Модуль #1
Введение в Java

Модуль 1

- **История создания**
- Особенности Java
- Понятие платформы Java
- Версионность
- Направления Java
- Стандартный инструментарий Java
- Процесс разработки и запуска
- Использование комментариев в JavaDoc

История создания

1991 - Внутренний проект (Green Project) компании Sun Microsystems по созданию платформ для встраиваемых систем. Вместо C++ создали новый язык – Oak. Автор Джэймс Гослинг.

История создания

1991 - Внутренний проект (Green Project) компании Sun Microsystems по созданию платформ для встраиваемых систем. Вместо C++ создали новый язык – Oak. Автор Джэймс Гослинг.

1992 - Первое демонстрационное устройство на новой платформе – PDA Star 7.

История создания

1991 - Внутренний проект (Green Project) компании Sun Microsystems по созданию платформ для встраиваемых систем. Вместо C++ создали новый язык – Oak. Автор Джэймс Гослинг.

1992 - Первое демонстрационное устройство на новой платформе – PDA Star 7.

1993 - Попытка занять область приставок для кабельного ТВ.

История создания

1991 - Внутренний проект (Green Project) компании Sun Microsystems по созданию платформ для встраиваемых систем. Вместо C++ создали новый язык – Oak. Автор Джэймс Гослинг.

1992 - Первое демонстрационное устройство на новой платформе – PDA Star 7

1993 - Попытка занять область приставок для кабельного ТВ.

1994 - Java перефокусировали на разработку апплетов. Язык переименовали в Java.

История создания

1991 - Внутренний проект (Green Project) компании Sun Microsystems по созданию платформ для встраиваемых систем. Вместо C++ создали новый язык – Oak. Автор Джэймс Гослинг.

1992 - Первое демонстрационное устройство на новой платформе – PDA Star 7

1993 - Попытка занять область приставок для кабельного ТВ.

1994 - Java перефокусировали на разработку апплетов. Язык переименовали в Java.

1996 - Java Development Kit.

История создания

1996 – Java Development Kit.

1997 – JDK 1.1

1998 – JDK 1.2, “Java 2”, разделение на ME/SE/EE

2000– J2SE 1.3

2002– J2SE 1.4

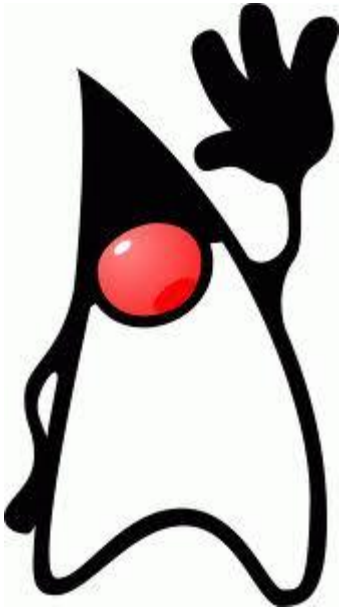
2004– J2SE 5.0, изменилась нумерация.

2006– Java SE 6, уход от “Java 2”.

2011– Java SE 7

2013– Java SE 8

История создания



Модуль 1

- История создания
- **Особенности Java**
- Понятие платформы Java
- Версионность
- Направления Java
- Стандартный инструментарий Java
- Процесс разработки и запуска
- Использование комментариев в JavaDoc

Особенности Java

- Простота и наглядность
- Переносимость
- Многопоточность
- Сборка мусора
- Безопасность

Простота и наглядность

- Изначальная объектно-ориентированность языка.
- Необходимо избежать сложность существующих языков, упростив синтаксис, но сохранив мощь.
- Необходимо дать пользователю возможность создавать гибкий, чистый код приложения любого размера.

Переносимость

- Подход C/C++

  исходный код → машинный код → процессор

  - Программа работает только на той платформе, под которую скомпилирована.

Переносимость

- Подход C/C++

  исходный код → машинный код → процессор

  - Программа работает только на той платформе, под которую скомпилирована.

- Подход Java

  исходный код → байт код ВМ → ВМ → процессор

  - Программа работает на любой платформе где есть ВМ Java
  - “Write once, run anyware!” (Windows, Linux, Solaris, NetWare, Mac OS)

Виртуальная машина и байт-код

- Как быстро работает виртуальная машина?

Виртуальная машина и байт-код

Как быстро работает виртуальная машина?

- Интерпретация байт кода медленнее выполнения аналогичного машинного кода ... 10-20 раз :-(

Виртуальная машина и байт-код

Как быстро работает виртуальная машина?

- Интерпретация байт кода медленнее выполнения аналогичного машинного кода ... 10-20 раз :-(

  … но появилась Just-In-Time компиляция … **:-)**

  - ВМ компилирует байт-код в машинный.
  - начиная с версии JDK 1.1

Виртуальная машина и байт-код

Как быстро работает виртуальная машина?

- Интерпретация байт кода медленнее выполнения аналогичного машинного кода ... 10-20 раз :-(

… но появилась Just-In-Time компиляция … **:-)**

- ВМ компилирует байт-код в машинный.
- начиная с версии JDK 1.1

… и HotSpot **:-)**

- Адаптивный оптимизирующий JIT-компилятор
- начиная с версии JDK 1.3

Сборка мусора

- Подход C/C++

выделил память → поработал → освободил память

  - управление памятью ложится на плечи программиста
  - управление памятью ведет к большому числу ошибок

Сборка мусора

- Подход C/C++

выделил память → поработал → освободил память

  - управление памятью ложится на плечи программиста
  - управление памятью ведет к большому числу ошибок

- Подход Java

выделил память → поработал → забыл

  - ВМ сама занимается освобождение памяти
  - нет ошибок, связанных с памятью

Многопоточность и распределенное программирование

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

Безопасность

- Верификация байт-кода.

(Некорректный байткод будет отвергнут перед исполнением)

- Автоматическое управление памятью

(Нет арифметики указателей, невозможно испортить память)

Безопасность

- Верификация байт-кода.

  (Некорректный байткод будет отвергнут перед исполнением)

- Автоматическое управление памятью

  (Нет арифметики указателей, невозможно испортить память)

- Встроенный механизм управления правами

  (Можно запустить код «с ограничениями», например без доступа к файлам, без сети или создания потоков. И это невозможно обойти)

- Java — сильно типизированный язык

  (Позволяет отловить большинство ошибок связаных с типами на этапе компиляции)

Модуль 1

- История создания
- Особенности Java
- **Понятие платформы Java**
- Версионность
- Направления Java
- Стандартный инструментарий Java
- Процесс разработки и запуска
- Использование комментариев в JavaDoc

Платформа Java

- Java – это не только язык программирования, но и платформа, которая включает в себя:
  - Язык Java
  - Виртуальная машина Java
  - Системная библиотека (JRE)
  - Инструментарий разработки (JDK)

Библиотеки

Java – это еще и огромное количество библиотек:

- **Общего назначения**
  - **Apache Commons, Google Guava, Joda Time ...**
- **Логирование**
  - **SL4J, Log4J, jLo, LogBack, Logging toolkit ...**
- **Тестирование**
  - **Junit, TestNG, EasyMock, Mockito ..**
- **....**

Системы сборки

- **Apache ANT**
  - Утилита для автоматизации процесса сборки программного продукта. Является платформонезависимым аналогом утилиты make
- **Apache Ivy**
  - Позволяет разработчику управлять зависимостями java библиотек при компиляции и развертывании java приложений т.е. автоматизировать сборку приложения в области закачки нужных java библиотек
- **Apache Maven**
  - Мощный инструмент по «управлению» проектом.

Среды разработки

- **Eclipse IDE**
  - Свободная интегрированная среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation.
- **NetBeans**
  - Проект NetBeans IDE поддерживается и спонсируется компанией Oracle, однако разработка NetBeans ведется независимым сообществом разработчиков-энтузиастов.
- **Intelij IDEA**
  - Коммерческая интегрированная среда разработки программного обеспечения на многих языках программирования.

Среды разработки

- **Eclipse IDE**
  - Свободная интегрированная среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation.
- **NetBeans**
  - Проект NetBeans IDE поддерживается и спонсируется компанией Oracle, однако разработка NetBeans ведется независимым сообществом разработчиков-энтузиастов.
- **Intelij IDEA**
  - Коммерческая интегрированная среда разработки программного обеспечения на многих языках программирования.

 Для чего нужны среды разработки?

Виртуальная машина

- Существуют реализации JVM, написанные для всех современных ОС.
- JVM, соответствующая спецификации SUN может запускать любой class файл.
- JVM управляет загрузкой и работой программы, продоставляет стек и другие области памяти, реализует сборщик мусора и другие функции.
- JVM выступает абстракцией между языком Java и платформой, на которую установлена Java.

Реализация Java

- **Oracle Java** - официальная реализация

  `http://java.oracle.com`

Реализация Java

- **Oracle Java** - официальная реализация

  `http://java.oracle.com`

- **OpenJDK**

  `http://openjdk.java.net`

- **Iced Tea**

  `http://icedtea.classpath.org`

- **JRockit**

- … и еще несколько десятков.

Языки запускаемые на JVM

- Groovy
- JRuby
- Jython
- Clojure
- Scala
- Kotlin
- Rhino
- Ceylon
- Phantom
- И

Тонкая настройка JVM

- Можно настроить JVM для повышения производительности сервера.

- Переменная JAVA_OPTS для параметров виртуальной машины

  - Начальный размер кучи

    -XmsIm, I – в мегабайтах

  - Максимальный размер кучи

    -XmxIm, I – в мегабайтах

Тонкая настройка JVM

- Параметры настройки разделены на 4 категории:
  - Поведенческие параметры

    -XX:-UseSerialGC -XX:+UseThreadPriorities
  - Параметры сборки мусора

    -XX:MaxGCPauseMillis=n
  - Параметры производительности

    -XX:MaxHeapFreeRatio=70 -XX:MaxHeapFreeRatio=70
  - Параметры отладки

    -XX:-HeapDumpOnOutOfMemoryError -XX:-CITime

Внимание! Более подробно на *http://www.oracle.com/technetwork/java/javase/tech/vmoptions-jsp-140102.html*

Модуль 1

- История создания
- Особенности Java
- Понятие платформы Java
- **Версионность**
- Направления Java
- Стандартный инструментарий Java
- Процесс разработки и запуска
- Использование комментариев в JavaDoc

Версионность

- Изменениями управляет Java Community Process (JCP)
- Изменения в версиях затрагивали как сам язык, так и саму платформу.
- Количество классов в системной библиотеке выросло от нескольких сотен до нескольких тысяч.
- Всегда сохранялся прицип “backward compatibility”.

Версионность

- JDK 1.0 (January 23, 1996)
- JDK 1.1 (February 19, 1997)
- J2SE 1.2 (December 8, 1998)
- J2SE 1.3 (May 8, 2000)
- J2SE 1.4 (February 6, 2002)
- J2SE 5.0 (September 30, 2004)
- Java SE 6 (December 11, 2006)
- Java SE 7.0 (July 7, 2011)
- Java SE 8.0 (в разработке)

Модуль 1

- История создания
- Особенности Java
- Понятие платформы Java
- Версионность
- **Направления Java**
- Стандартный инструментарий Java
- Процесс разработки и запуска
- Использование комментариев в JavaDoc

Направления Java

- **Java SE (Standard Edition)**
  - Стандартная версия. Отлично подходит для десктопных приложений.

Направления Java

- **Java SE (Standard Edition)**
  - Стандартная версия. Отлично подходит для десктопных приложений.
- **Java ME (Micro Edition)**
  - Урезанная версия Java. Предназначена для устройств с ограниченными ресурсами.

Направления Java

- **Java SE (Standard Edition)**
  - Стандартная версия. Отлично подходит для десктопных приложений
- **Java ME (Micro Edition)**
  - Урезанная версия Java. Предназначена для устройств с ограниченными ресурсами.
- **Java EE (Enterprise Edition)**
  - Для корпоративных приложений масштаба предприятия.

Направления Java

- **Java FX**
  - Платформа, предназначеная для создания приложений с мультимедийным контентом и графическим интерфейсом пользователя (RIA).

Направления Java

- **Java FX**
  - Платформа, предназначеная для создания приложений с мультимедийным контентом и графическим интерфейсом пользователя (RIA).
- **Java TV (Television)**
  - Основанная на JavaME версия, предназначенная для простой, быстрой и безопасной разработки Java-приложений, работающих на телевизионных приемниках.

Направления Java

- **Java FX**
  - Платформа, предназначеная для создания приложений с мультимедийным контентом и графическим интерфейсом пользователя (RIA).
- **Java TV (Television)**
  - Основанная на JavaME версия, предназначенная для простой, быстрой и безопасной разработки Java-приложений, работающих на телевизионных приемниках.
- **Java Card**
  - Технология предназначенная для создания приложений очень ограниченных в ресурсах, которые запускаются в смарт-картах.

Режимы работы Java

- Client mode
  - Быстрый вход на максимальную производительность.

Режимы работы Java

Client mode

- Быстрый вход на максимальную производительность.

Server mode

- Долго накапливает данные о приложении, производит оптимизацию самых HotSpot точек (20%). В результате сбалансированный и оптимизированный код.
- Автоматичесий переход при:
  - 2 или более CPU ядер
  - 2 или более GB ОЗУ

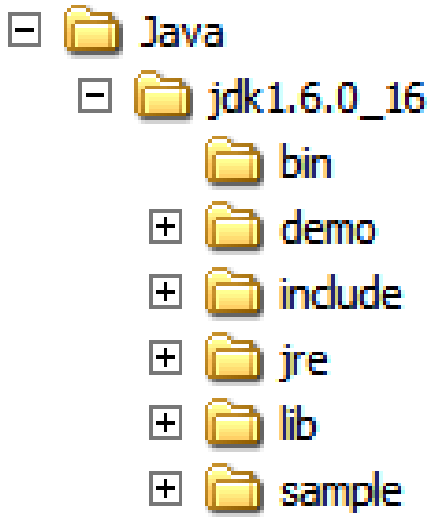
```
java -client
java -server
```

Модуль 1

- История создания
- Особенности Java
- Понятие платформы Java
- Версионность
- Направления Java
- **Стандартный инструментарий Java**
- Процесс разработки и запуска
- Использование комментариев в JavaDoc

Обзор JDK

- JDK можно скачать с сайта www.oracle.com/technetwork/java.
- JDK распространяется как инсталятор.
- По умолчанию ставится в C:\Program Files\Java.
- Можно установить только JRE.



Обзор JDK

- javac – компилятор языка Java, соответствующей спецификации JLS и возвращающий байт-код спецификации JVM.
- java – загрузчик Java приложений, реализация JVM.
- jar – архиватор .class файлов.
- javadoc — генератор документации.
- jdb – отладчик

Обзор JDK

- javap – дизассемблер class файлов.
- jvisualvl – профайлер.
- jarsigner – инструмент для подписи jar-файлов.
- jconsole – графический инструмент мониторинга для контроля JVM.

Внимание! Под большинство утилит есть API для программного вызова.
Например, javac → `ToolProvider.getSystemJavaCompiler()`

javac

- **Java Compiler**

- Компилирует исходый код (*.java) в байткод (*.class)

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

Модуль 1

- История создания
- Особенности Java
- Понятие платформы Java
- Версионность
- Направления Java
- Стандартный инструментарий Java
- **Процесс разработки и запуска**
- Использование комментариев в JavaDoc

Процесс разработки и запуска

- Процесс создания простейшего приложения включает 3 этапа:
  - Редактирование файла исходного кода в текстовом редакторе или IDE.
  - Компиляцию файла с исходным кодом.
  - Запуски скомпилированного класса.

Процесс разработки и запуска

- Каждый файл исходного кода на java должен иметь расширение «.java». Имя файла должно совпадать с именем **public** класса.

```
public class HelloWorld {

    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```

- Java coding convensions

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

Процесс разработки и запуска

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

Процесс разработки и запуска

- Для компиляции файла исходного кода необходимо выполнить утилиту javac из набора JDK:

```
javac HelloWorld.java
```

- В случае успешной компиляции генерируется соответствующий .class (например, HelloWorld.class), содержащий скомпилированный байт-код исходного файла.

Процесс разработки и запуска

- Скомпилированный файл можно запустить в JVM используя команду:

```
java HelloWorld
```

Внимание! Аргумент утилиты java – имя класса, а не имя файла с байт-кодом.

Модуль 1

- История создания
- Особенности Java
- Понятие платформы Java
- Версионность
- Направления Java
- Стандартный инструментарий Java
- Процесс разработки и запуска
- **Использование комментариев в JavaDoc**

JavaDoc

- JavaDoc – набор правил описания комментариев, а также специальных директив в файле исходного кода, позволяющий с помощью утилиты javadoc, входящей в JDK, сгенерировать HTML документацию, описывающую:
  - Пакет
  - Описание класса, его поля и методы.

JavaDoc

Тег	Описание	Применим к
@author	Автор	Класс, интерфейс
@version	Версия, не больше одного	Класс, интерфейс
@since	С какой версии доступно	Все
@see	Ссылка на другое место в документации	Все
@param	Входной параметр метода	Метод
@return	Описание возвращаемого значения	Возвращаемое значение
@throws	Описание выбрасываемого исключения	Метод
@deprecated	Описание устаревших блоков кода	Все

JavaDoc

```
/**
 * This is simplest for of Java class. It prints hello world message.
 * @author Peter Pen
 */
public class HelloWorld {

    /**
     * Definition for hello world message.
     */
    public static final String HELLO_MESSAGE = "Hello, World";

    /**
     * Main methods which is tun by JVM and prints the message.
     *
     * @param args Command line arguments
     */
    public static void main(String[] args) {
        System.out.println(HELLO_MESSAGE);
    }
}
```

- javadoc HelloWorld.java

- Сгенерированый файл – index.html

JavaDoc

- Документация слишком часто устаревает
- Не забыть описать краевые случаи
  - Как поведет себя метод, если на входе будет **`null`**
  - Модификация параметров метода
  - . . .
- Описание параметров.

Упражнение 1

Установка и настройка рабочей среды

Упражнение 2

Анализ и запуск первого Java-приложения

Модуль 1

- История создания
- Особенности Java
- Понятие платформы Java
- Версионность
- Направления Java
- Стандартный инструментарий Java
- Процесс разработки и запуска
- Отладка приложений
- Использование комментариев в JavaDoc