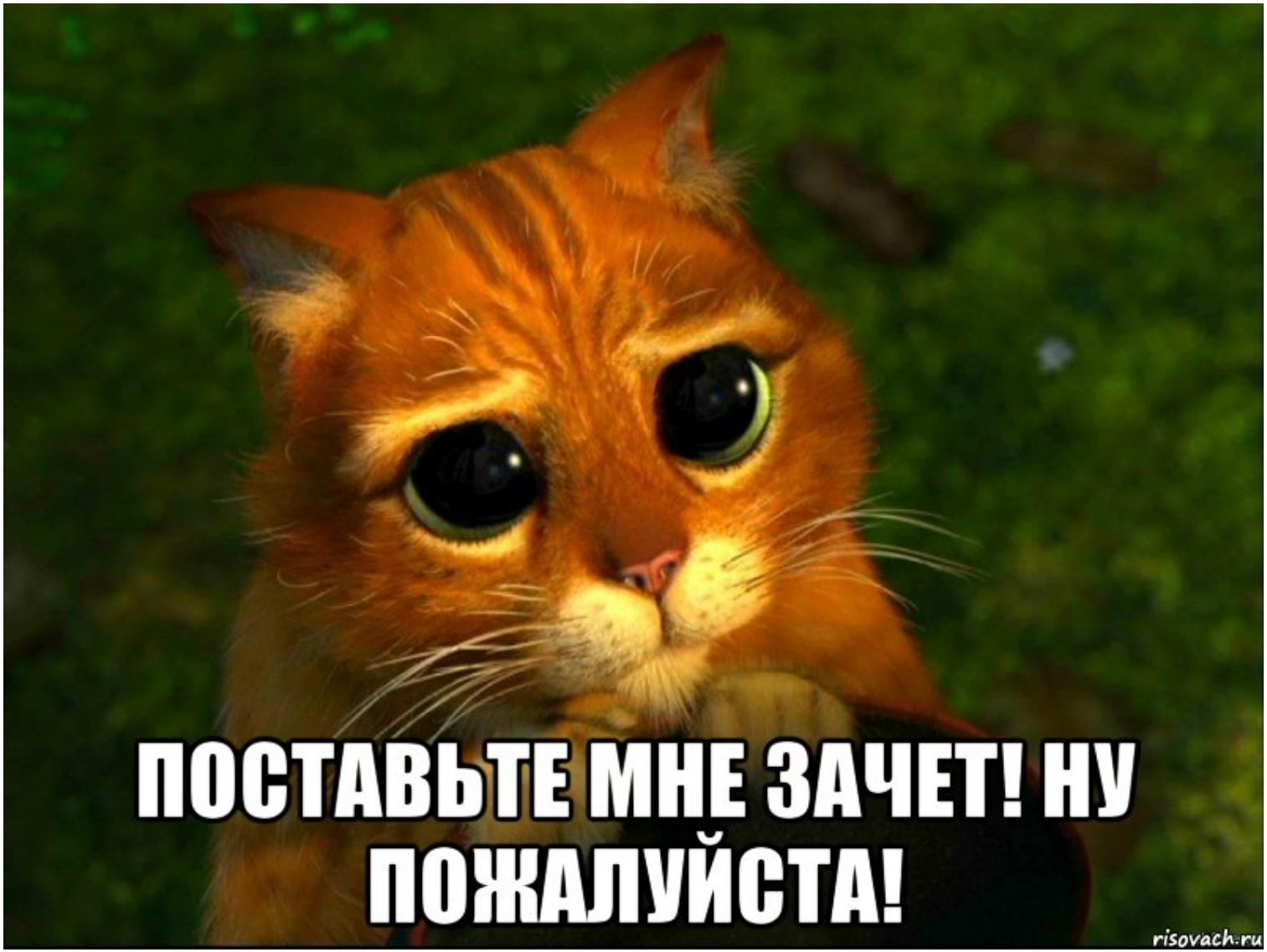
Введение в Java

Алексей Владыкин

12 сентября 2013


ПОСТАВЬТЕ МНЕ ЗАЧЕТ! НУ ПОЖАЛУЙСТА!
risovach.ru

О курсе

- Лекции, практические сессии, домашние задания
- Доступны материалы прошлого года (презентации и видео): `http://compscicenter.ru/program/course/java2012`

Источники

Bruce Eckel

Thinking in Java

Joshua Bloch

Effective Java

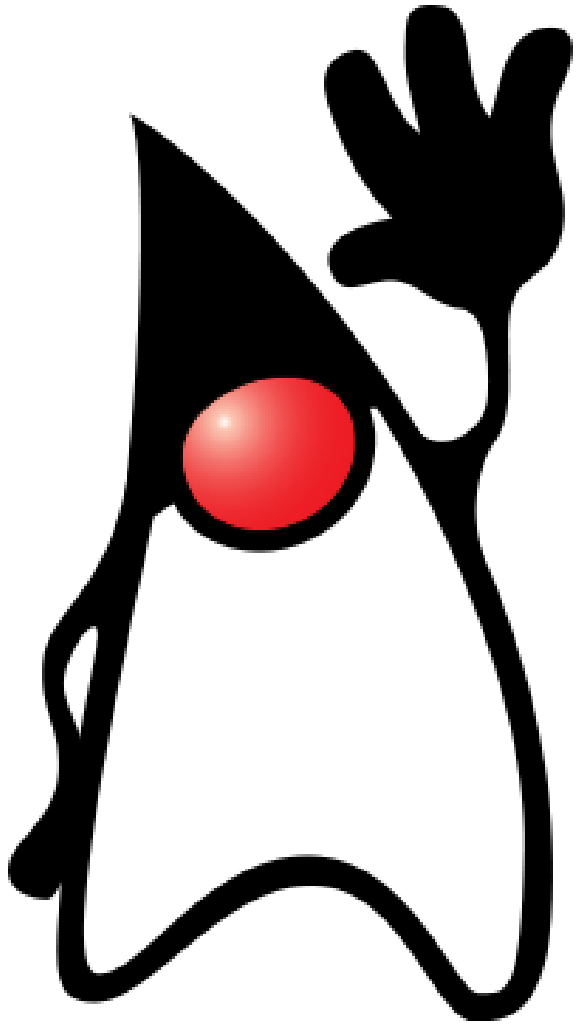
The Java Language Specification

`http://docs.oracle.com/javase/specs/jls/se7/html/index.html`

The Java Virtual Machine Specification

`http://docs.oracle.com/javase/specs/jvms/se7/html/index.html`


Java™



- Один из самых популярных языков программирования
  По индексу TIOBE на сентябрь 2013 — самый популярный
- Используется в смарт-картах (в том числе SIM-картах), в мобильных устройствах, на настольных компьютерах, на серверах

- Java — это не только ~~ценный мех~~ язык программирования, но и. . .

- Обширная стандартная библиотека
- Сторонние библиотеки и фреймворки
- Инструменты разработки (сборка, тестирование)
- Методология ООП, паттерны проектирования
- Платформа для альтернативных языков (Clojure, Groovy, JRuby, Jython, Kotlin, Scala)

Редакции Java

- Standard Edition (SE)
- Micro Edition (ME)
  подмножество SE + специфические библиотеки
- Enterprise Edition (EE)
  SE + дополнительные библиотеки и возможности
- Java Card
  сильно урезанная версия SE, изменения в виртуальной машине
- JavaFX
  инструментарий для создания интерактивных графических приложений
- Android

JRE/JDK

- Java Runtime Environment (JRE)
  виртуальная машина и стандартная библиотека классов для запуска скомпилированных программ

- Java Development Kit (JDK)
  набор инструментов для разработчиков (компилятор), включает в себя JRE

Реализации Java

- Oracle Java
  `http://java.oracle.com/`
- OpenJDK
  `http://openjdk.java.net/`
- IBM J9
- Azul Zing
- Apache Harmony (retired)
- еще несколько десятков

James Gosling

1991 внутренний проект Sun Microsystems по созданию платформы для разработки встраиваемых систем — Green Project;
вместо C++ решили создать новый язык, названный Oak

1991 внутренний проект Sun Microsystems по созданию платформы для разработки встраиваемых систем — Green Project;
вместо C++ решили создать новый язык, названный Oak

1992 первое демонстрационное устройство на новой платформе — PDA Star7

1991 внутренний проект Sun Microsystems по созданию платформы для разработки встраиваемых систем — Green Project;
вместо C++ решили создать новый язык, названный Oak

1992 первое демонстрационное устройство на новой платформе — PDA Star7

1993 попытка занять нишу TB-приставок для кабельного телевидения

1991 внутренний проект Sun Microsystems по созданию платформы для разработки встраиваемых систем — Green Project;
вместо C++ решили создать новый язык, названный Oak

1992 первое демонстрационное устройство на новой платформе — PDA Star7

1993 попытка занять нишу TB-приставок для кабельного телевидения

1994 фокус на разработке интерактивных приложений (апплетов) для веб-страниц;
язык переименован в Java

1991 внутренний проект Sun Microsystems по созданию платформы для разработки встраиваемых систем — Green Project;
вместо C++ решили создать новый язык, названный Oak

1992 первое демонстрационное устройство на новой платформе — PDA Star7

1993 попытка занять нишу TB-приставок для кабельного телевидения

1994 фокус на разработке интерактивных приложений (апплетов) для веб-страниц;
язык переименован в Java

1996 Java Development Kit 1.0

1996 Java Development Kit 1.0

1996 Java Development Kit 1.0

1997 JDK 1.1, JIT-компиляция

1996 Java Development Kit 1.0

1997 JDK 1.1, JIT-компиляция

1998 J2SE 1.2, «Java 2», разделение на ME/SE/EE

1996 Java Development Kit 1.0

1997 JDK 1.1, JIT-компиляция

1998 J2SE 1.2, «Java 2», разделение на ME/SE/EE

2000 J2SE 1.3

1996 Java Development Kit 1.0

1997 JDK 1.1, JIT-компиляция

1998 J2SE 1.2, «Java 2», разделение на ME/SE/EE

2000 J2SE 1.3

2002 J2SE 1.4

1996 Java Development Kit 1.0

1997 JDK 1.1, JIT-компиляция

1998 J2SE 1.2, «Java 2», разделение на ME/SE/EE

2000 J2SE 1.3

2002 J2SE 1.4

2004 J2SE 5.0, изменение нумерации

1996 Java Development Kit 1.0

1997 JDK 1.1, JIT-компиляция

1998 J2SE 1.2, «Java 2», разделение на ME/SE/EE

2000 J2SE 1.3

2002 J2SE 1.4

2004 J2SE 5.0, изменение нумерации

2006 Java SE 6, уход от понятия «Java 2»

1996 Java Development Kit 1.0

1997 JDK 1.1, JIT-компиляция

1998 J2SE 1.2, «Java 2», разделение на ME/SE/EE

2000 J2SE 1.3

2002 J2SE 1.4

2004 J2SE 5.0, изменение нумерации

2006 Java SE 6, уход от понятия «Java 2»

2011 Java SE 7

1996 Java Development Kit 1.0

1997 JDK 1.1, JIT-компиляция

1998 J2SE 1.2, «Java 2», разделение на ME/SE/EE

2000 J2SE 1.3

2002 J2SE 1.4

2004 J2SE 5.0, изменение нумерации

2006 Java SE 6, уход от понятия «Java 2»

2011 Java SE 7

2014 Java SE 8


KEEP
CALM
AND
LOVE
JAVA

Виртуальная машина и байткод

- Подход C/C++:
  исходный код $\rightarrow$ машинный код $\rightarrow$ процессор
  - программа работает только на той платформе, под которую она скомпилирована

Виртуальная машина и байткод

- Подход C/C++:
  исходный код $\rightarrow$ машинный код $\rightarrow$ процессор
  - программа работает только на той платформе, под которую она скомпилирована
- Подход Java:
  исходный код $\rightarrow$ байткод виртуальной машины $\rightarrow$ виртуальная машина $\rightarrow$ процессор
  - программа работает на любой платформе, где есть виртуальная машина Java
  - “Write once, run anywhere!”

Виртуальная машина и байткод

- Как быстро работает виртуальная машина?

Виртуальная машина и байткод

- Как быстро работает виртуальная машина?
- Интерпретация байткода на порядок (10–20 раз) медленнее исполнения аналогичного машинного кода...

Виртуальная машина и байткод

- Как быстро работает виртуальная машина?
- Интерпретация байткода на порядок (10–20 раз) медленнее исполнения аналогичного машинного кода...
- но есть Just-In-Time компиляция
  - виртуальная машина компилирует байткод в машинный код
  - используется с JDK 1.1

Виртуальная машина и байткод

- Как быстро работает виртуальная машина?
- Интерпретация байткода на порядок (10–20 раз) медленнее исполнения аналогичного машинного кода...
- но есть Just-In-Time компиляция
  - виртуальная машина компилирует байткод в машинный код
  - используется с JDK 1.1
- а также HotSpot
  - адаптивный оптимизирующий JIT-компилятор
  - используется с JDK 1.3

Виртуальная машина и байткод

- Как быстро работает виртуальная машина?
- Интерпретация байткода на порядок (10–20 раз) медленнее исполнения аналогичного машинного кода...
- но есть Just-In-Time компиляция
  - виртуальная машина компилирует байткод в машинный код
  - используется с JDK 1.1
- а также HotSpot
  - адаптивный оптимизирующий JIT-компилятор
  - используется с JDK 1.3
- в результате Java 7 всего в 1.5–2 раза медленнее C, а в некоторых тестах не хуже или даже быстрее!

Сборка мусора

- Подход C/C++:
  выделил память $\rightarrow$ поработал $\rightarrow$ освободил память
  - всё управление памятью в руках программиста

Сборка мусора

- Подход C/C++:
  выделил память $\rightarrow$ поработал $\rightarrow$ освободил память
  - всё управление памятью в руках программиста
- Подход Java:
  выделил память $\rightarrow$ поработал $\rightarrow$ молодец
  - виртуальная машина считает ссылки на каждый объект
  - освобождает память, когда ссылок больше нет

Безопасность

- Верификация байткода
  - некорректный байткод будет отвергнут перед исполнением
- Автоматическое управление памятью
  - нет арифметики указателей
  - невозможно испортить память
- Встроенный механизм управления правами
  - можно запустить код в «песочнице» без доступа к файлам, к сети, без возможности создавать потоки и т. п.

Многопоточное и распределенное программирование

- Многопоточность
  - встроенная поддержка потоков
  - богатая библиотека примитивов синхронизации
- Распределенность
  - встроенные сетевые возможности
  - пересылка данных и объектов по сети
  - работа с удаленными объектами (RMI)


HELLO
WORLD

HelloWorld.java

```
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello world!");
  }
}
```

- Java Coding Conventions
  http://www.oracle.com/technetwork/java/codeconv-138413.html

HelloWorldWithComments.java

```
/**
 * Prints "Hello world!" and terminates.
 * @author Alexey Vladykin
 */
public class HelloWorldWithComments {
  /* Canonical example program since 1978,
     many thanks to K&R :) */
  public static void main(String[] args) {
    // todo: i18n
    System.out.println("Hello world!");
  }
}
```

PrintArguments.java

```
public class PrintArguments {
  public static void main(String[] args) {
    for (int i = 0; i < args.length; ++i) {
      System.out.println("args["+i+"] = "+args[i]);
    }
  }
}
```

BottlesOfBeer.java

```
public class BottlesOfBeer {
  public static void main(String args[]) {
    int k = 99;
    while (k > 0) {
      System.out.printf(
          "%d bottle%s of beer on the wall,\n" +
          "%d bottle%s of beer,\n" +
          "take one down, pass it around,\n" +
          "%d bottle%s of beer on the wall.\n\n",
          k,
          ((k != 1) ? "s" : ""),
          k,
          ((k != 1) ? "s" : ""),
          --k,
          ((k != 1) ? "s" : "")
      );
    }
    System.out.println("No more bottles of beer on the wall!");
  }
}
```

javac

- Java Compiler
- Компилирует исходный код (`*.java`) в байткод (`*.class`)

- `javac MyClass.java YetAnotherClass.java`
- `javac -d classes MyClass.java`
- `javac -classpath library.jar -d classes MyClass.java`
- `javac -version`

Отступление: о classpath

- Все используемые классы должны быть доступны в classpath
- Всегда содержит классы стандартной библиотеки (`jre/lib/rt.jar`)
- По умолчанию содержит текущую директорию «`.`»

- Задается как список директорий и/или JAR-файлов
- Разделитель списка
  - «`:`» в Unix/Linux/Mac OS X
  - «`;`» в Windows

jar

- Java Archive Tool
- Создает и распаковывает JAR-файлы

- `jar cf library.jar -C classes_dir .`
- `jar cfm library.jar manifest.mf -C classes_dir .`
- `jar cfe library.jar MyMainClass -C classes_dir .`
- `jar tf library.jar`
- `jar xf library.jar`

Отступление: о MANIFEST.MF

- Любой JAR-файл содержит `META-INF/MANIFEST.MF`
- Пример:

```
Manifest-Version: 1.0
Created-By: 1.6.0_35 (Sun Microsystems Inc.)
```

- `Main-Class` — имя класса с методом `main`
- `Class-Path` — список необходимых JAR'ов, через пробел

java

- Java Virtual Machine
- Исполняет байткод
- Главный класс должен иметь метод
  `public static void main(String[] args)`

- `java MyClass`
- `java -classpath classes_dir;library.jar MyClass`
- `java -jar library_with_main_class.jar`
- `java -version`

javap

- Java Disassembler
- Выводит class-файлы в читабельном виде

- `javap MyClass`
- `javap -c MyClass`
- `javap -v MyClass`

Среды разработки

- Eclipse
- IntelliJ IDEA
- NetBeans

- Подсветка синтаксиса
- Автодополнение, гиперссылки
- Рефакторинг
- Интерактивный отладчик

Инструменты для сборки

- Ant
- Gradle
- Maven

- Гибкость
- Независимость от IDE
- Возможность сборки в командной строке

Что сегодня узнали

- Что такое Java и с чем её едят
- В чем особенности Java и отличия от C/C++
- Как выглядят программы на Java
- Как собрать программу на Java
- Как запустить программу на Java