



Михаил Абрамович Ройтберг

зав. кафедрой АТП

27.12.1952 - 16.08.2017

Программирование на Java

Лектор: Яковлев Виктор Вадимович

Автор материалов курса: (С) Лаврентьев Федор Сергеевич

МФТИ, 2016-2017

Мы изучим

1. Технику программирования на Java
2. Популярные фреймворки экосистемы Java
3. Основы объектно-ориентированного дизайна
4. Основы дизайна распределенных приложений
5. Работу из Java с сопутствующими технологиями:
 1. Базами данных
 2. Протоколами сериализации
 3. REST-сервисами

Курс состоит из

- 17 лекций
- 17 семинаров
- 5 практических работ (от 0 до 2 баллов за каждую)
- 1 устного зачета (от 0 до 2 баллов)

Практические работы

- Работающее задание с идеальным кодом — 2 балла
- Работающее задание с посредственным кодом — 1 балл
- Разбор задания — на последнем семинаре месяца
- Не успел до разбора — получаешь не больше 1 балла
- Дедлайн работы — 23:59 MSK последнего дня месяца
- **Не успел до дедлайна — получаешь 0 (ноль) баллов**
- У работ 4 и 5 нет дедлайна — успехов в Новом Году!

Единственный способ
научиться программировать —
много программировать

1. ОСНОВЫ

Лектор: Яковлев Виктор Вадимович

Автор материалов курса: (С) Лаврентьев Федор Сергеевич

МФТИ, 2016-2017

История появления Java

1991-1993, Project Green, Oak

- Sun Microsystems, Джеймс Гослинг
- Разработка ПО для бытовой техники и PDA
- Проблемы:
 - Обилие процессорных архитектур
 - Существенные различия между различными ОС и платформами
 - Неустойчивость программ с прямой адресацией
 - Массовые проблемы с безопасностью

Ключевые решения

- Забудьте про аппаратные платформы и операционные системы
- Есть только виртуальная машина (JVM)
- JVM осуществляет всё взаимодействие с окружением
- Программа взаимодействует только с JVM
- JVM API одинаковый для всех окружений
- Программа компилируется в инструкции для JVM

1994-1997, HotJava – Java 1.1

- Разворот в сторону интернет-рынка
- 1994 – браузер HotJava
- 1995 – прообраз Java Applets, управление правами внутри JVM
- 1996 – Java 1.02, единый JVM API для всех платформ
- 1997 – Java 1.1
 - AWT – GUI
 - JavaBeans – стандарт проектирования
 - RMI, JDBC, Serialization – клиент-серверное взаимодействие
 - Reflection – управление кодом в Runtime

1998, Java 1.2 aka J2SE

- JIT-compiler
 - Collections framework
 - Swing GUI
 - Java Applets
-
- Java 2 Enterprise Edition (J2EE)
 - Java 2 Micro Edition (J2ME)

Основные понятия Java

Действующие лица

- Byte code
- Heap
- Stack
- Native binaries

Java Runtime Environment (JRE)

- Java Virtual Machine (JVM)
- Just-in-time (JIT) compiler
- Java Runtime Library (rt.jar)
- Garbage Collector (gc)

Java Software Development Kit (SDK)

- Java 2 Standard Edition (Java SE, J2SE) API Libraries
- `javac` – компилятор
- `jar` – архиватор
- `keytool` – управление сертификатами безопасности
- `jvisualvm` – профайлер
- `jhat` – анализатор памяти
- `jstack` – съемщик `stack trace`'ов
- ...

Спецификация Java

- Java Virtual Machine specification
<http://docs.oracle.com/javase/specs/jvms/se8/html/index.html>
- Java Language Specification
<http://docs.oracle.com/javase/specs/jls/se8/html/index.html>
- Java API
<http://docs.oracle.com/javase/8/docs/api/index.html>
- Code Conventions
<http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

Примитивы

Встроенные типы

- Примитивы: `boolean`, `char`, `byte`, `short`, `int`, `long`, `float`, `double`
 - Хранятся либо на стеке, либо как параметры объекта
 - При передаче как параметры значение копируется
 - Для хранения в куче производится упаковка (boxing)
- `void` — тип отсутствия значения
- `Reference` — ссылка на объект в куче
 - При передаче как параметр значение ссылки копируется

Boolean

- В JVM не реализован
- Представляется как `int`
 - `false` – 0
 - `true` – 1
- Массив `boolean` – это массив `byte`

Char

- Представлен как unsigned int16
- Представляет символы в кодировке UTF-16

Целые числа

- byte – signed int8
- short – signed int16
- int – signed int32
- long – signed int64 (два слота на стеке JVM)
- Числа с плавающей точкой

Числа с плавающей запятой

- float – 32 бита
- double – 64 бита (два слота на стеке JVM)
- Специальные значения:
 - NaN
 - Сравнение с NaN всегда false: $(x == x) == \text{false}$, если x – NaN
 - (int) Double.NaN == 0
 - Положительный/отрицательный ноль
 - $0.0 == -0.0$
 - Положительная/отрицательная бесконечность
 - `Double.POSITIVE_INFINITY == (1.0 / 0.0)`
 - `Double.NEGATIVE_INFINITY == (1.0 / -0.0)`

Void

- Значения не ожидается
- Невозможно инстанцировать

Объекты и классы

Объекты

- Объект – абстракция элемента, участвующего в решении задачи
- Всё* является объектом
- Объект состоит из других объектов
- У объекта есть тип (объект является экземпляром класса)
- Все объекты типа могут получать одинаковые сообщения
- Программа – это группа объектов, общающихся с помощью сообщений

Классы

```
public class Student {  
    private final String name;  
    private final int age;  
  
    public Student(String name, int age) {  
        this.age = age;  
        this.name = name;  
    }  
  
    public int getAge() {  
        return age;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

Ссылки (Reference)

- Переменная содержит ссылку на объект в куче (не сам объект)
- Ссылка типизирована, т.е. несет информацию о классе объекта
- Пользовательский код работает только со ссылками
- Прямая работа с памятью (указатели) недоступна
- Значение ссылки по умолчанию – null

Методы класса Object

- equals(), hashCode()
- clone()
- toString()
- finalize()
- wait(), notify(), notifyAll()
- getClass()