

C++

Урок 2

Разбор ДЗ

A hand-drawn blue oval frame with a slightly irregular, sketchy border, centered on the page. The word "Повторение" is written inside this frame in a bold, black, sans-serif font.

Повторение

Синтаксис условного оператора

Условный оператор if

Формат:

```
if (условие){  
    действие  
}
```

**В каком случае мы не ставим
фигурные скобки?**

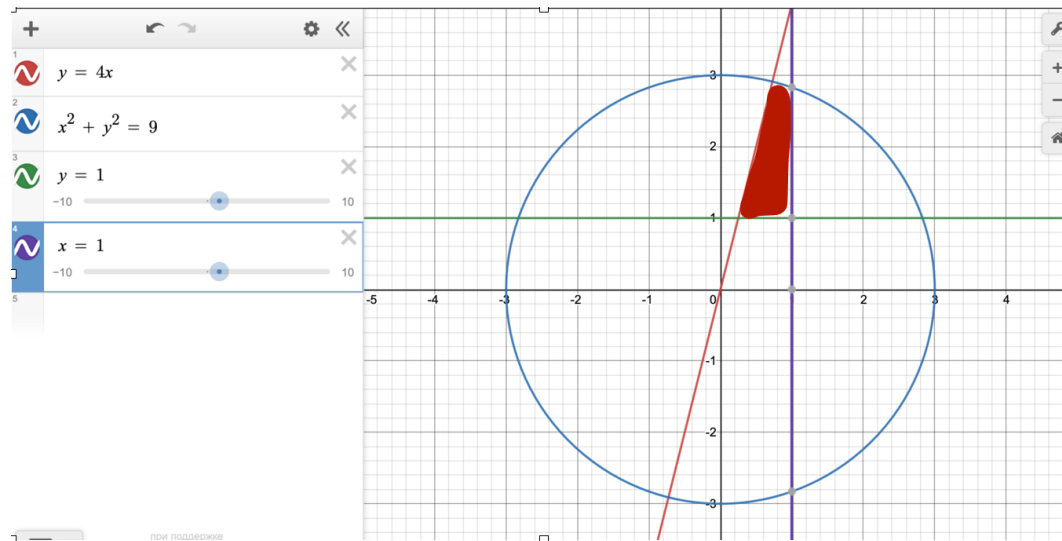
Условный оператор if

Формат:

```
if (a % 5 != 0) cout << " Не кратно";
```

При наличии одной операции

Задача



Условие: Пользователь вводит координаты точки (x и y), нужно вывести результат входит ли точка в эту область. (Область выделена красным цветом)

Булевы переменные

Булевы переменные

- Принимают 2 значения – **True (1)** или **False (0)**;

Применение: хранение значения логических выражений.

Булевы переменные

Вспомним данный пример:

```
int val;  
cin >> val;  
// [2, 8] диапазон  
if (val >= 2 && val <= 8){  
    cout << "Число принадлежит промежутку";  
}  
else {  
    cout << "Не принадлежит";  
}
```

**Где можно использовать
булеву переменную?**

Булевы переменные

Реализация через булеву переменную:

```
int val;  
cin >> val;  
bool flag = (val >= 2 && val <= 8);  
if (flag) cout << "Число принадлежит промежутку";  
//то же самое что и if (flag != 0)  
else cout << "Не принадлежит";  
}
```

Задача

Пользователь вводит в консоль число;

Задача: проверить, что введенное число входит в данные промежутки, интервалы и полуинтервалы $(-4; 9) \cup \{18\} \cup (65; 90]$; (**через булеву переменную**)

**Что такое тернарный
оператор?**

Тернарный оператор

тернарный оператор

Формат:

(условие) ? если истина : если ложь;

```
int a;  
cin >> a;  
int rez = (a % 2 == 0)? a: a*(-1);
```

Применение: краткая форма записи if else

тернарный оператор

Для трех исходов:

```
(a > 0)? cout <<"Больше":(a == 0)? cout <<"Равно 0": cout <<"Меньше";
```

На выводе: больше, меньше или равно 0

Задача

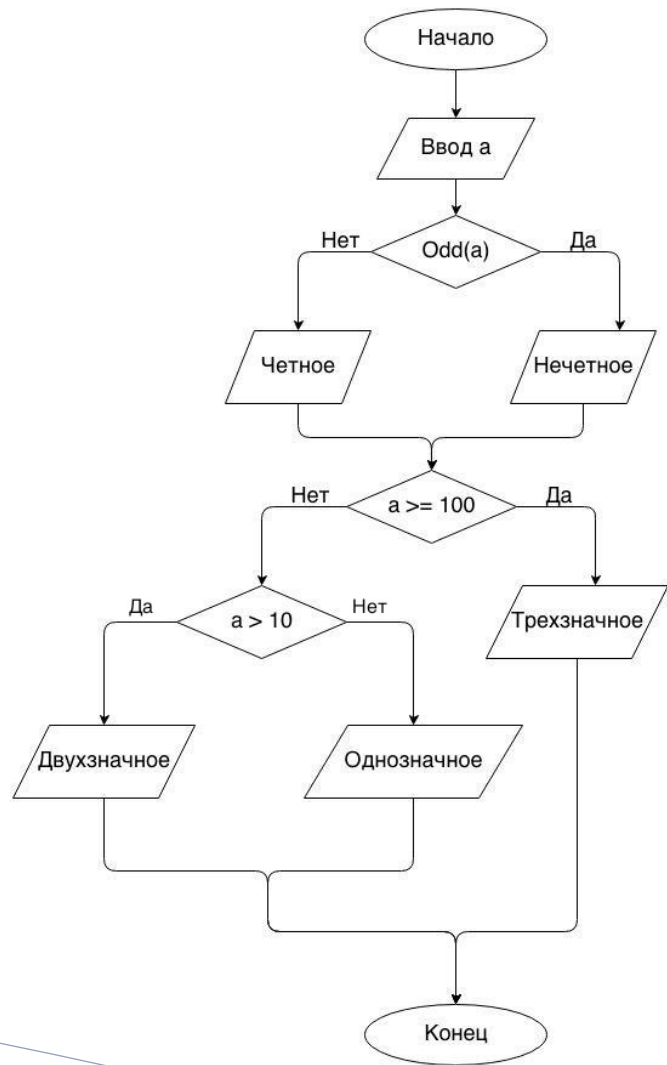
Пользователь вводит в консоль натуральное число.

Задача: проверить, что второй разряд справа числа четный.

(решение через тернарный оператор)



Вложенный условный оператор





Задача

Умпа-Лумп решил взять с собой в лилипутию песеля. При прохождении иноплатентного таможенного контроля были выявлены следующие ограничения на ввоз домашних животных:

- вес $\in [1.5 - 6]$ кг;
- рост $\in [30 - 90]$ см;
- длина $\in [40 - 150]$ см;
- Должна быть переноска. (булевая переменная)

Задача: Выведите сообщение таможенного пограничника. Варианты ответа: Пропускаем! Не подходит вес! Не подходит длина! Нет переноски!

Оператор Switch

Оператор выбора

Оператор выбора

Формат:

```
switch ( <переменная> ) {  
  case значение1:  
    Выполнить если <переменная> == значение1  
    break;  
  case значение2:  
    Выполнить если <переменная> == значение2  
    break;  
  ...  
  default:  
    выполнить, если ни один вариант не подошел  
    break;  
}
```


Оператор выбора

Пример:

```
unsigned short int mes;  
cin >> mes;  
  
switch(mes) {  
    case 1:  
        cout << "Январь";  
        break;  
    case 2:  
        cout << "Февраль";  
        break;  
  
    case 3:  
        cout << "Март";  
        break;  
    default:  
        cout << "Вы не правильно указали";  
}
```

Оператор выбора

```
unsigned short int mes;  
cin >> mes;  
if (mes == 12) mes = 0;  
switch (mes) {  
case 0 ... 2: cout << "Зима"; break;  
case 3 ... 5: cout << "Весна"; break;  
case 6 ... 8: cout << "Лето"; break;  
case 9 ... 11: cout << "Осень"; break;  
default: cout << "Неправильно введенный месяц";  
break;  
}
```

Запись диапазонов значений

Задача

Пользователь вводит в консоль число; Это число проверяется на наличие призов. [1 - 15] – **ничего**, [16 - 18] – **конфетка**, [20 - 24] – **сковородка**; 25 – **главный приз**;

Задача: выведите сообщение о том, что выиграл пользователь;

Инкремент и декремент

Что такое инкремент?

Что такое декремент?

Инкремент и декремент

Увеличим переменную a на **единицу**;

```
int a = 0;  
//увеличим a на 1  
a = a + 1; // a = 1
```

Инкремент и декремент

Увеличим переменную a на **единицу**;

```
int a = 0;  
//увеличим a на 1  
a = a + 1; // a = 1
```

Некрасивая запись



Инкремент и декремент

Увеличим переменную a на **единицу**;

```
int a = 0;  
//увеличим a на 1  
a += 1; // a = 1
```

Инкремент и декремент

Увеличим переменную a на **единицу**;

```
int a = 0;  
//увеличим a на 1  
a += 1; // a = 1
```

Уже лучше, но все равно не то



Инкремент и декремент

Увеличим переменную a на **единицу**;

```
int a = 0;  
//увеличим a на 1  
a++; // инкремент a = 1
```



Инкремент и декремент

Увеличим переменную a на **единицу**;

```
int a = 0;  
//увеличим a на 1  
a++; // инкремент a = 1
```

Где a++ – операция **инкрементирования**;

Инкремент и декремент

```
a++; // инкрементирование (увеличение на 1)  
a--; // декрементирование (уменьшение на 1)
```

Инкремент и декремент

Пример:

```
int number, cnt;  
cnt = 0;  
cin >> number;  
if (number % 2 == 0) cnt++;
```

Задача

Пользователь вводит в консоль 4 целых числа;

Задача: посчитать количество кратных 8 чисел и вывести, если таких чисел нет, то вывести сообщение “не найдено”.

Префикс и постфикс

Что это такое?

Префикс и постфикс

постфикс

$a++;$

префикс

$++a;$

Префикс и постфикс

Отличие:

- **Префикс** - это математическая запись, в которой операторы предшествуют своим операндам (аргументы операции);
- **Постфикс** - это математическая запись, в которой операторы следуют за своими операндами.

Префикс и постфикс

Пример:

```
int a = 42;  
cout << a++<<endl; // 42  
cout << a <<endl; // 43  
  
int b = 42;  
cout << ++b<<endl; // 43  
cout << b <<endl; // 43
```

A hand-drawn blue oval frame with a double-line border, centered on the page. The word 'Циклы' is written inside this frame.

Циклы

Где применяются циклы?

Циклы

Цикл — это конструкция языка, которая позволяет выполнить определенные действия заданное количество раз. (итераций).

Итерация – разовое выполнение операций, находящихся в теле цикла.

Тело цикла - набор действий (программных операций), которые подразумеваются к повторению раз за разом при работе цикла.

**Какие виды циклов вы
знаете?**

Виды циклов

Виды:

пока

делай пока

счетный

Цикл пока – while(условие);

Цикл делай пока – do .. while(условие);

Счетный цикл – for.