

Курс "Конструирование программного обеспечения"

Семинар 02.

План семинара

1. "Горячие клавиши" IntelliJ IDEA (Menu: Help -> [Keymap Reference](#)).
2. Обсуждение [Java Code Conventions](#)
3. Ключевые слова Java
4. Примитивы
5. Использовать режим отладки "step-by-step" среды IntelliJ IDEA на примерах.
6. Пройти Q&A (Questions and Answers) для семинара 2.
7. Создание приложения конвертации расстояния (мили в километры и наоборот)

"Горячие клавиши" IntelliJ IDEA

Menu: Help -> [Keymap Reference](#)

Соглашения по оформлению кода Java

[Java Code Conventions](#)

Ключевые слова

Ключевые слова - зарезервированные слова, которые нельзя использовать для именования классов, переменных и методов. Подробнее: [Keywords Java](#) или [Список ключевых слов](#)

Пример не ключевого слова:

```
int then = 101;
```

Пример ошибки с использованием ключевого слова:

```
int if = 101;
```

Примитивы

Все классы-оболочки примитивов - неизменяемые классы.

Java примитивы:

- Целочисленные типы
 - byte
 - short
 - char ([UTF-16 \(буквы и цифры\)](#)).
 - int
 - long
- Типы с плавающей точкой [IEEE 754-1985](#)
 - float
 - double

- Логический тип
 - boolean

Константы

Постоянное значение, которое задается один раз в начале работы программы.

```
public static final int MAGIC_NUMBER = 2;
public static final double USA_MILE = 1609.34;
```

Неизменяемые и изменяемые классы

Для неизменяемого (immutable) класса, после его создания, состояние не может быть изменено. Например: Integer, BigInteger, BigDecimal, Float, String и т.д. Для изменяемого (mutable) класса, его состояние может меняться после создания. Например: java.awt.Point.

Инициализация примитивов

[Инициализация примитивов](#)

```
private static Integer k;
private static int z;
private static int i;

public static void nullIntegerExample() {
    i = k + z;
    System.out.println(i);
}
```

Операторы

[Operators](#) [Integer Operations](#) [Приоритет операций](#)

- Логическое исключающее ИЛИ, (XOR), ^ (Возвращает true, если операнды – разные)
System.out.println(9^3);
- Тернарный оператор Примеры записи:

```
alpha = (aLongBooleanExpression) ? beta : gamma;
```

```
alpha = (aLongBooleanExpression) ? beta
: gamma;
```

```
alpha = (aLongBooleanExpression)
? beta
: gamma;
```

Инкремент и декремент

- инкремент ++
- декремент --

```
i = 0;
i++; // => 1
```

```
i++; // => 2
i--; // => 1
i--; // => 0
```

Есть постфиксная (i++) и префиксная (++i) формы:

- префиксная нотация – сначала происходит изменение переменной, а потом возврат получившегося значения.
- постфиксной нотация – после изменения возвращается то значение, которое было до изменения переменной.

```
var x = 3;
System.out.println(++x); /* => 4 */
System.out.println(x);   /* => 4 */
System.out.println(x++); /* => 4 */
System.out.println(x);   /* => 5 */
```

Примеры byte

Пример 1.

```
byte b1 = 127;
b1 +=1;
System.out.println(b1);
```

Примеры с плавающей точкой

```
float f = 29.1f;
double d = 29.1;
System.out.println( f == d );
f = 29.1f;
d = 29.1f;
System.out.println( f == d );
```

Массивы

[Array Variables Varargs](#)

Задание для написания первого приложения

Задание: Реализовать следующие методы:

- Вариант 1: перевод миль в километры и наоборот (Mile2Km и обратно – Km2Mile);
- Вариант 2: перевод градусов (Celsius2Fahrenheit и обратно – Fahrenheit2Celsius).

Требование:

- Входной параметр получать из командной строки. Результат печатать в консоль.

Создание *.jar в IntelliJ IDEA

- File -> Project Structure (Ctrl+Alt+Shift+S) -> Artifacts -> Настроить проект
- Build -> Build Artifacts [Подробнее](#)

Дополнительно: [Создание *.jar без среды разработки](#)