

Тема 1.4. Основные алгоритмические конструкции языка C/C++

Операторы языка с/с++

Оператор – неделимый элемент программы, выполняющий определенные алгоритмические действия

Условные операторы (if...else;
switch...)

- Условные конструкции

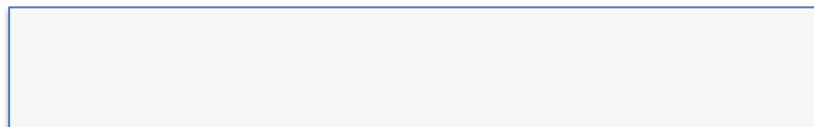
Циклические операторы (for();
while();do...while())

- Циклические конструкции

Оператор разрыва break; оператор
продолжения continue; оператор
возврата return

- Безусловные конструкции

Простой оператор; пустой
оператор; составной



Разветвляющиеся алгоритмы

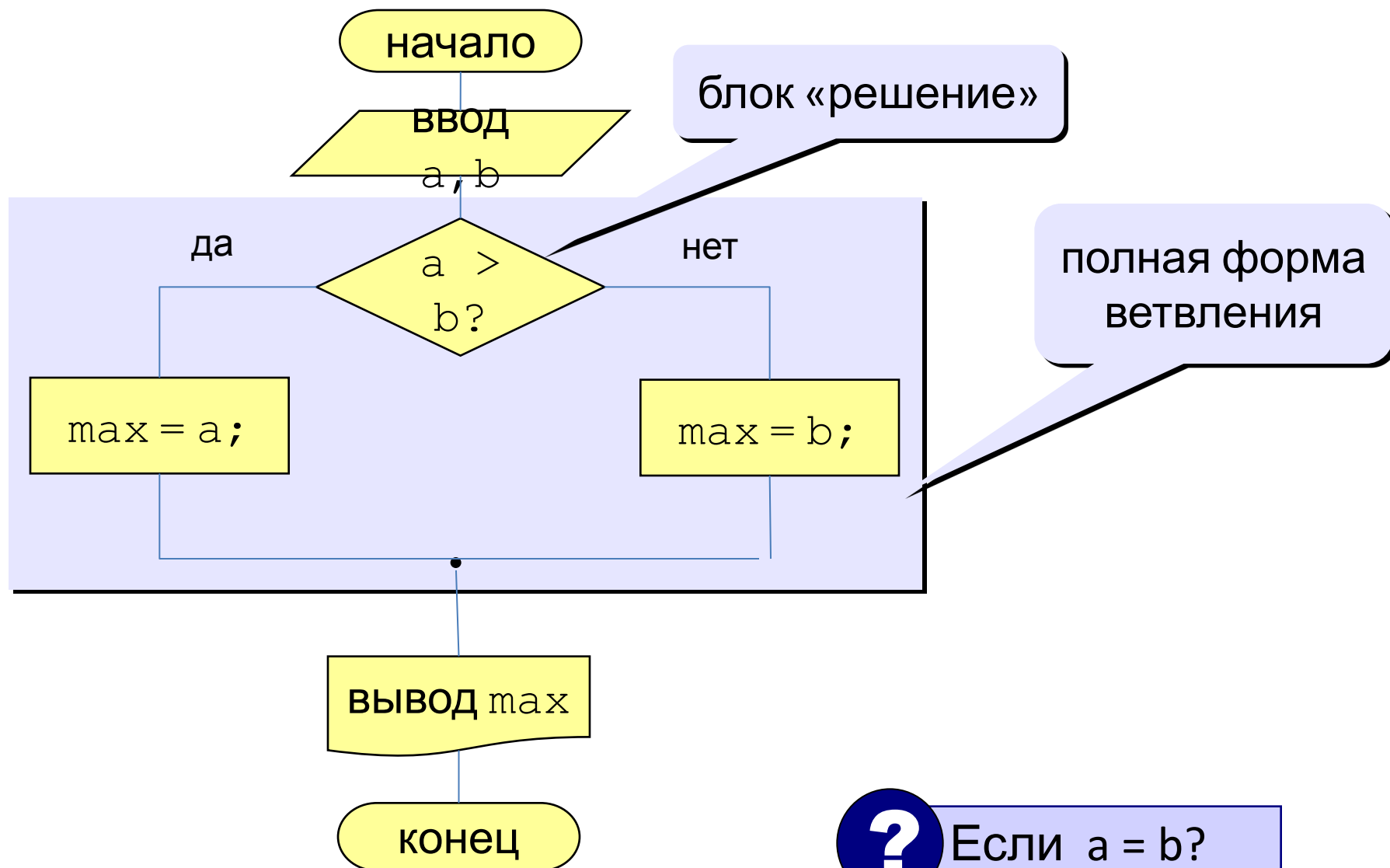
Задача. Ввести два целых числа и вывести на экран наибольшее из них.

Идея решения: надо вывести на экран первое число, если оно больше второго, или второе, если оно больше первого.

Особенность: действия исполнителя зависят от некоторых условий (*если ... иначе ...*).

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

Вариант 1. Блок-схема



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

Условный оператор

```
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 { }

Что неправильно?

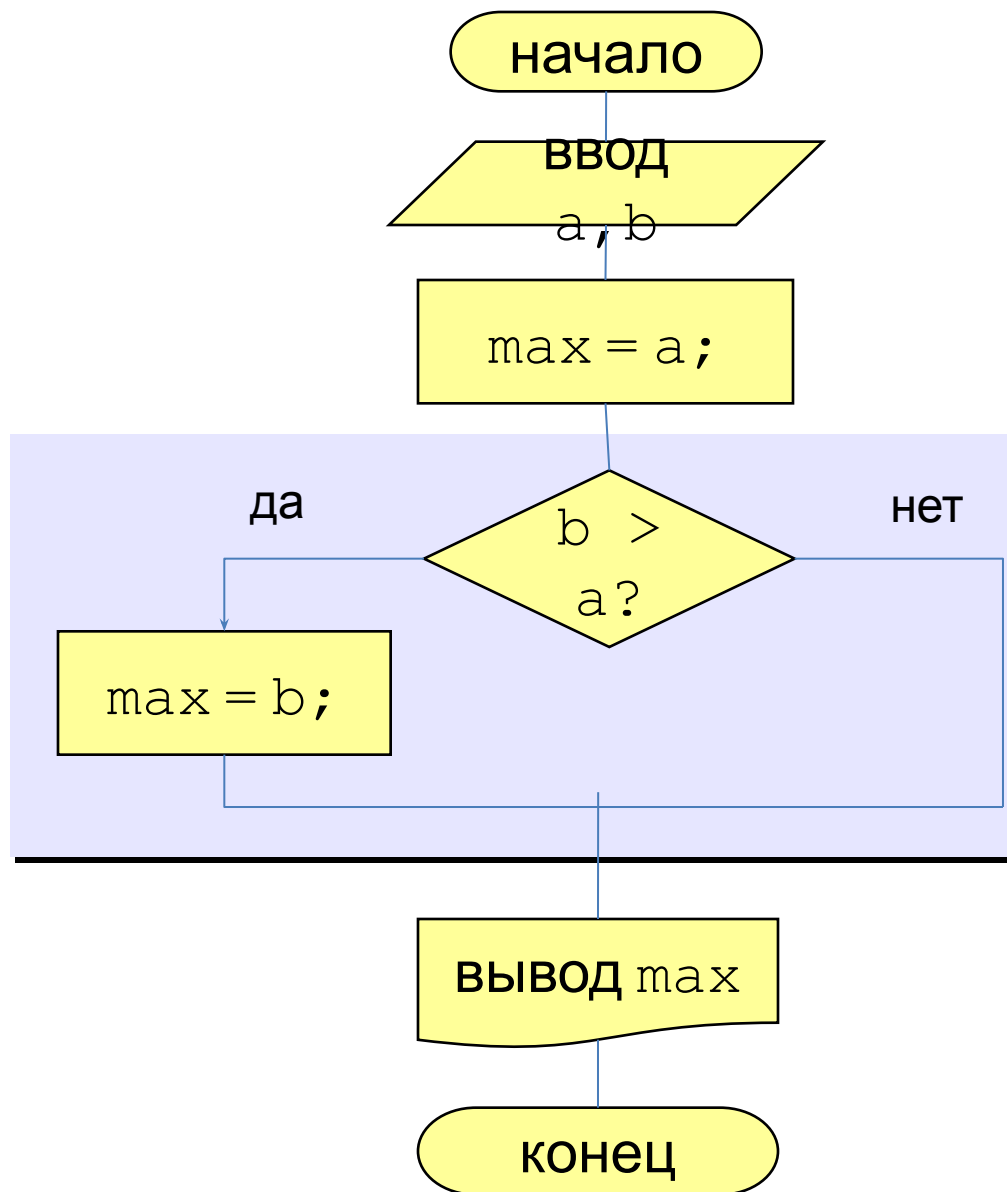
```
if ( a > b ) {  
    a = b;  
}  
else  
    b = a;
```

```
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;
```

Вариант 2. Блок-схема



неполная форма
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

Вариант 2Б. Программа

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

Задания

Ввести три числа и найти наибольшее из них.

Пример:

Введите три числа:

4 15 9

Наибольшее число 15

Ввести пять чисел и найти наибольшее из них.

Пример:

Введите пять чисел:

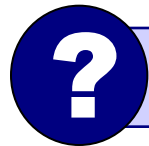
4 15 9 56 4

Наибольшее число 56

Сложные условия

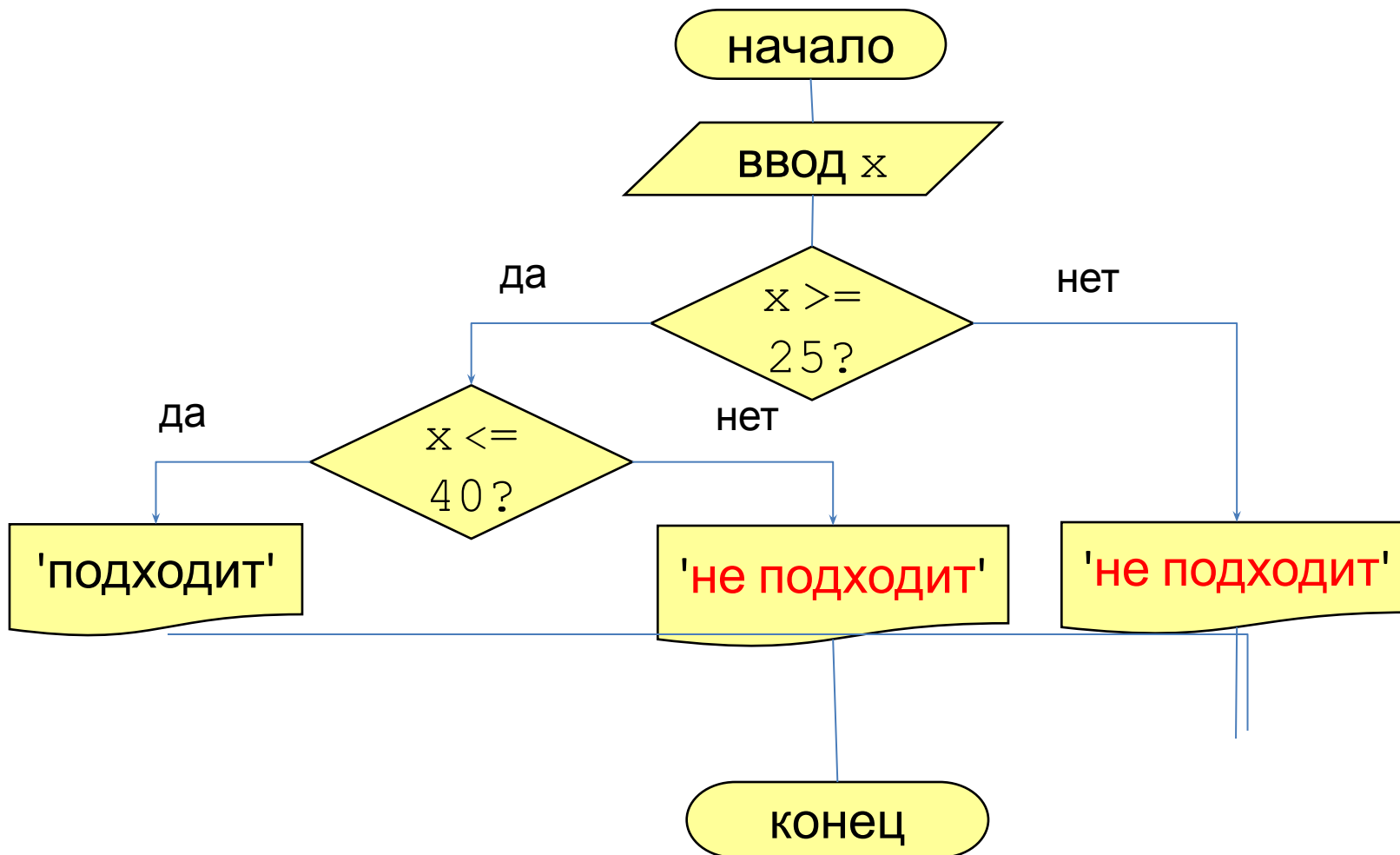
Задача. Фирма набирает сотрудников от 25 до 40 лет включительно. Ввести возраст человека и определить, подходит ли он фирме (вывести ответ «подходит» или «не подходит»).

Особенность: надо проверить, выполняются ли два условия одновременно.



Можно ли решить известными методами?

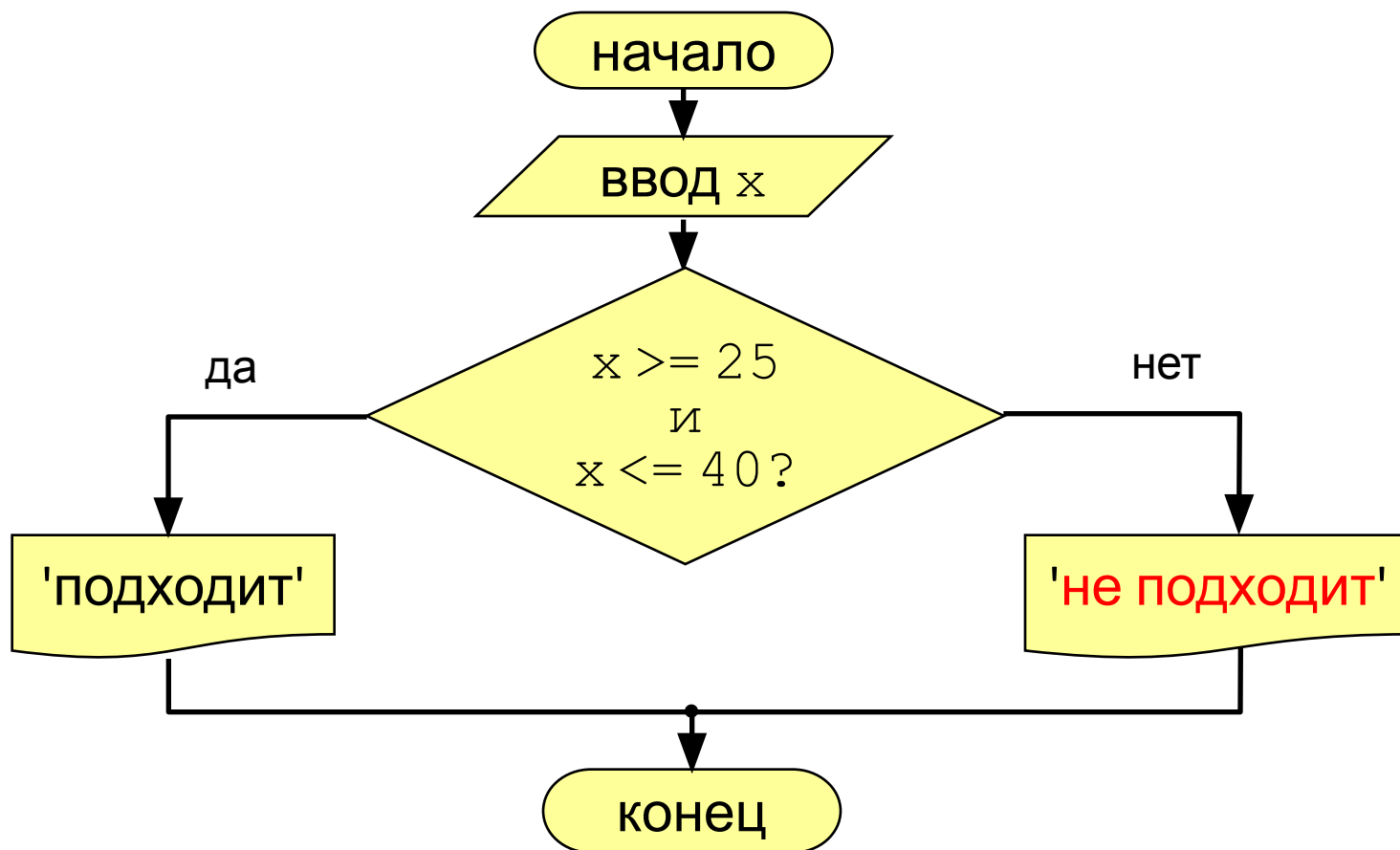
Вариант 1. Алгоритм



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");
}
```

Вариант 2. Алгоритм



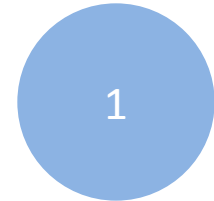
Вариант 2. Программа

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

сложное
условие

Вложенность условных конструкций

```
• if (условие1)
•     if (условие2)
•         оператор2;
•     else оператор3;
• else оператор3;
•     if (условие1) оператор1;
•         else if (условие2) оператор2;
•             else оператор3;
• if (условие1)
•     if (условие2)
• оператор1;
•     else оператор2;
```



оператор1;



СЛЕДУЕТ ЗНАТЬ:

- Если проверяемые условия не влияют друг на друга, т.е. последовательность их вычисления безразлична, то в тексте программы их располагают по принципу: условие, с наибольшей вероятностью принимающее значение true – на первом месте, с меньшей вероятностью на втором и т.д.
- При вложенности операторов каждое else соответствует ближайшему if.
- Конструкции со степенью вложенности более 2-3 избегать, из-за сложности анализа при отладке программы.



Сложные условия

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 условия

Порядок выполнения сложных условий:

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

Пример:

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

Сложные условия

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

$!(a > b)$

1

$a < b \ \&\& \ b < c$

1

$!(a \geq b) \ || \ c == d$

1

$a < c \ || \ b < c \ \&\& \ b < a$

1

$a > b \ || \ !(b < c)$

0

Для каких значений x истинны условия:

$x < 6 \ \&\& \ x < 10$

$x < 6 \ \&\& \ x > 10$

$x > 6 \ \&\& \ x < 10$

$x > 6 \ \&\& \ x > 10$

$x < 6 \ || \ x < 10$

$x < 6 \ || \ x > 10$

$x > 6 \ || \ x < 10$

$x > 6 \ || \ x > 10$

$(-\infty, 6)$	$x < 6$
\emptyset	
$(6, 10)$	
$(10, \infty)$	$x > 10$
$(-\infty, 10)$	$x < 10$
$(-\infty, 6) \cup (10, \infty)$	
$(-\infty, \infty)$	
$(6, \infty)$	$x > 6$

Задания

Ввести номер месяца и вывести название времени года.

Пример:

Введите номер месяца:

4

весна

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

Пример:

Введите возраст:

24

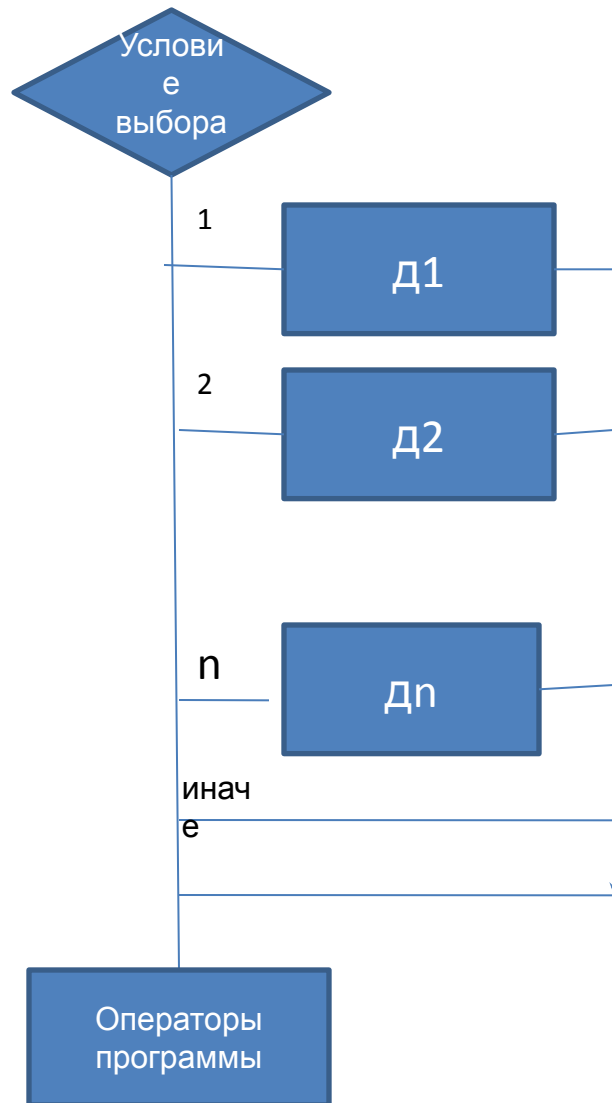
Вам 24 года

Введите возраст:

57

Вам 57 лет

Оператор поливариантного выбора



```
switch (выражение) {  
    case constant1: операторы;  
    break;  
    case constant2: операторы;  
    break;  
    ...  
    case constantn: операторы;  
    break;  
    default: операторы;  
}
```

Пример: проанализировать значение переменной *rez*, которая является выставленной оценкой

...

```
switch (rez) {  
    case 5: cout<<"оценка - отлично"; break;  
    case 4: cout<<"оценка - хорошо"; break;  
    case 3: cout<<"оценка -  
        удовлетворительно";          break;  
    case 2: cout<<"оценка -  
        неудовлетворительно"; break;  
    default : cout<<"неверное значение  
        rez";  
}
```

...