

Теория и технология программирования

Программирование на языке Java

Лекция 1. Введение в технологию Java

Глухих Михаил Игоревич, к.т.н., доц.

[mailto: glukhikh@mail.ru](mailto:glukhikh@mail.ru)

Структура курса

- ❑ Технология Java
- ❑ Типы, операции, конструкции, ввод-вывод
- ❑ Структура проекта, классы, интерфейсы, наследование
- ❑ Коллекции и утилиты
- ❑ Тестирование на Java: JUnit
- ❑ GUI на Java: AWT, Swing
- ❑ Апплеты
- ❑ Архитектура MVC
- ❑ Многопоточные приложения

Аттестация

- Упражнения: курсовой проект с оценкой
- Лекции: зачет

Литература

- ❑ Ильдар Хабибуллин. Java 7. СПб.: БХВ-Петербург, 2012
- ❑ Арнольд К., Гослинг Дж., Холмс Д. Язык программирования Java, 3-е издание. М.: Издательский дом «Вильямс», 2001.
- ❑ James Gosling, Bill Joy, Guy Steele, Gilad Bracha. The Java Language Specification, Second Edition.
- ❑ Joshua Bloch. Effective Java: Programming Language Guide, second edition. ISBN 978-0-321-35668-0, 2008.

Технология Java

- Разработчик – Sun Microsystems (1995)
 - в 2009 году куплена корпорацией Oracle
- Состав
 - Язык Java
 - Java Development Kit
 - Java Runtime Environment
 - Java Virtual Machine
 - Среды разработки

Основные принципы

- Простота
- Объектная ориентированность
- Строгая типизация
- Безопасность
- Архитектурная независимость
- Высокая производительность
- Интерпретируемость
- Многопоточность

Язык Java

- ❑ Полностью объектно-ориентированный язык
- ❑ Многие заимствованы из языка Си++ (в частности, почти весь набор базовых конструкций и операций)
- ❑ Из Си++ убраны сомнительные элементы и привнесены несколько новых
- ❑ Программирование в значительной степени упрощено (меньше вероятность ошибок)

Ключевые отличия от C++ (общая структура)

- ❑ Нет глобальных переменных и функций (есть классы и их члены)
- ❑ Введена иерархия пакетов для разграничения действия имён
- ❑ Убрано разделение на заголовочные файлы и файлы с исходным кодом

Ключевые отличия от C++ (типы)

- ❑ Устранен тип «указатель» (оставлены ссылочные типы)
- ❑ Введен самостоятельный тип «строка» (более не является массивом символов)
- ❑ Разграничены целый и логический типы (их более нельзя перемешивать)
- ❑ Запрещены некоторые преобразования типов

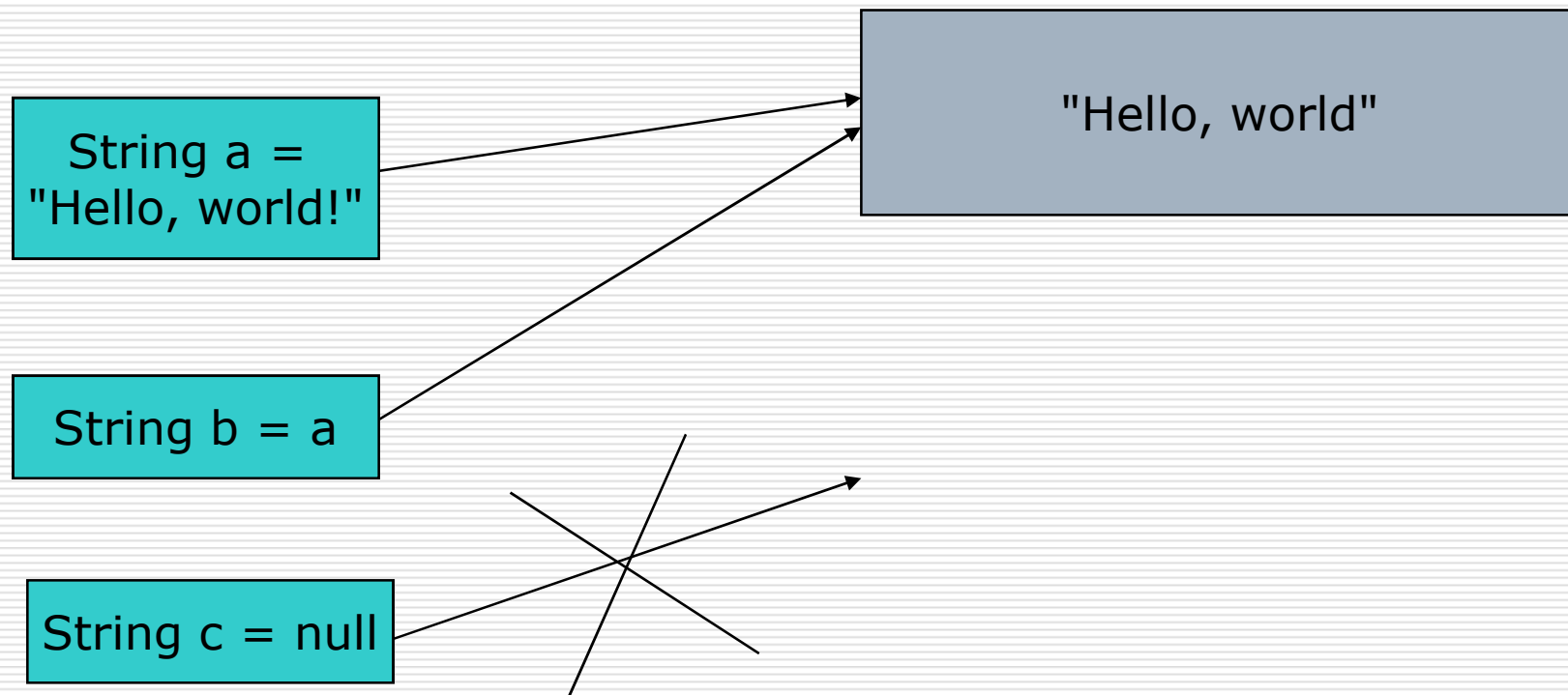
Ключевые отличия от C++ (безопасность)

- ❑ Безопасный язык – гарантирует целостность своих собственных элементов
- ❑ При интерпретации производится контроль:
 - границ массивов
 - доступа по нулевой ссылке
 - переполнения стека
 - преобразования типов
 - ...
- ❑ Ошибки приводят к исключениям

Ключевые отличия от C++ (работа с памятью)

- ❑ Переменные примитивных типов (целый, символьный, вещественный, логический) в java хранятся в стеке (или в статической памяти)
- ❑ Переменные других типов (массив, строка, класс, перечисление), в java хранятся в динамической памяти (при этом в стеке запоминается ссылка)
- ❑ Нет деструкторов и операторов освобождения памяти (вместо них работает сборщик мусора)

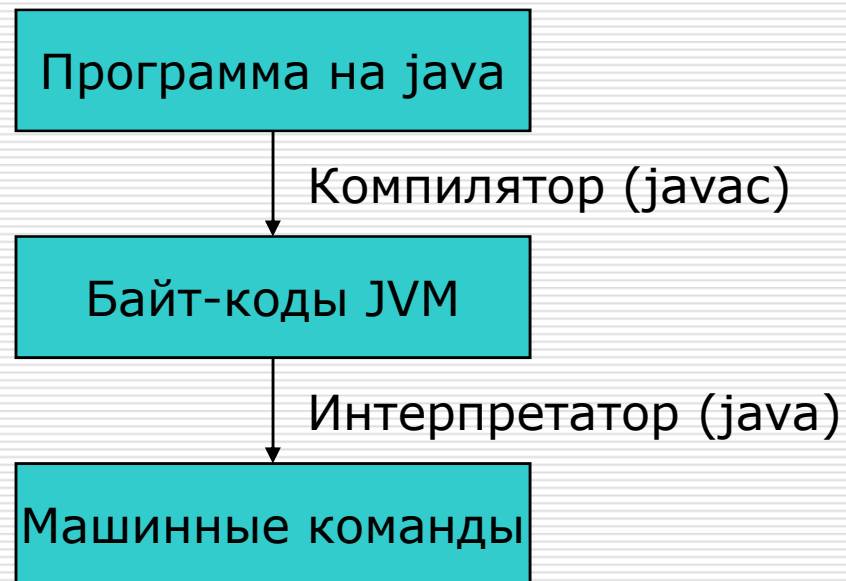
Ключевые отличия от C++ (ссылочные типы)



Ключевые отличия от C++ (иерархия классов)

- ❑ Нет множественного наследования классов
- ❑ Нет закрытого и защищенного наследования
- ❑ Введено понятие интерфейса
 - интерфейс = класс без полей, только с чисто виртуальными методами (определяет, ЧТО можно делать, но не КАК)
 - есть множественное наследование интерфейсов

Компиляция и исполнение программ на языке Java



Java VS C/C++ (достоинства)

□ C/C++

- Высокая производительность программ
- Меньшие затраты памяти
- Возможность системного программирования

□ Java

- Высокая скорость и удобство разработки
- Кроссплатформенность

Java VS C/C++ (применение)

□ C/C++

- Системное ПО
- Алгоритмы высокой трудоемкости (математика, сложный перебор и пр.)

□ Java

- GUI-приложения
- Web-приложения
- приложения для мобильных телефонов
- приложения для встроенных систем

Java Development Kit (JDK)

- JDK предназначен для разработки на языке Java и включает:
 - Компилятор (javac)
 - Отладчик (jdb)
 - Архиватор (jar)
 - Сборщик документации (javadoc)
 - Исходные тексты библиотеки классов Java API
 - + Java Runtime Environment

Java Runtime Environment (JRE)

- JRE предназначен для исполнения программ на языке Java, включает:
 - Java Virtual Machine (JVM)
 - интерпретатор (java)
 - включает возможность Just-In-Time компиляции
 - Байт-коды библиотечных классов

Java Virtual Machine (JVM)

- ❑ Совокупность команд вместе с системой их исполнения
- ❑ В отличие от машинных команд, не зависят от архитектуры

Варианты реализации

- ❑ **Java 2 Standard Edition** – для обычных машин (рабочих станций)
- ❑ Java 2 Enterprise Edition – серверный вариант
- ❑ Java 2 Micro Edition – для мобильных устройств

Версии Java SE

- ❑ Java SE 7 – 2011 год, включает ряд небольших изменений в языке (с обратной совместимостью),
JDK 7 update X или JDK 7uX
- ❑ Java SE 6 – 2006 год,
JDK 6 update X или JDK 6uX

Известные среды разработки

Бесплатные

- Eclipse (Eclipse Foundation, IBM)

- **NetBeans (Sun)**

Коммерческие

- IntelliJ IDEA (JetBrains)

- Borland JBuilder (Borland)

- Oracle9i JDeveloper (Oracle)

- Novell exteNd (Novell)

Основные поддерживаемые ОС

- Windows
- Unix
- Solaris
- Mac OS

Где достать

- <http://java.oracle.com> – официальный сайт технологии Java (JRE, JDK)
- <http://netbeans.org> – официальный сайт NetBeans
- <http://www.jetbrains.com/idea> – страничка IntelliJ IDEA

Что требуется для работы

- ❑ Java Standard Edition (SE) Development Kit (последняя версия 7u7)
- ❑ NetBeans (последняя версия 7.2)
 - или бесплатная версия IntelliJ IDEA
- ❑ И то, и другое – свободно распространяемое

Желательная конфигурация ПК

- ❑ ОЗУ ≥ 512 Мб
(рекомендуется ≥ 2 Гб)
- ❑ CPU \geq Pentium IV / Athlon
(рекомендуется многоядерный вариант)

Демонстрация работы среды

□ См.

Простейшая программа на Java

```
// Файл Hello.java
```

```
// См. пример
```

```
package test;
```

```
public class Hello {
```

```
    public static void main(String[] args) {
```

```
        System.out.println("Здравствуй, мир!");
```

```
    }
```

```
}
```

Имена в языке Java

- ❑ Формируются по правилам C/C++
- ❑ Общепринятые соглашения Java-разработчиков (Java Code Style):
 - имя класса (интерфейса) начинается с прописной буквы (**Collection, LinkedList**)
 - имя переменной начинается со строчной буквы и является существительным (**index**)
 - имя функции начинается со строчной буквы и является глаголом (**addAll**)
 - имя константы состоит из прописных букв (**INSTANCE**)

Комментарии

```
/* Комментарий
```

```
в стиле
```

```
языка C */
```

```
// Комментарий
```

```
// в стиле языка C++
```

```
/**
```

```
* Документирующий комментарий
```

```
* в стиле языка Java
```

```
*/
```

Примитивные типы

- Целые
 - byte (1 байт, -128...127)
 - short (2 байта, -32768...32767)
 - int (4 байта, $-2^{31} \dots 2^{31}-1$)
 - long (8 байт, $-2^{63} \dots 2^{63}-1$)
- Вещественные
 - float (4 байта)
 - double (8 байт)
- Символьный
 - char (2 байта, Unicode, 0...65535)
- Логический
 - boolean (true или false)

Константы

□ Целые

- 57, +323, -48 (десятичная форма, 4 байта)
- 024, -0634, 0777 (восьмеричная форма)
- 0xabcd, -0x19f (шестнадцатеричная форма)
- 0b010001001 (двоичная форма, **только в JDK 7**)
- 43_934 (форма с _, **только в JDK 7**)
- 1234567890123L, 0abcdef1234L (8-байтные)

□ Вещественные

- 37.29, -19.41 (обычная форма, 8 байт)
- 3e+12, -1.1e-7 (экспоненциальная форма)
- 3.6F, -1.0e-1F (4-байтные)

Константы

□ Символьные

- 'a', '?', ' ', '\n', '\t', '\\' (обычный вариант)
- '\40', '\62' – восьмеричный код
- '\u0053' – юникод

□ Строковые

- "Hello, world\n"
- "Сложение " + "строк"

Операции

- Арифметические: + - * / % ++ --
- Логические: & && | || ^ !
- Сравнения: > < >= <= == !=
- Побитовые: ~ & | ^
- Сдвиговые: << >> >>>
- Присваивания: = += -= *= /= %= &= |=
^= <<= >>= >>>=
- Условная: a>b?a:b
- Приведения типа: int a = (int)2.5;

ИТОГИ

- Рассмотрены особенности технологии Java
- Рассмотрена простейшая программа
- Рассмотрены примитивные типы
- Рассмотрены константы и операции