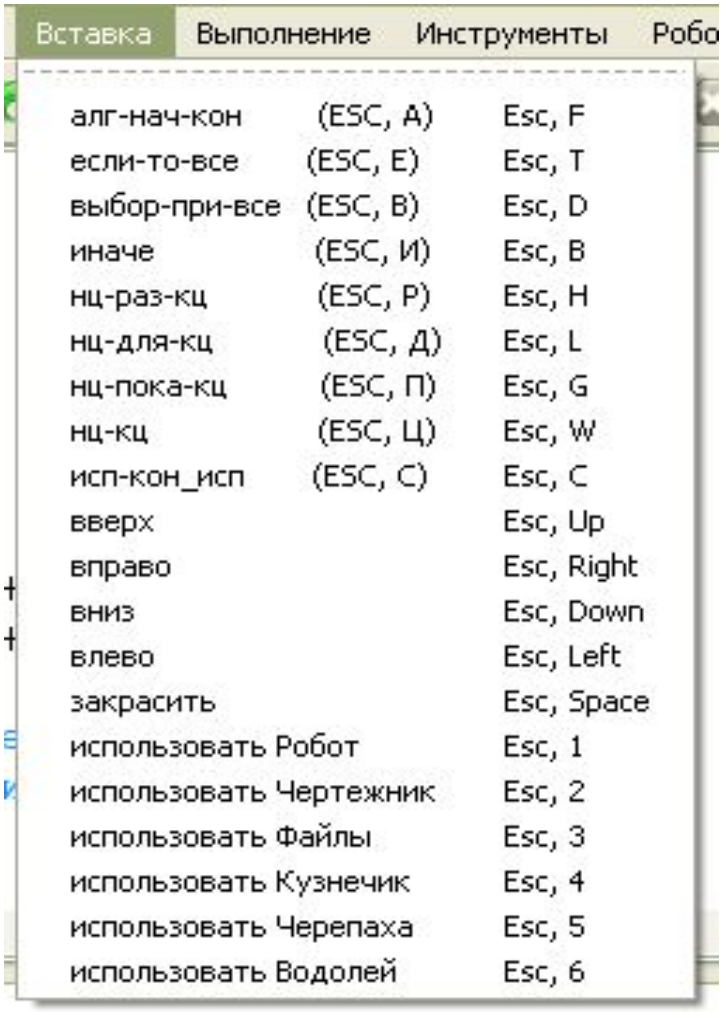
Главное меню системы КуМир

Команды меню
Редактирование



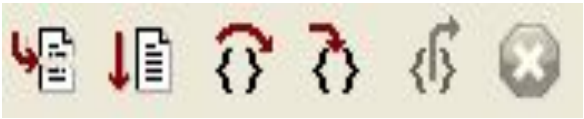
Главное меню системы КуМир

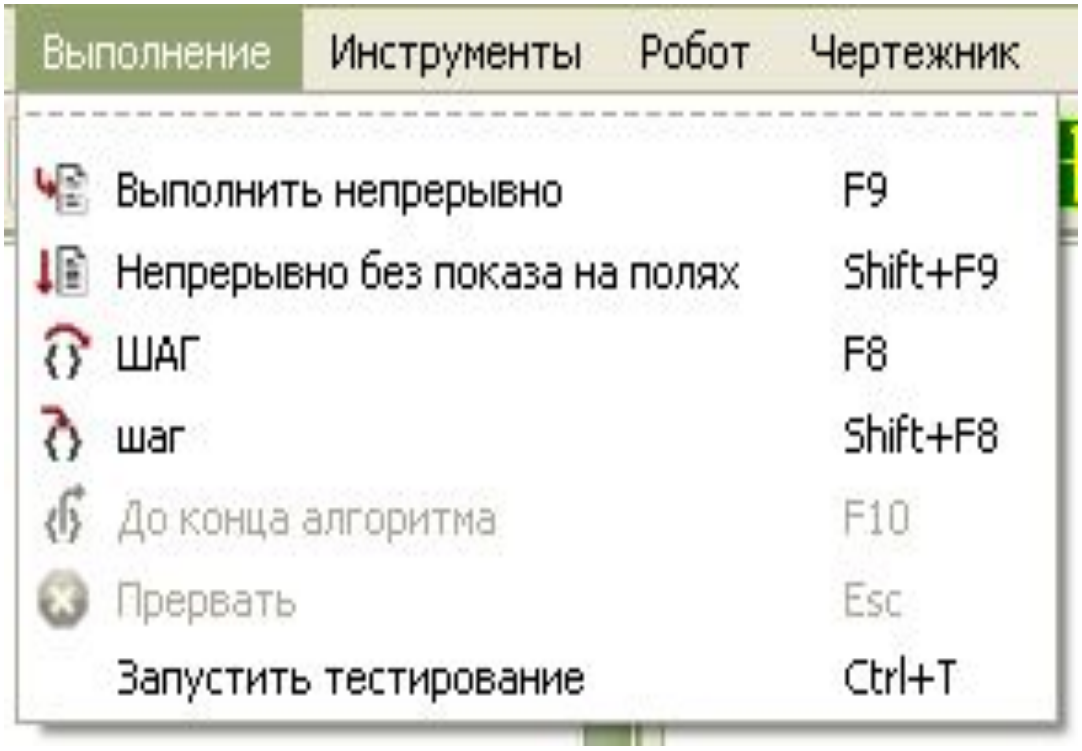
Команды меню **Вставка**



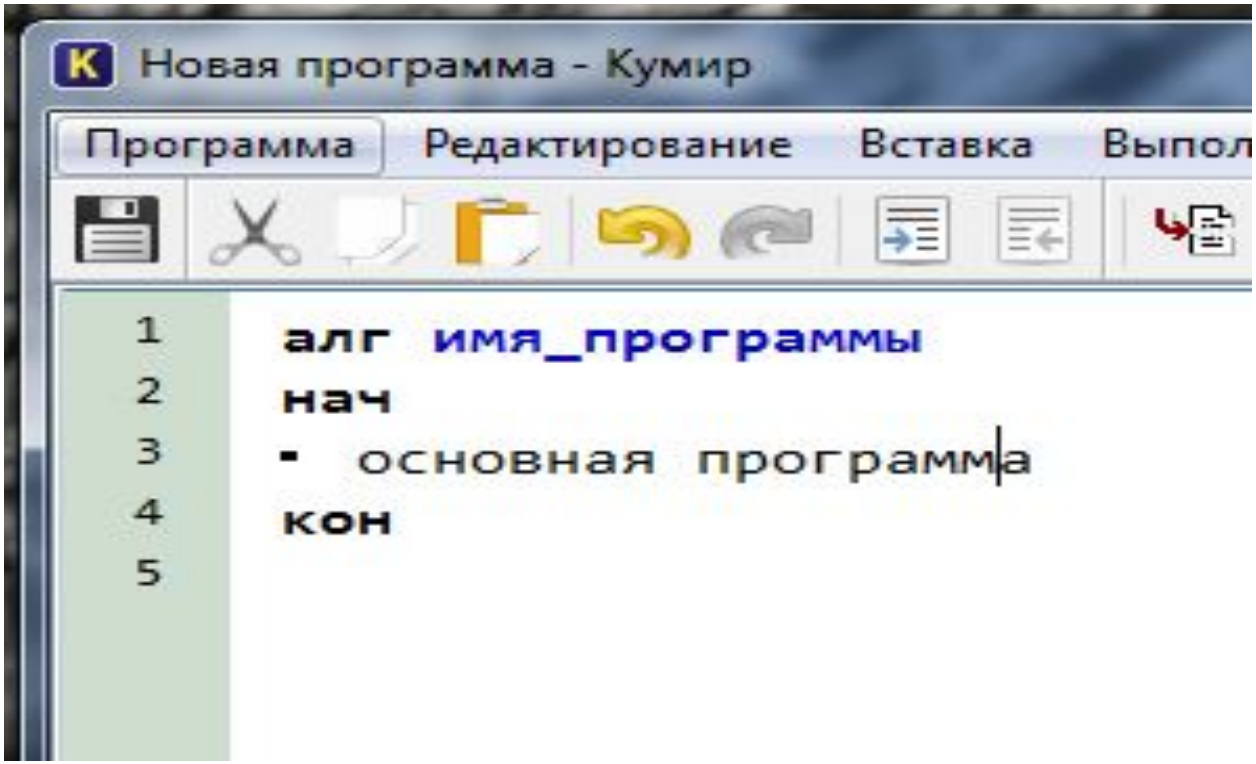
Главное меню системы КуМир

Команды меню **Выполнение**



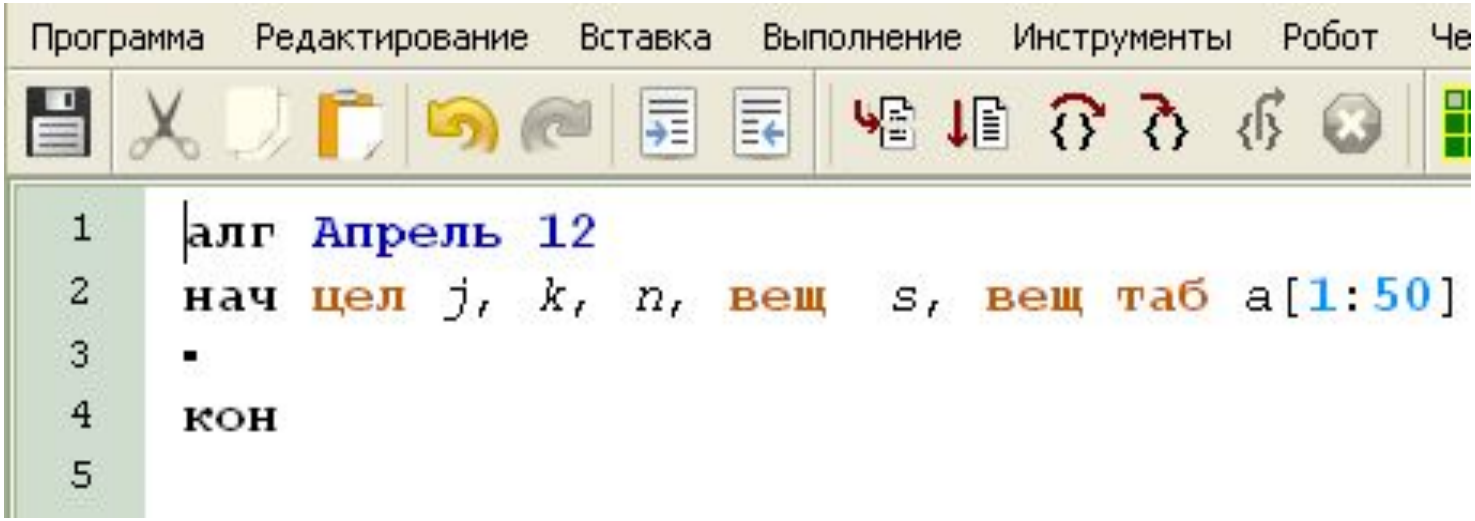


Общая структура программы (без параметров) в системе КуМир



Типы величин

Базовые типы	Типы для таблиц
целые числа **цел**	целые числа **цел таб**
вещественные числа **вещ**	вещественные числа **вещ таб**
логические **лог**	логические **лог таб**
символьные **сим**	символьные **сим таб**
литерные **лит**	литерные **лит таб**



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	sqrt(x)
абсолютная величина	abs(x) и iabs(x)
знак числа	(-1, 0 или 1) sign(x)

Арифметические действия и стандартные функции

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

Арифметические действия и стандартные функции

Название операции или функции	Форма записи
минимум из чисел x и y	min(x,y)
максимум из чисел x и y	max(x,y)
остаток от деления x на y	(x, y - целые) mod(x,y)
частное от деления x на y	(x, y - целые) div(x,y)
целая часть числа x	int(x)
случайное число в диапазоне от 0 до x	rnd(x)
оператор присваивания	a: = a +1

Операции сравнения

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

Логические операции

Операция	Форма записи
конъюнкция	a **и** b
дизъюнкция	a **или** б
отрицание	**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:=arcctg(k)
5  ▪ вывод s
```

Этапы решения задач

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

Простейшая программа (линейный алгоритм)

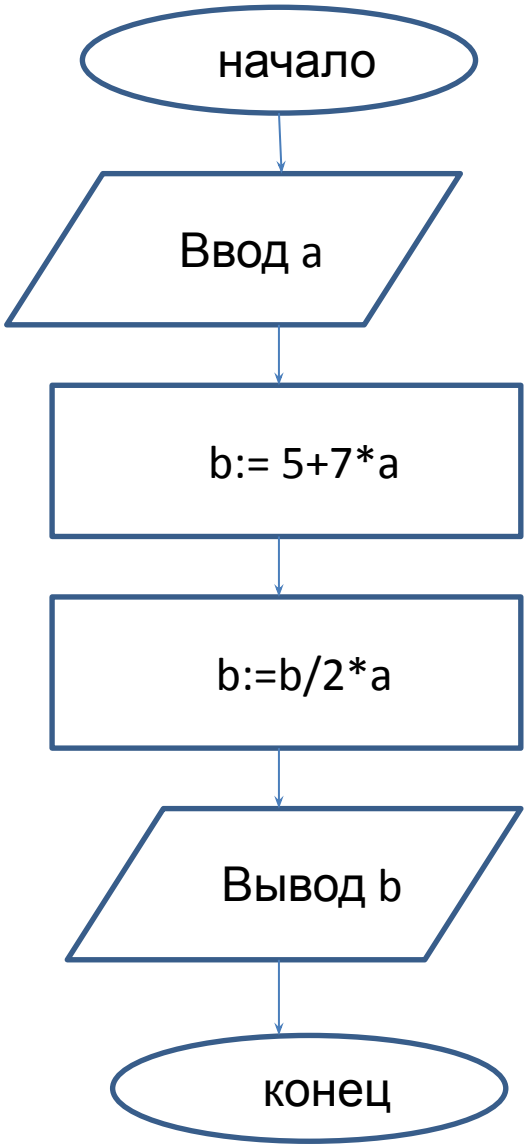
(ГИА задание №10 и №12)

В алгоритме, записанном ниже, используются действительные переменные a и b.
Определите значение переменной b после выполнения следующего фрагмента алгоритма:

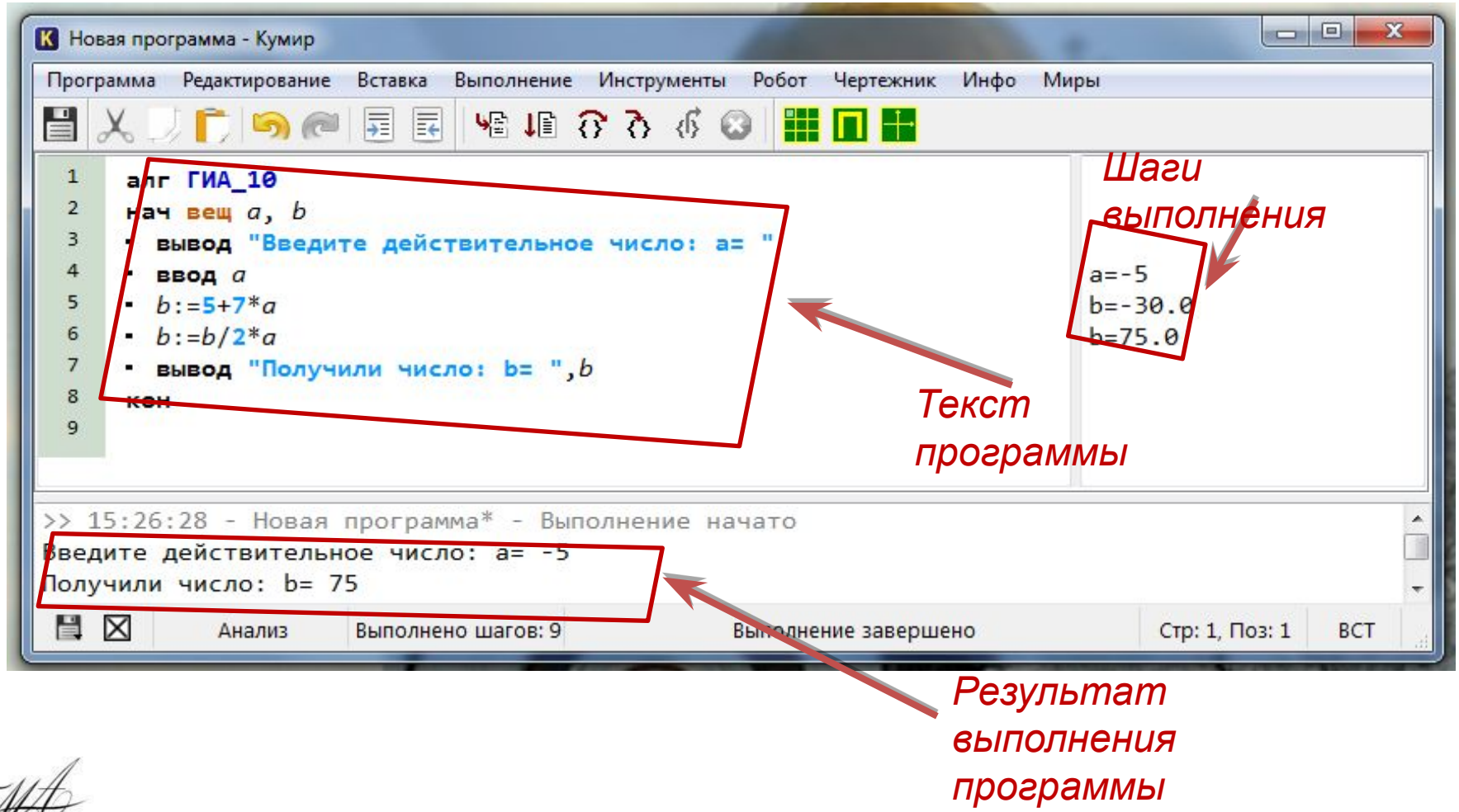
```
a:=-5;
b:=5+7*a;
b:=b/2*a;
```

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

Линейный алгоритм (блок-схема)



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



Структура «ветвление»

Команда *если – то - все* или неполное ветвление

Общий вид команды:

если условие
то *действия*
все

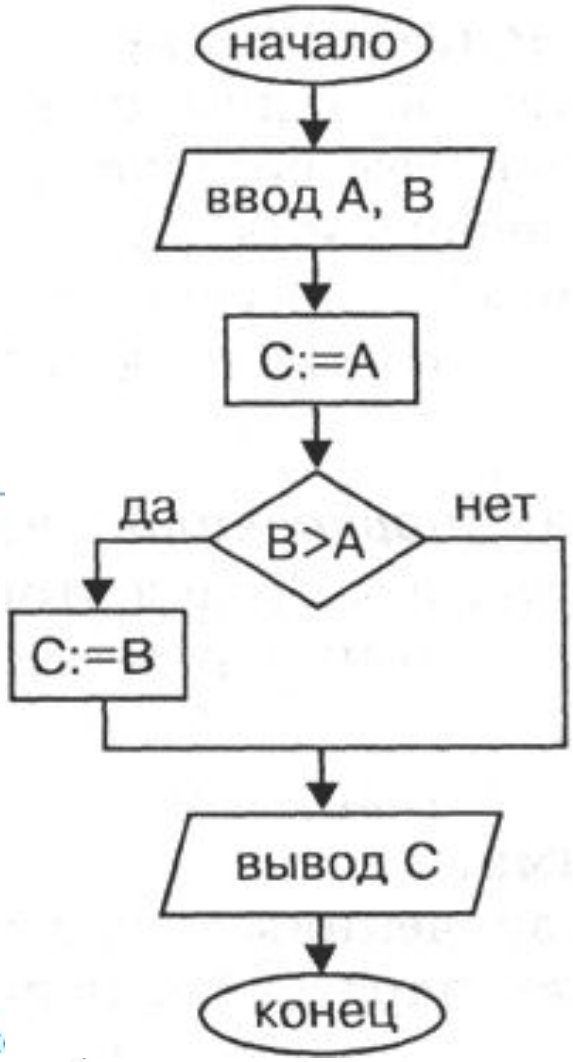
```
1   алг ветвление
2   нач вещ x, y
3   ▪ вывод "Введите значение x= "
4   ▪ ввод x
5   ▪ вывод "Введите значение y= "
6   ▪ ввод y
7   ▪ если x>0
8   ▪ ▪ то y:=y+5
9   ▪ все
10  ▪ вывод "Новое значение y= ",y
11  кон
12
```

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 чилса 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 равн
12  кон
13
```



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

Общий вид команды:

если условие

 то *действия 1*

 иначе *действия 2*

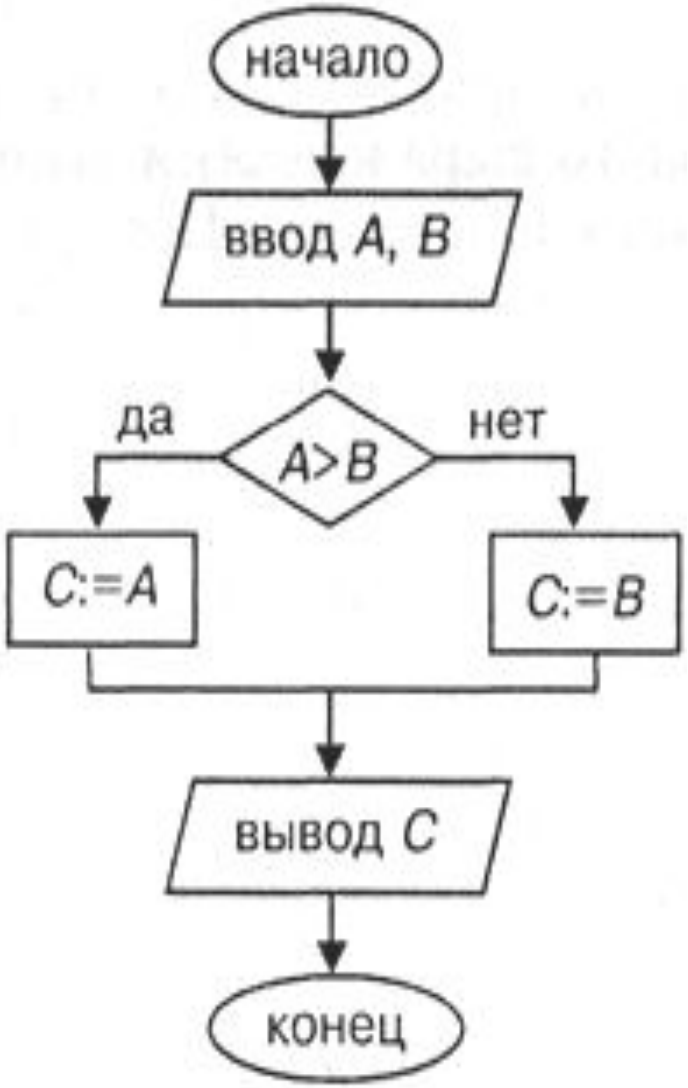
все

```
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
```

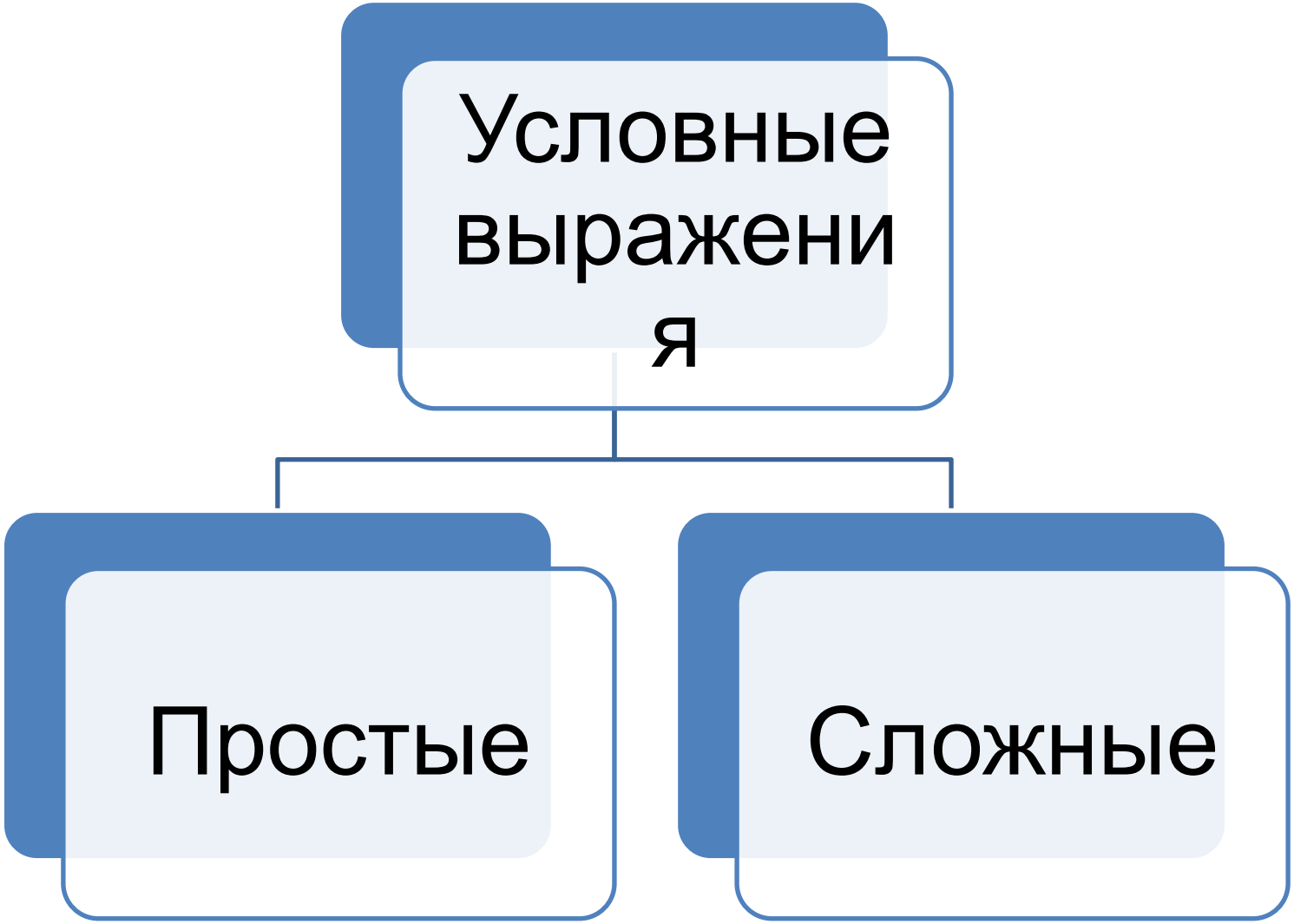
Пример 2

(полное ветвление)

Даны два числа; выбрать большее из них.




```
1   алг неполное ветвление
2   нач вещ a, b, c
3   • вывод "Введите значение чилса 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 выражения
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 действие1
        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 a[1:50] – массив целых чисел из 50 элементов;
вещ таб a[1:50] – массив вещественных чисел из 50 элементов.

```
нач
. цел N=20
. цел таб a[1:N], b[1:N]
...
```

Циклические конструкции

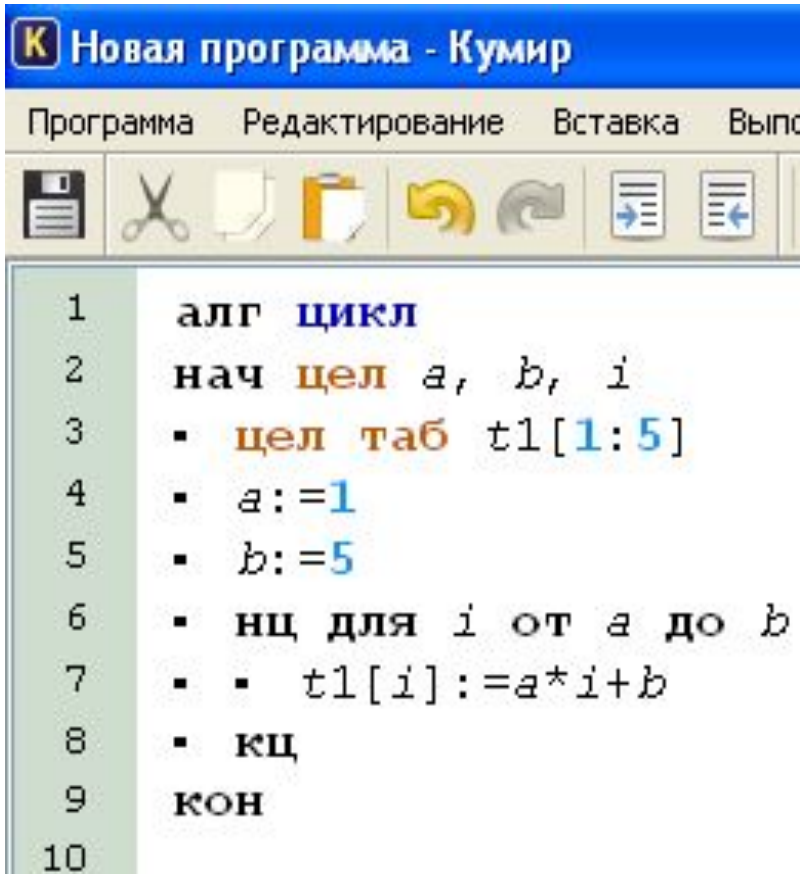
Цикл «для»

Общий вид:

нц для i **от** i1 **до** i2

тело цикла (последовательность команд)

кц



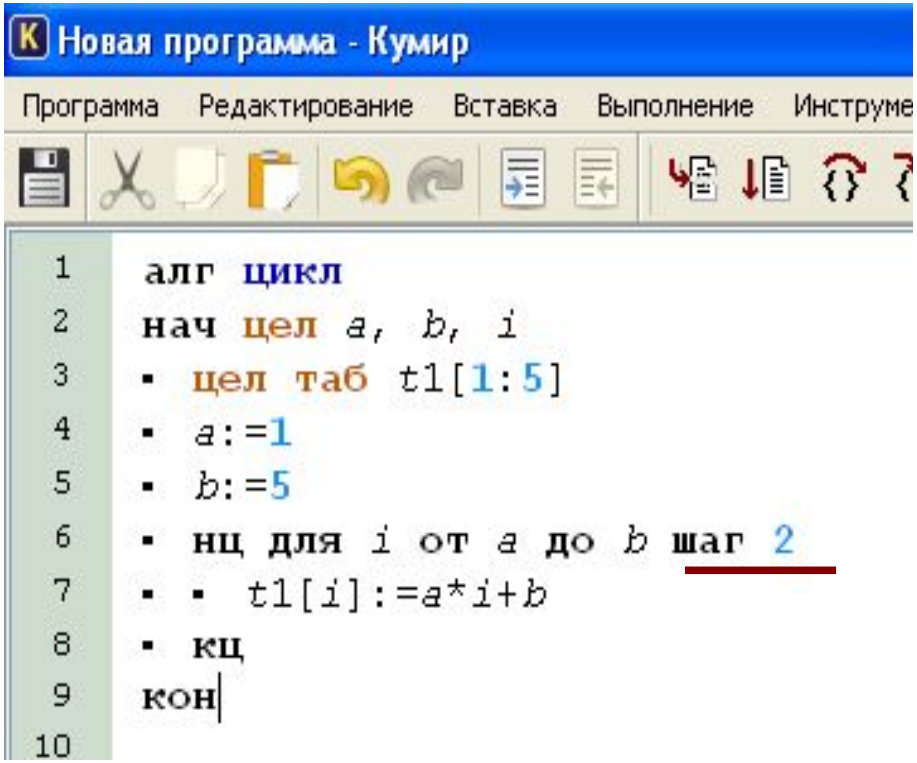
Где i, i1, i2 — переменные целого типа.
Параметр цикла i изменяется от i1 до i2 с шагом 1. Должно выполняться неравенство **i1<i2**, если же **i1=i2**, то тело цикла выполнится один раз, если **i1>i2**, то тело цикла не выполнится ни разу.

Цикл «для»
(общий вид с шагом)

нц для i **от** i1 **до** i2 **шаг** i3

тело цикла (последовательность команд)

кц



- Если шаг i3>0 , то тело цикла выполняется до тех пор, пока i≤i2 с условием, что i1≤i2.
- Если шаг i3<0 , то тело цикла выполняется до тех пор, пока i≥i1, с условием, что i1≥i2.

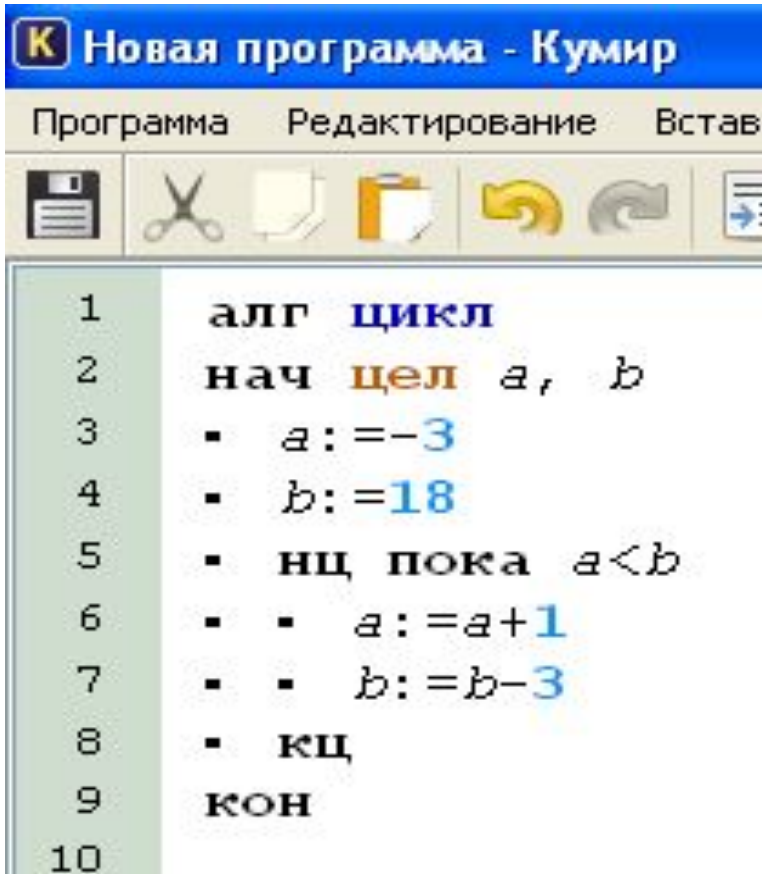
Цикл «пока»

Общий вид:

нц пока условие

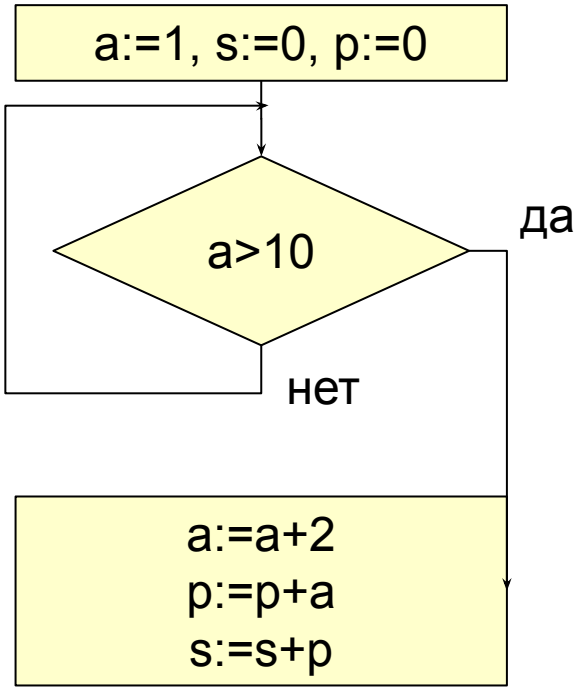
тело цикла (последовательность команд)

кц



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

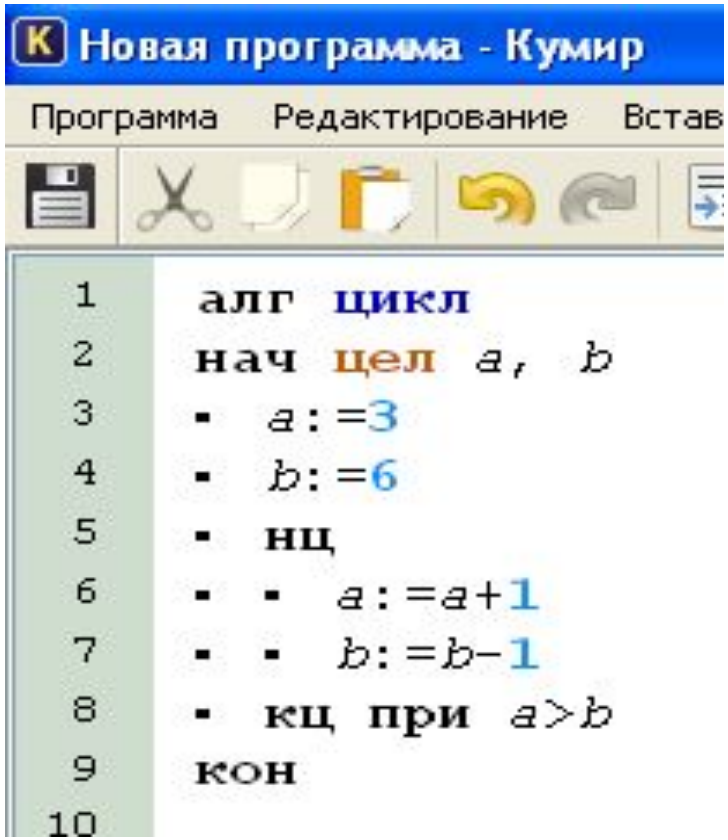
Цикл «кц_при»

Общий вид:

нц

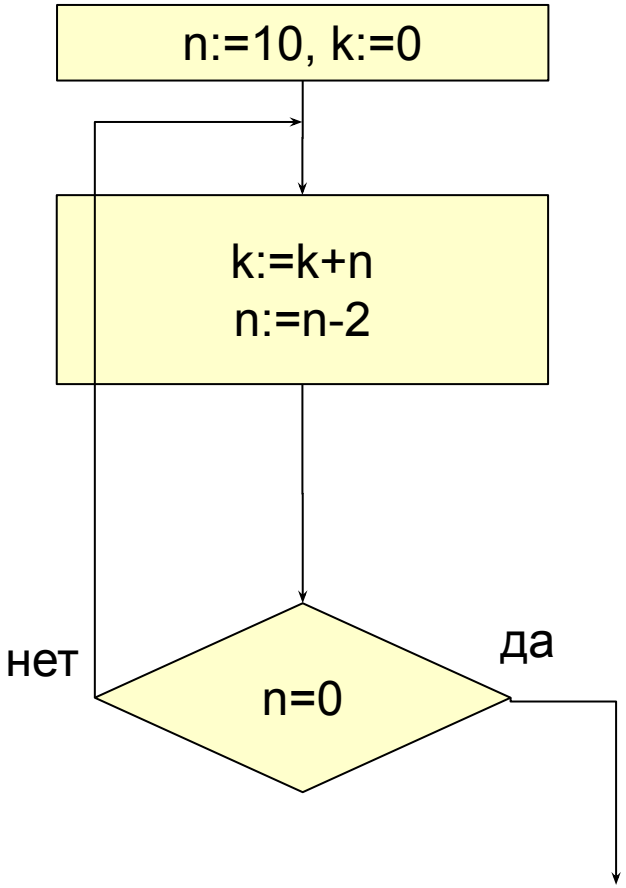
тело цикла
(последовательность
команд)

кц при условие



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	**длин**(стр)
Код символа в таблице КОИ-8	**код**(с)
Символ таблицы КОИ-8	**символ**(х)
Код символа в таблице Юникод	**юникод**(с)
Символ таблицы Юникод	**символ2**(х)
Строковое представление целого числа	**цел_в_лит**(х)

Команды обработки литерных величин

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

Исполнитель Строки

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