

# Условия, циклы, массивы



# Условные операторы

- Практически любая программа содержит ветвление
- Один вариант разбора условий: if - else if - else
- В качестве выражения для условия принимается что-то, имеющее тип boolean
- Второй вариант switch() { case }
- В качестве выражения для разбора может принимать численных и строковые типа

# Циклы for, while, do-while

- Выполнение большого количества однотипных действий
- Обработка больших объемов однотипных данных
- Руками можно и не дописать до конца
- Общий принцип работы циклов: повторение действий, указанных в теле цикла до тех пор, пока выполняется указанное условие
- Циклы могут быть вложенные!
- Выполнение цикла можно прервать командой `break`;
- Пропустить текущую итерацию можно с помощью команды `continue`

## Немного синтаксического сахара

Для удобства в языке есть укороченные бинарные операторы.

Вместо  $a = a + 15$ ; мы можем написать  $a += 15$ .

То же самое с остальными операторами:

- $b -= 10$ ;
- $b *= 12$ ;
- $b /= 2$ ;

Также увеличение и уменьшение элемента на единицу вынесены в отдельный оператор инкремента/декремента:

$i++$

$j--$

# Задачи на простые циклы

- Посчитать сумму чисел от 1 до  $n$
- Вычислить факториал числа  $n$ ;  $n! = 1 * 2 * \dots * n$
- Возвести число  $n$  в степень  $k$ ;  $result = n * n * \dots * n$
- Посчитать сумму цифр числа; 123 -> 6
- Проверить, делится ли бооооольшооооое число на 3
- Проверить, является ли строка палиндромом; abcba

## **Для себя:**

- Сумма квадратов цифр
- Является ли число степенью двойки?
- Проверить, что в строке только латинские буквы нижнего регистра

# Массивы

- Необходимо хранить большое количество однотипной информации и иметь к ней быстрый доступ
- Индексация с нуля, так же, как в строках
- Обращение к элементам по индексу
- Изменение элементов также по индексу
- Бывают многомерные массивы

# Задачи на работу с массивами

- Распечатать на экране массив в обратном порядке
- Найти максимальный и минимальный элемент во введенных данных
- Ввести n строк и напечатать строку максимальной длины, если таких несколько, печатать последнюю введенную
- Найти элемент с максимальной суммой цифр
- Отсортировать элементы по возрастанию (пузырёк)

## **Для себя:**

- Проверить, является ли массив упорядоченным по возрастанию
- Заменить все числа на их максимальный целый делитель, меньше самого числа

## Приятные встроенные плюшки

Самые распространенные операции с массивами (сортировка, поиск элемента и т. д.) уже реализованы в стандартной библиотеке.

Эти функции находятся в классе `Arrays`.

Они как правило имеют понятные названия и подробную документацию. Используйте их!