Программирование на языке Си (Dev-C++)

Столбова Юлия Викторовна
СПб ГБПОУ «ПКГХ»

Программа – это

- алгоритм, записанный на каком-либо языке программирования
- набор команд для компьютера

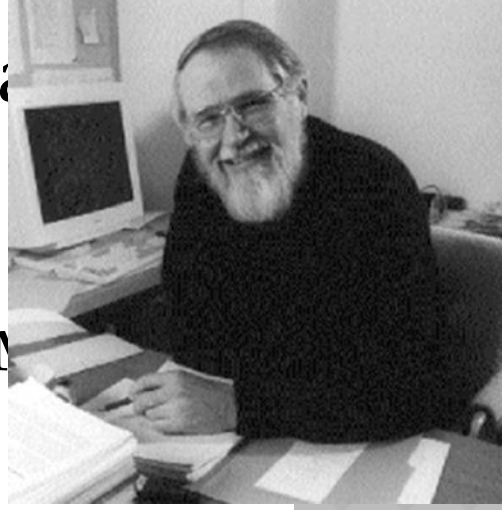
Команда – это описание действий, которые должен выполнить компьютер.

- откуда взять исходные данные?
- что нужно с ними сделать?
- куда поместить результат?

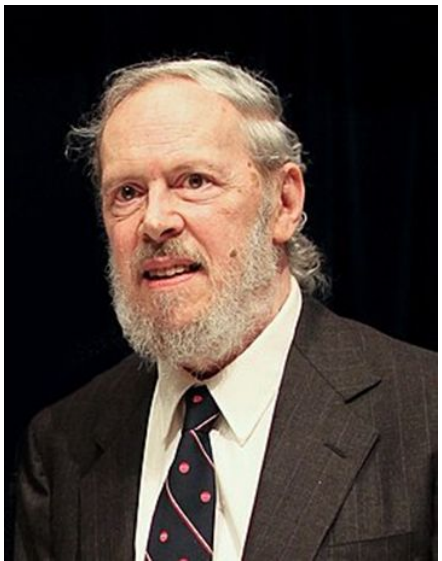
ТЕОРИЯ

ЯЗЫК Си

1972-1974 – Брайан Уилсон Кернига
и Де́ннис Макалистэйр Ритчи



- высокая скорость работы програм
- много возможностей
- стал основой многих современных языков *(C++, C#, Javascript, Java, ActionScript, PHP)*



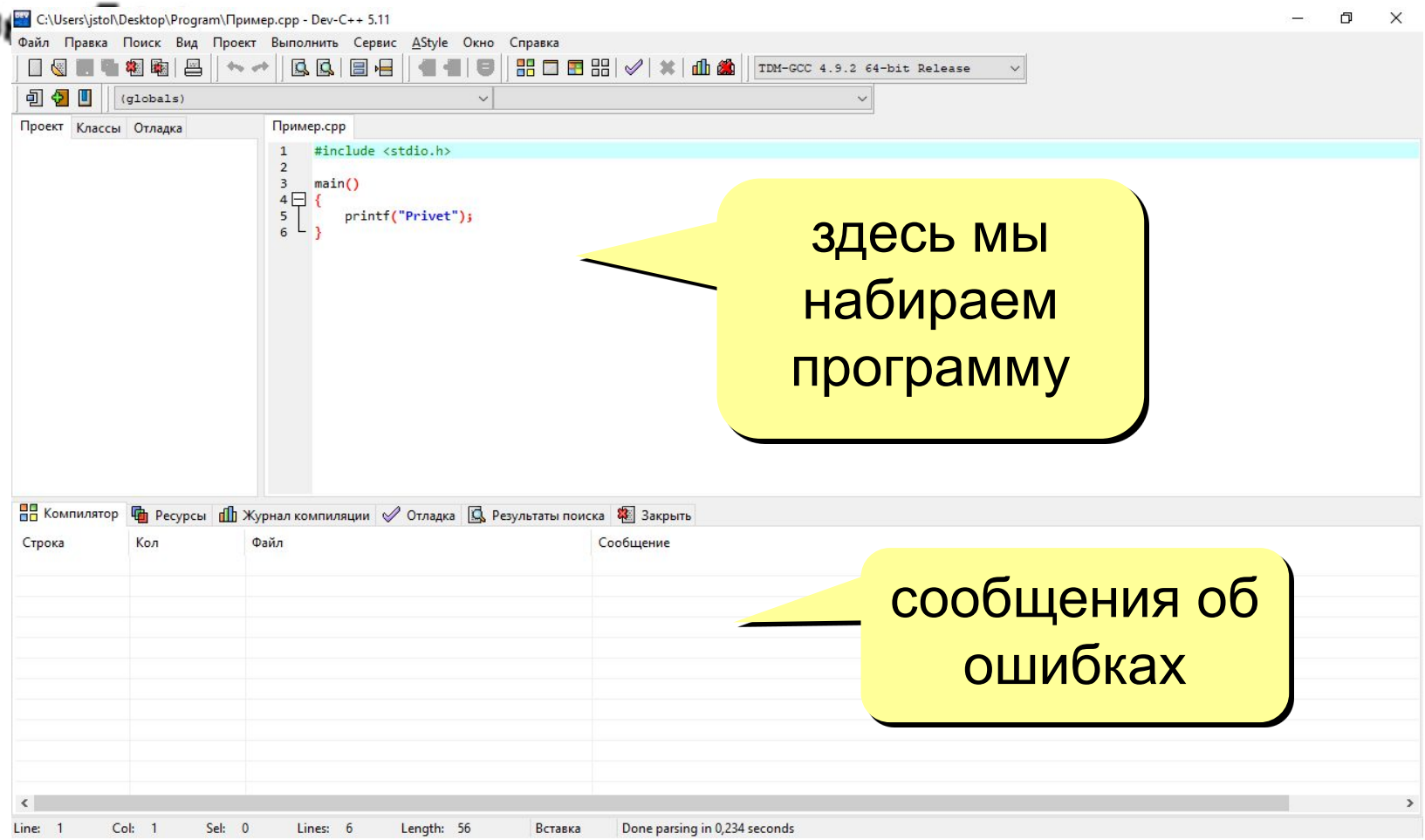
- много шансов сделать ошибку, которая не обнаруживается автоматически

IDE = *Integrated Development Environment*

интегрированная среда разработки:

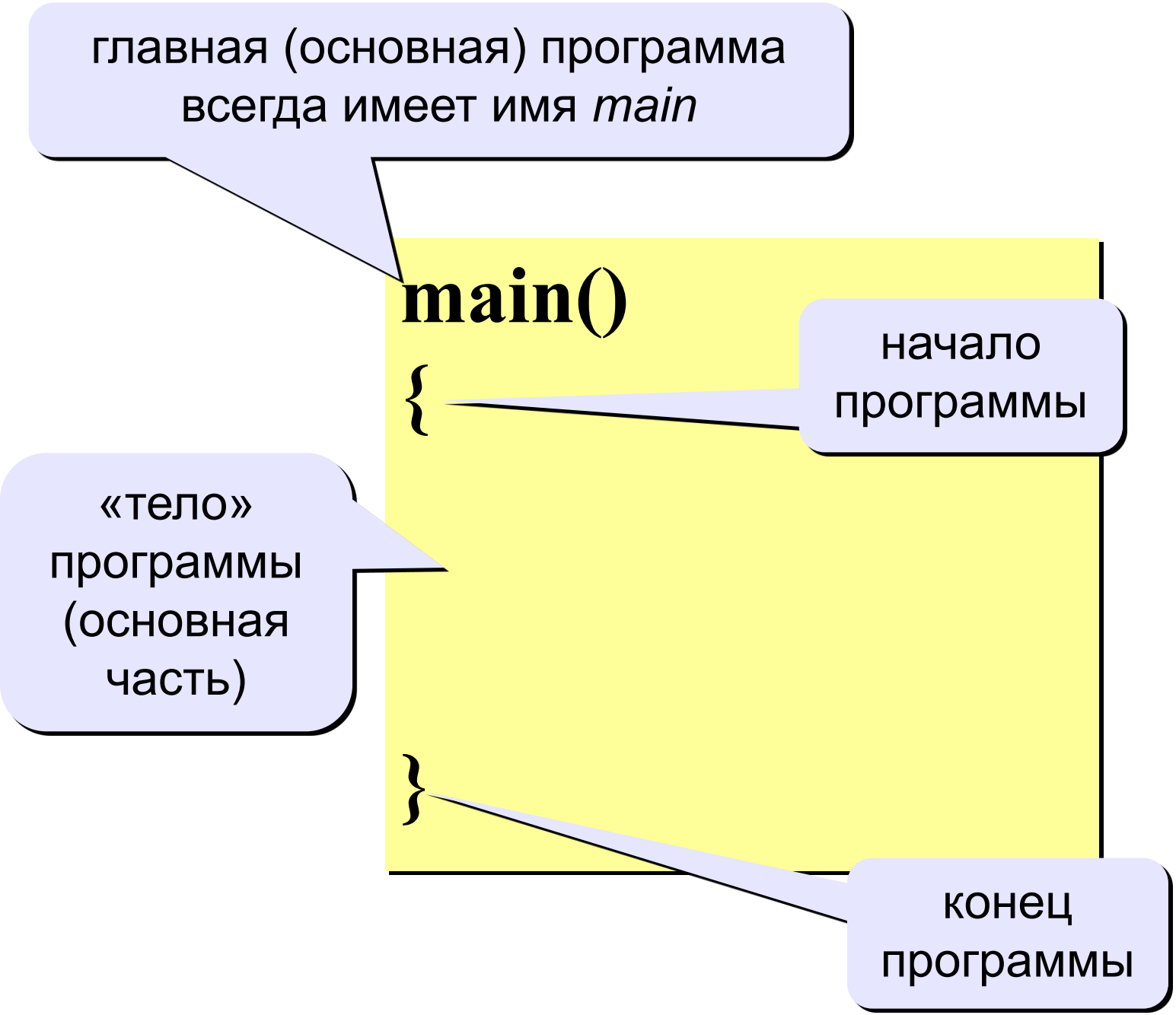
- текстовый редактор для создания и редактирования текстов программ
- транслятор для перевода текстов программ на Си и Си++ в команды процессора
- компоновщик для создания исполняемого файла (EXE-файла), подключаются стандартные функции
- отладчик для поиска ошибок в программах

Оболочка Dev C ++


C:\Users\jstol\Desktop\Program\Пример.cpp - Dev-C++ 5.11
Файл Правка Поиск Вид Проект Выполнить Сервис AStyle Окно Справка
TDM-GCC 4.9.2 64-bit Release
(globals)
Проект Классы Отладка
Пример.cpp
#include <stdio.h>
main()
{
printf("Privet");
}
здесь мы набираем программу
Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска Закрыть
Строка Кол Файл Сообщение
сообщения об ошибках
Line: 1 Col: 1 Sel: 0 Lines: 6 Length: 56 Вставка Done parsing in 0,234 seconds

Как начать

Простейшая


главная (основная) программа всегда имеет имя *main*
main()
{
начало программы
«тело» программы (основная часть)
}
конец программы

#include <stdio.h> описание стандартных функций ввода и вывода.

#include <conio.h> описание функций для работы с клавиатурой и монитором.

#include <locale.h> используется для задач, связанных с локализацией.

Локаль — это сочетание языковых и культурных аспектов. Они включают в себя: язык сообщений, различные наборы символов, лексикографические соглашения и т.п. setlocale(LC_ALL,"");

#include <stdlib.h> файл содержит в себе функции, занимающиеся выделением памяти, контролем процесса выполнения программы, преобразованием типов и другие.

#include <time.h> файл, содержащий типы и функции для работы с датой и временем. srand(time(NULL));

#include <math.h> математические функции

Библиоте

Printf(" ");

```
#include <stdio.h>
#include <conio.h>
#include <locale.h>
main()
 {
setlocale(LC_ALL,"");
printf("Привет!");
 }
```

Вывод текста на

Printf(" ");

Переход на новую строку

```
#include <stdio.h>
#include <conio.h>
#include <locale.h>
main()
{
setlocale(LC_ALL,"");
printf("Привет,\n Вася!");
getch();
}
```

ожидание нажатия на любую клавишу

последовательность \n (код 10) переход на новую строку

на экране:

```
Привет,
Вася!
```

Вывод текста на

Комментарии к

```
#include <stdio.h>
#include <conio.h>
#include <locale.h>
main()
{
setlocale(LC_ALL,"");
printf("Привет,\n Вася!"); // вывод на экран
getch(); /* ждать нажатия клавиши */
}
```

комментарий до конца строки

комментарий между /* несколько строк кода или текста*/

Задания

1: Вывести на экран текст "лесенкой"

Вася

пошел

гулять

2: Вывести на экран рисунок из букв

Ж
ЖЖЖ
ЖЖЖЖЖ
ЖЖЖЖЖЖЖ
НН НН
ZZZZ

Самостоятельное

Что такое

Переменная – это ячейка в памяти компьютера, которая имеет имя и хранит некоторое значение.

- Значение переменной может меняться во время выполнения программы.
- При записи в ячейку нового значения старое стирается.

Типы переменных

- **int** – целое число (4 байта)
- **float** – вещественное число, *floating point* (4 байта)
- **char** – символ, *character* (1 байт)

Могут включать

- латинские буквы (A-Z, a-z)
- знак подчеркивания _
- цифры 0-9

! Имя не может начинаться с цифры или знака подчеркивания!

НЕ могут включать

- русские буквы
- пробелы
- скобки, знаки +, =, !, ? и др.

Объявить переменную – определить ее тип, имя, начальное значение, и выделить ей место в памяти.

```
main()
{
int a; // переменная a целого типа
float b, c; // переменные b и c вещественного типа
int Tu104, Il86=23, Yak42; // переменные целого типа
float x=4.56; // целая и дробная части отделяются точкой
char c, c2='A', m; // символьные переменные c, m, c2 = 'A'
```

Если начальное значение не задано, в этой ячейке находится «мусор»!

Объявление

Оператор – это команда языка программирования высокого уровня.

Оператор присваивания служит для изменения значения переменной.

Общая

имя переменной = *выражение;*

куда записать

что записать

Арифметическое выражение может включать

- константы (постоянные)
- имена переменных
- знаки арифметических операций:

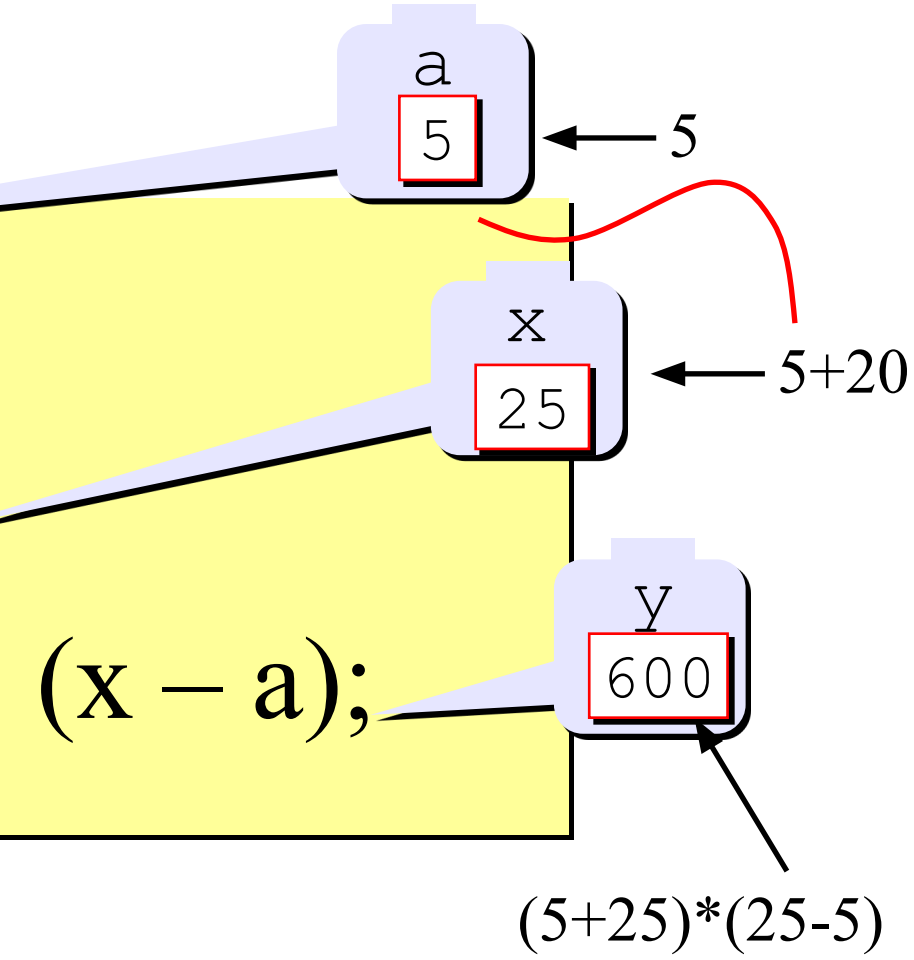
  \+ - * / %

- вызовы функций
- круглые скобки ( )

остаток от деления

Пример

```
a = 5;

x = a + 20;

y = (a + x) * (x – a);
```



scanf ("% ", &); – форматный ввод

```
scanf ("%d%d", &a, &b);
```

Пример: Ввести два целых числа и вывести на экран их

```
#include <stdio.h>
#include <conio.h>
#include <locale.h>
main()
{
  setlocale(LC_ALL,"");
  int a, b, c;
  printf("Введите два целых числа\n"); // оформление интерфейса
  scanf ("%d%d", &a, &b);
  c = a + b;
  printf("%d", c);
  g
}
```

В кавычках оператора scanf не должно быть лишних пробелов!

Ввод чисел с

Ввод чисел с

scanf – форматный ввод

формат ввода

адреса ячеек, куда записать введенные числа

```
scanf ("%d%d", &a, &b);
```

ждать ввода с клавиатуры двух целых чисел (через пробел или *Enter*), первое из них записать в переменную a, второе – в b

Формат – символьная строка, которая показывает, какие числа вводятся (выводятся).

- %d – целое число
- %f – вещественное число
- %c – 1 символ
- %s – символьная строка

&a – адрес переменной a

7652

a 12

значение переменной a

```
int a, b;
scanf ("%d", a);
scanf ("%d", &a, &b);
scanf ("%d%d", &a);
scanf ("%d  %d", &a, &b);
scanf ("%f%f", &a, &b);
```

&a

%d%d

&a, &b

убрать пробел

%d%d

Что

Вывод чисел на

здесь вывести целое число

это число взять из ячейки

```
printf ("%d", c);
```

можно добавить любой текст

```
printf ("Результат: %d", c);
```

```
printf ("%d+%d=%d", a, b, c );
```

формат вывода

список значений
элементы списка разделяются запятыми

```
printf ("%d+%d=%d", a, b, a+b );
```

арифметическое выражение

Вывод целых

```
int x = 1234;
printf ("%d", x);
```

минимальное число позиций

Результат

```
1234
```

```
printf ("%9d", x);
```

всего 9 позиций

Результат

```
     1234
```

5 4

Вывод вещественных чисел

```
float x = 123.4567;
printf ("%f", x);
```

минимальное число позиций, 6 цифр в дробной части

Результат `123.456700`

```
printf ("%9.3f", x);
```

всего 9 позиций, 3 цифры в дробной части

Результат `  123.456`

```
printf ("%e", x);
```

стандартный вид: $1{,}23456 \cdot 10^2$

Результат `1.234560e+02`

```
printf ("%10.2e", x);
```

всего 10 позиций, 2 цифры в дробной части мантиссы

Результат `  1.23e+02`

Особенность деления в

При делении целых чисел остаток отбрасывается!

```
main()
{
int a = 7;
float x;
x = a / 4;
x = 4 / a;
x = float(a) / 4;
x = 1.*a / 4;
}
```

1

0

1.75

1.75

Сокращенная запись операций в Си

инкремент

полная запись	сокращенная запись
a = a + 1;	a++;
a = a + b;	a += b;
a = a - 1;	a--;
a = a – b;	a -= b;
a = a * b;	a *= b;
a = a / b;	a /= b;
a = a % b;	a %= b;

декремент

Что будет

```
int a = 1, b = 3;
printf("a+%d=a+b", b);
```

a+3=a+b

```
int a = 1, b = 3;
printf("%d=F(%d)", a, b);
```

1=F(3)

```
int a = 1, b = 3;
printf("a=F(%d);", b);
```

a=F(3);

```
int a = 1, b = 3;
printf("%d>%d!", a+b, b);
```

4>3!

```
int a = 1, b = 3;
printf("F(%d)=X(%d)", b, a);
```

F(3)=X(1)

Как записать оператор вывода?

X(3)=1

```
int a = 1, b = 3
printf("X(%d)=%d", b, a);
```

4=1+3

```
int a = 1, b = 3
printf("%d=%d+%d",a+b,a,b);
```

f(1)>f(3)

```
int a = 1, b = 3
printf("f(%d)>f(%d)", a, b);
```

<1<>3>

```
int a = 1, b = 3
printf("<%d<>%d>", a, b);
```

1+3=?

```
int a = 1, b = 3
printf("%d+%d=?", a, b);
```

Например: введенное четырехзначное число нужно разбить на отдельные цифры используя деление и взятие по модулю.

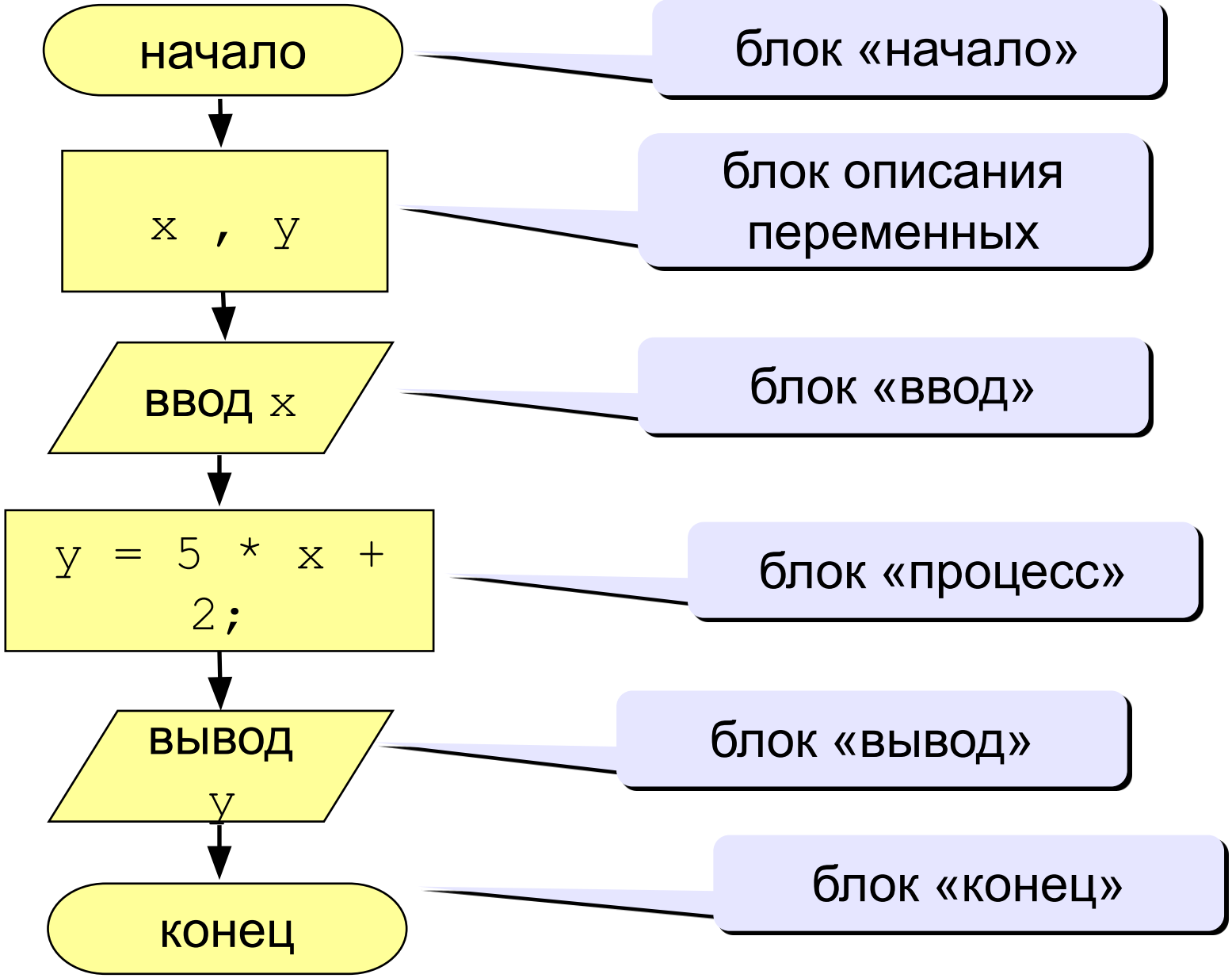
```
int x;      /*Введенное число*/
int A1;     /*Переменная первой цифры*/
int A2;     /*второй*/
int A3;     /*третьей*/
int A4;     /*четвертой*/
```

Разбивание числа на

```
printf(«Введите любое четырехзначное число: ");
    scanf("%d", & x);
/*-------------------------------------------------*/
/*Разбивание числа на отдельные цифры*/

/*Тысячи*/
    A1 = (x - (x % 1000)) / 1000;

 /*Сотни*/
    A2 = ((x - (x % 100)) - (x - (x % 1000))) / 100;

 /*Десятки*/
 A3 = ((x - (x % 10)) - (x - (x % 100))) / 10;

 /*Единицы*/
 A4 = x % 10;
 /*-------------------------------------------------*/
```

Разбивание числа на

Блок-схема линейного алгоритма



Написать программу по алгоритму. Оформить интерфейс программы.

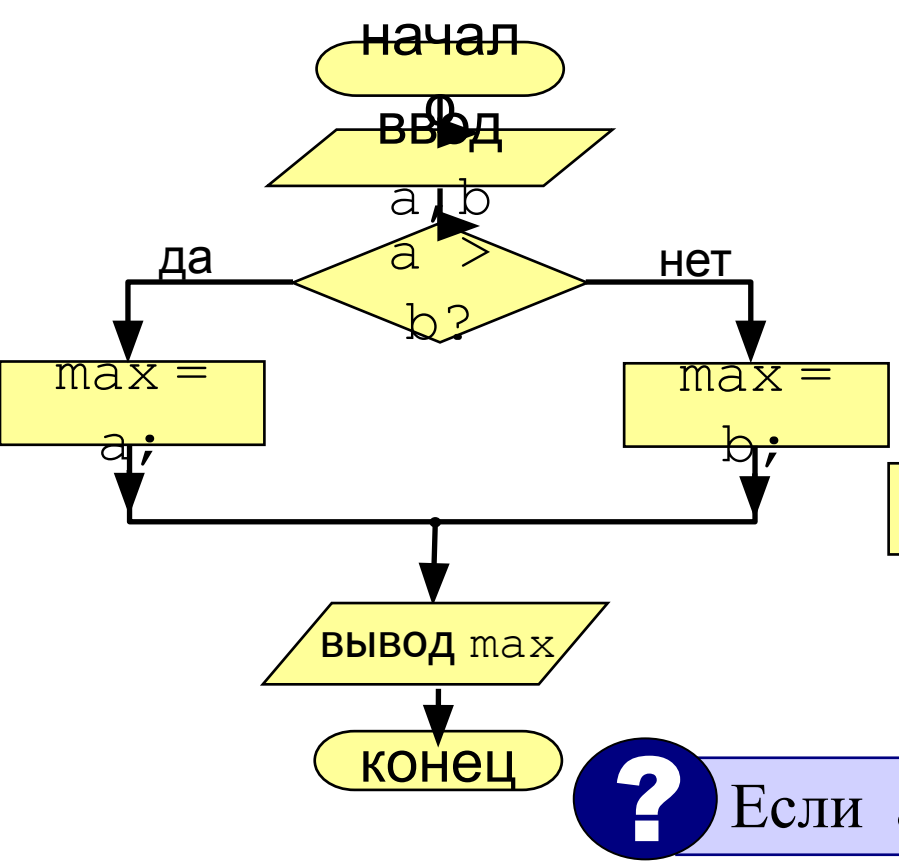
Программу сохранить как - Тест1

Алгоритмы, в которых последовательность шагов зависит от выполнения некоторых условий, называются разветвляющимися.

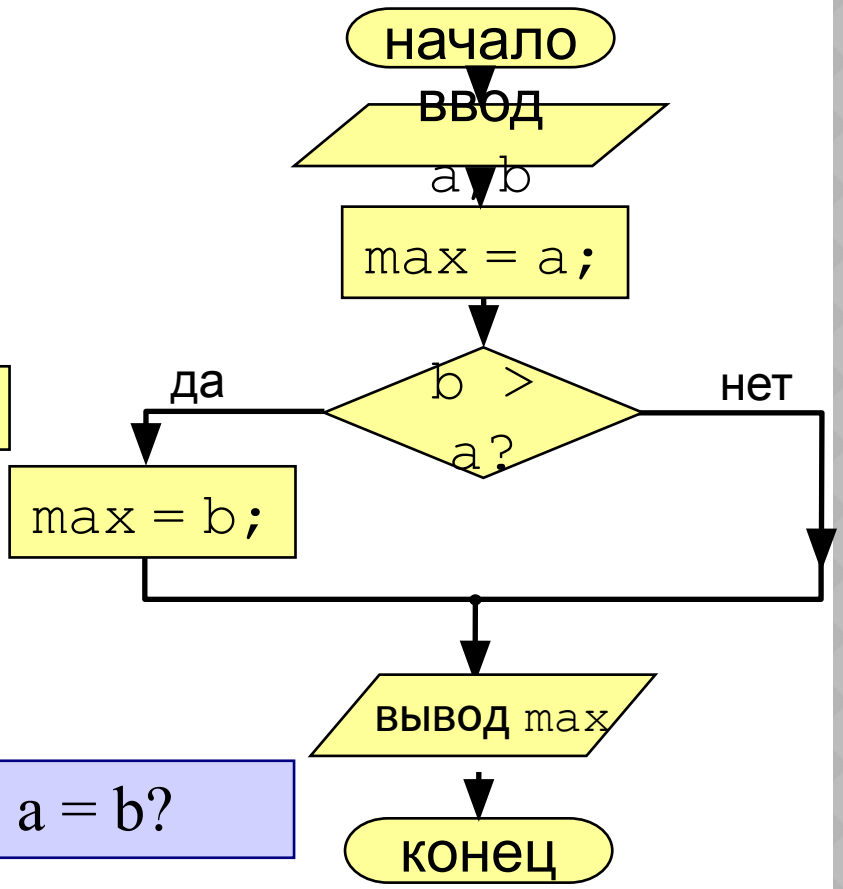
ВЕТВЛЕНИЯ

Блок-схема

Вариант 1.



Вариант 2.



? Если a = b?

```
if ( условие )
  {
     // что делать, если условие верно
  }
else
  {
     // что делать, если условие неверно
  }
```

Особенности:

- вторая часть (*else* …) может отсутствовать (неполная форма)
- если в блоке один оператор, можно убрать { }

Вариант 1. Программа

```
main()
{
   int a, b, max;
  printf("Введите два целых числа\n");
  scanf("%d%d", &a, &b );
  if (a > b) {
max = a;
   }
  else {
max = b;
   }
  printf("Наибольшее число %d", max);
  }
```

полная форма условного оператора

Что неправильно?

```
if (a > b ) {
 a = b;
 }
else
 b = a;
```

```
if ( a > b ) {
 a = b; }
else
 b = a;
```

```
if ( a > b )
a = b;
else
 b = a;
```

```
if ( a > b ) {
a = b;
c = 2*a; }
else
b = a;
```

Операторы ветвления

Вариант 2.

Программа

```
main()
{
 int a, b, max;
printf("Введите два целых числа\n");
scanf("%d%d", &a, &b );
max = a;
if (b > a)
max = b;
printf("Наибольшее число %d", max);
}
```

неполная форма условного оператора

Задания

1: Ввести два числа и вывести их в порядке возрастания.

Пример:

Введите два числа:

15 9

Ответ: 9 15

2: Ввести три числа и найти наибольшее из них.

Пример:

Введите три числа:

4 15 9

Наибольшее число 15

3: Ввести пять чисел и найти наибольшее из них.

Пример:

Введите пять чисел:

4 15 9 56 4

Наибольшее число 56

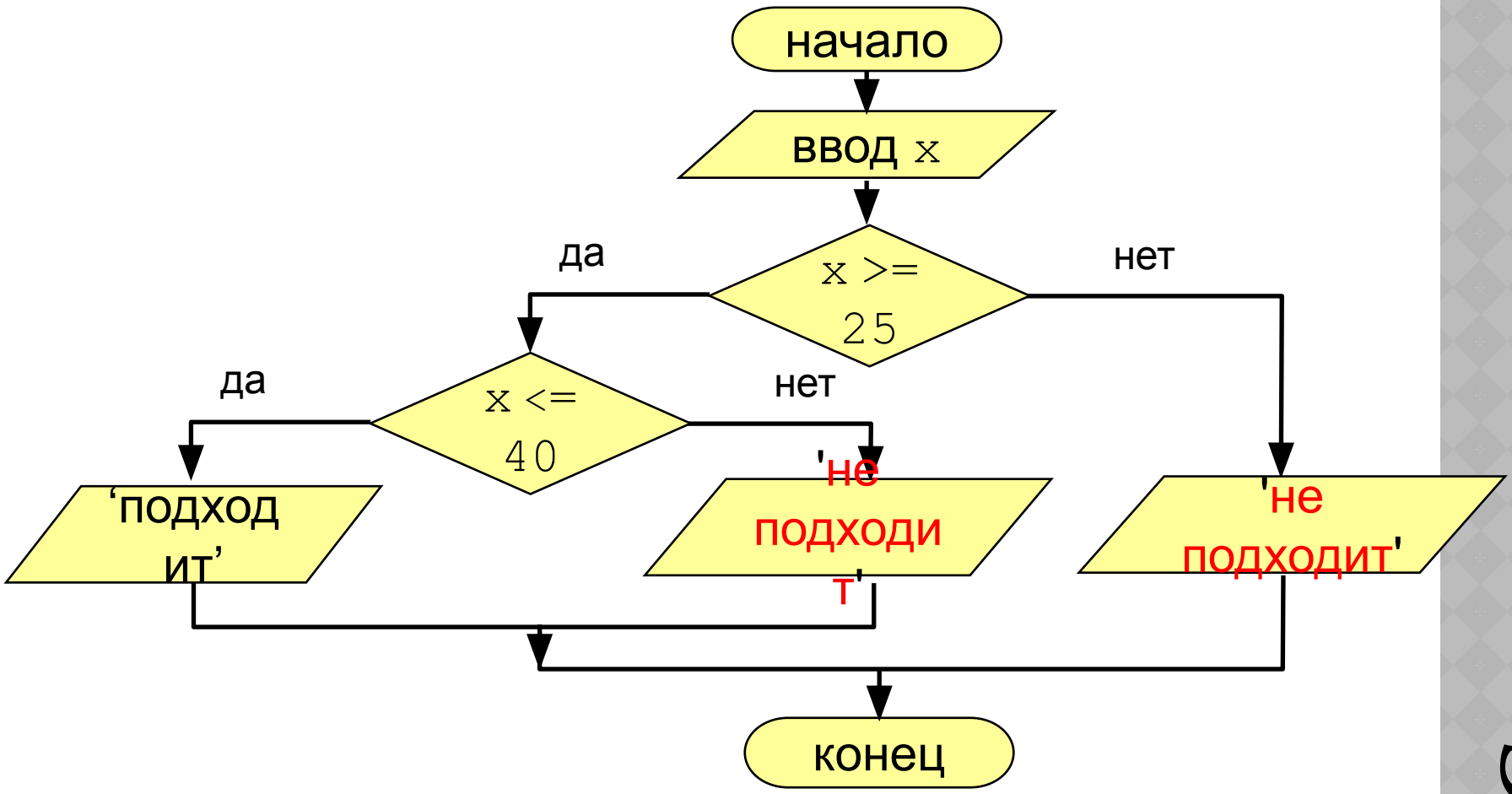
Задача. Фирма набирает сотрудников от 25 до 40 лет включительно. Ввести возраст человека и определить, подходит ли он фирме (вывести ответ «подходит» или «не подходит»).

Особенность: надо проверить, выполняются ли два условия одновременно.

? Можно ли решить известными методами?

Сложные

Вариант 1. Алгоритм



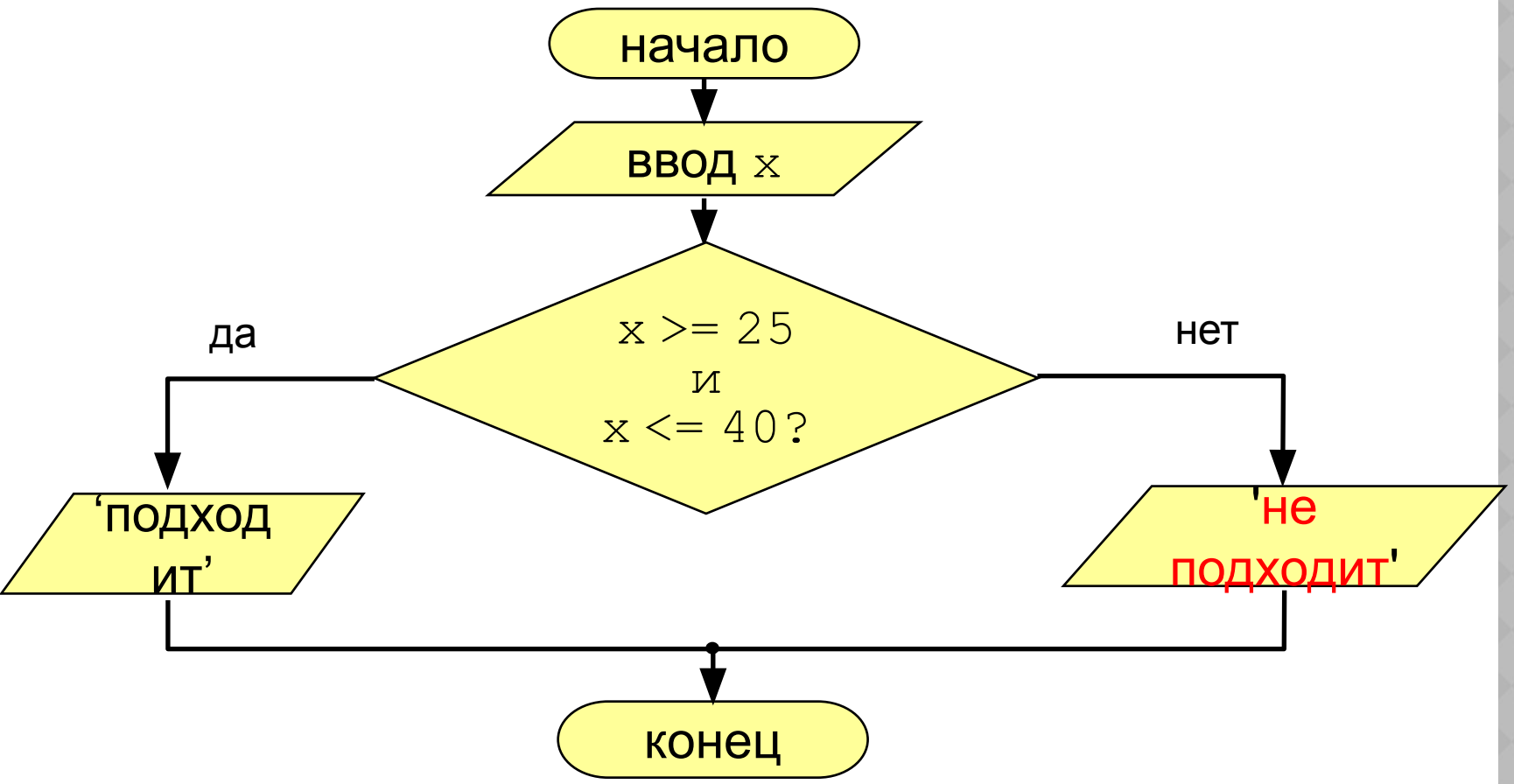
Сложные

Вариант 1. Программа

```
main()
{
  int x;
 printf("Введите возраст\n");
 scanf("%d", &x);
 if (x >= 25)
   if (x <= 40)
        printf("Подходит");
   else printf("Не подходит");
 else
   printf("Не подходит");
}
```

Сложные

Вариант 2. Алгоритм



Сложные

Вариант 2. Программа

```
main()
 {
 int x;
 printf("Введите возраст\n");
 scanf("%d", &x);
 if ( x >= 25 && x <= 40 )
       printf("Подходит");
 else printf("Не подходит");
 }
```

сложное условие

Сложные

Сложное условие – это условие, состоящее из нескольких простых условий (отношений), связанных с помощью логических операций:

! – НЕ (*not*, отрицание, инверсия)

&& – И (*and*, логическое умножение, конъюнкция, одновременное выполнение условий)

|| – ИЛИ (*or*, логическое сложение, дизъюнкция, выполнение хотя бы одного из условий)

Простые условия (отношения)

< <= > >= == !=

равно

не равно

Сложные

Порядок выполнения сложных условий:

- выражения в скобках
- ! (НЕ, отрицание)
- <, <=, >, >=
- ==, !=
- && (И)
- || (ИЛИ)

Пример:

```
      2  1      6    3      5        4
if ( !(a > b) || c != d && b == a)
  {
  ...
  }
```

Сложные

Примеры

Истинно или ложно при a = 2; b = 3; c = 4;

!(a > b) — 1

a < b && b < c — 1

!(a >= b) || c == d — 1

a < c || b < c && b < a — 1

a > b || !(b < c) — 0

Для каких значений x истинны условия:

x < 6 && x < 10	(-∞, 6)
x < 6 && x > 10	∅
x > 6 && x < 10	(6, 10)
x > 6 && x > 10	(10, ∞)
x < 6 || x < 10	(-∞, 10)
x < 6 || x > 10	(-∞, 6) ∪ (10,∞)
x > 6 || x < 10	(-∞, ∞)
x > 6 || x > 10	(6, ∞)

Задания

«1»: Ввести три числа и определить, верно ли, что они вводились в порядке возрастания.

Пример:

Введите три числа: 4 5 17
да

Введите три числа: 45 3 20
нет

«2»: Ввести номер месяца и вывести название времени года.

Пример:

Введите номер месяца: 4
весна

Введите номер месяца: 15
такого номера нет

«3»: Ввести возраст человека (от 1 до 150 лет) и вывести его вместе с последующим словом «год», «года» или «лет».

Пример:

Введите возраст: 24
Вам 24 года

Введите возраст: 57
Вам 57 лет

Сложные

Цикл

Цикл – это многократное выполнение одинаковых действий.

- цикл с известным числом шагов
- цикл с неизвестным числом шагов (цикл с условием)

Задача. Вывести на экран 5 раз слово «Привет».

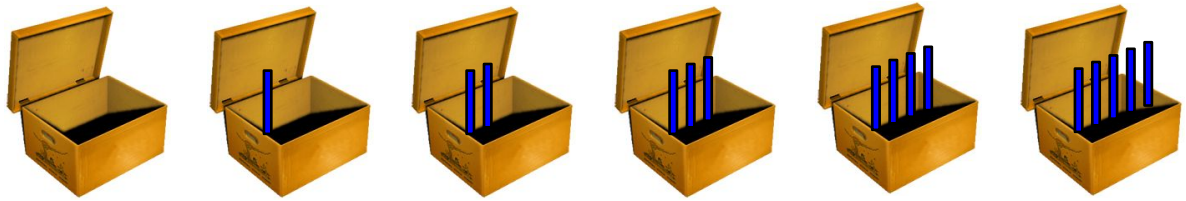
Особенность: одинаковые действия выполняются 5 раз.

5 раз вывести слово привет используя printf

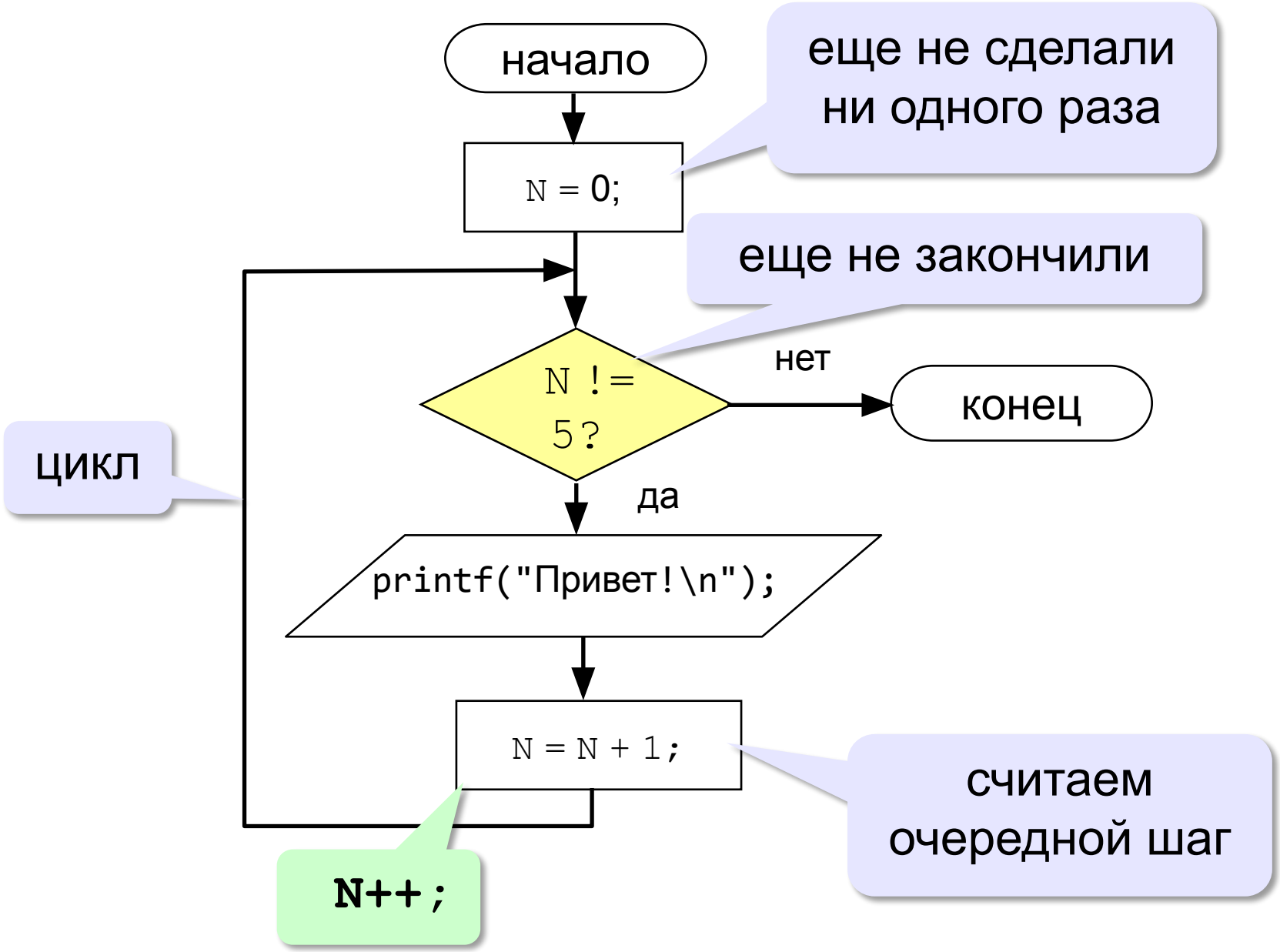
Цикл

? Как отсчитать ровно 5 раз?

? Как запоминать, сколько раз уже сделали?



N := N + 1


начало
N = 0;
еще не сделали ни одного раза
еще не закончили
N != 5?
нет
конец
да
цикл
printf("Привет!\n");
N = N + 1;
считаем очередной шаг
N++;

Цикл с предусловием While

```
main()
{
  int N;
  N = 0;
  while ( N != 5 )
  {
    printf("Привет!\n");
    N ++;
  }
}
```

экране?

```
main()
{
  int N;
  N = 2;
  while ( N != 5 )
  {
    printf("%d\n", N);
    N += 2;
  }
}
```

2
4
6
8
10
12
14
16
...

! Условие цикла никогда не станет ложным – это зацикливание!

Что получим в результате работы программы на экране?

```
main()
{
  int N;
  N = 1;
  while ( N != 5 )
  {
    printf("%d\n",
  N*N*N);
    N = N + 1;
  }
}
```

1
8
27
64
125

Задания

«1»: Ввести натуральное число вывести квадраты и кубы всех чисел от 1 до этого числа.

Пример: Введите натуральное число: 3

1: 1 1

2: 4 8

3: 9 27

«2»: Ввести два целых числа a и b ($a \leq b$) и вывести квадраты все чисел от a до b.

Пример: Введите два числа: 4 5

4*4=16

5*5=25

«3»: Ввести два целых числа a и b ($a \leq b$) и вывести сумму квадратов всех чисел от a до b.

Пример: Введите два числа: 4 10

Сумма квадратов 371

Циклы с

Цикл с неизвестным числом шагов

Пример: Отпилить полено от бревна. Сколько раз надо сделать движения пилой?

Задача: Ввести целое число (<2000000) и определить число цифр в нем.

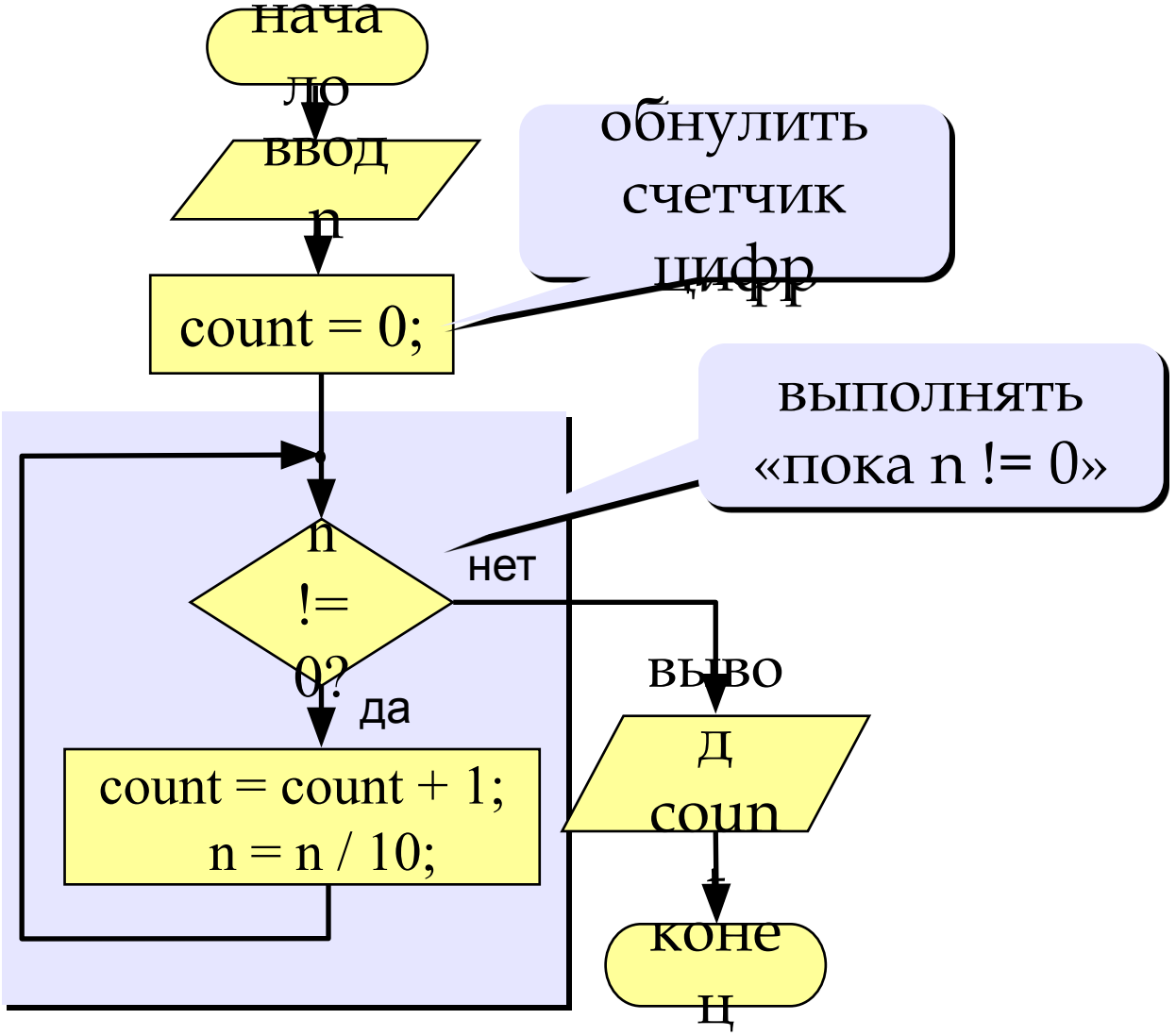
Идея решения: Отсекаем последовательно последнюю цифру, увеличиваем счетчик.

n	count
123	0
12	1
1	2
0	3

Проблема: Неизвестно, сколько шагов надо сделать.

Решение: Надо остановиться, когда n = 0, т.е. надо делать «пока n != 0».

Алгоритм



Циклы с