

JAVA

ОСНОВЫ

ORACLE®
Certified Professional
Java SE 6 Programmer

ORACLE®
Certified Expert
Java EE 6 Web
Component Developer

Бардин Дмитрий Сергеевич
dbardin@croc.ru

Road map

- Введение
- Среда разработки
- Типы данных
- Основы синтаксиса

Базовые ссылки

Virtual Machine specification

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

Language Specification

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

API

<http://docs.oracle.com/javase/7/docs/api/index.html>

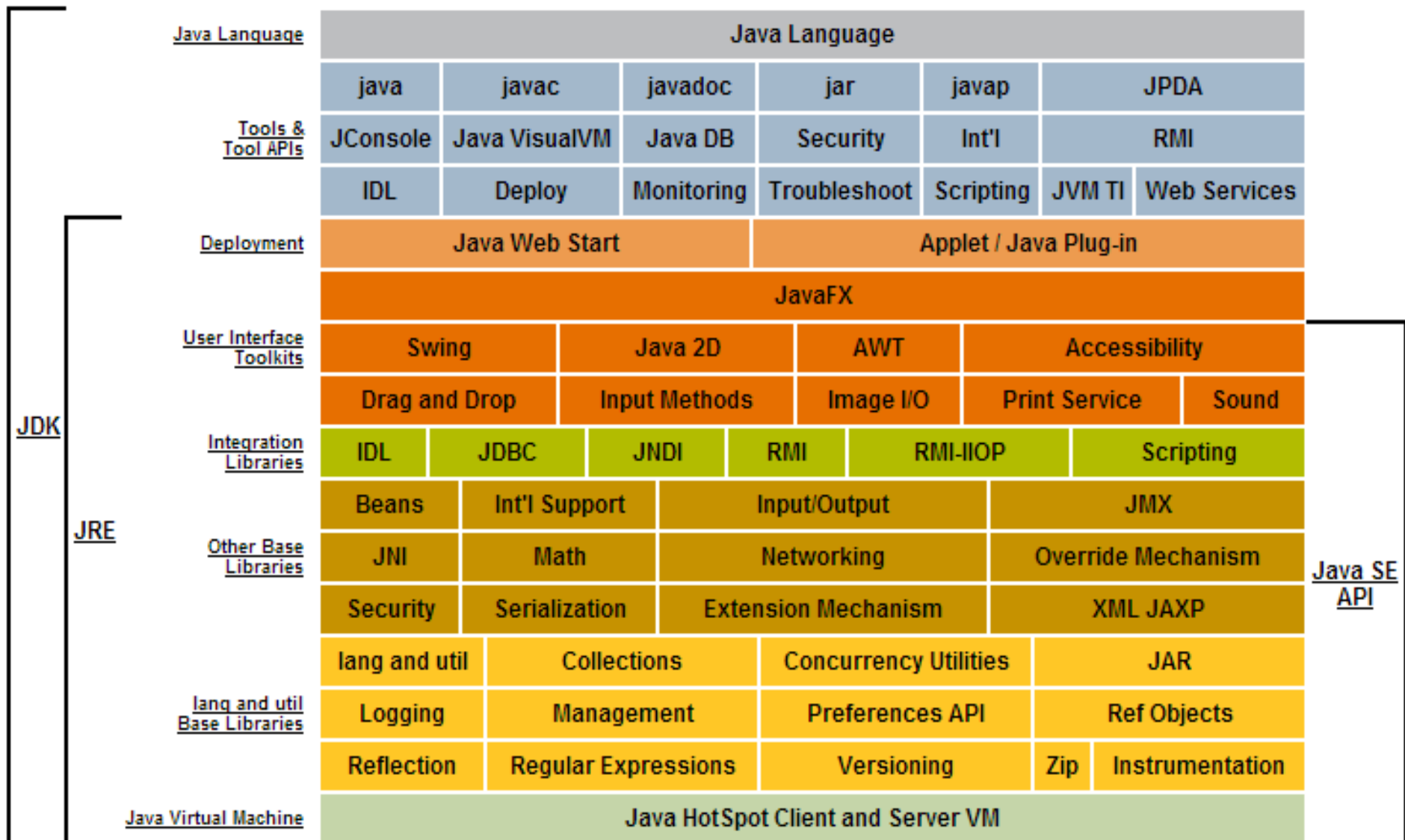
Code Conventions

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

Варианты поставок

- Java Standard Edition (Java SE)
- Java Enterprise Edition (Java EE)
- Java Micro Edition (ME)
- Java Card

Редакция Java SE



Java среда (environment)

- JRE

минимальная реализация,
необходимая для
исполнения приложений, без
компилятора и других средств
разработки.

- JDK

бесплатно распространяемый комплект
разработчика приложений включающий
в себя компилятор (`javac`), стандартные
библиотеки классов, примеры,
документацию, различные утилиты. В
состав не входит интегрированная
среда разработки (IDE)

JVM

исполнение байт-кода,
предварительно
созданного из исходного
кода компилятором
(`javac`)

Java Virtual Machine

- Byte code
- JIT-компиляция (по мере необходимости, во время выполнения)
- Garbage collector

Ссылки:

[как работает](#)

[GC](#)

Среда разработки

- Eclipse 4.2.1 <http://www.eclipse.org/>
- IntelliJ IDEA 12 <http://www.jetbrains.com/idea/>
- NetBeans 7.3 <http://netbeans.org/>

Основные понятия

- JAVA_HOME
- Пакет (package)
- .class , .jar , .war, .ear
- Classpath
- Класс, метод, поле, свойства (методы доступа)

Компиляция (.java -> .class)

javac -classpath <библиотеки .jar> <имя файла .java>

Ex.: javac Test.java

Выполнение

java -classpath <путь к классам и библиотекам .jar> <полное имя класса с функцией main() >

Ex.: java -classpath . Test

Основные понятия

- Добавление пакетов в проект

```
import java.util.*;
```

```
import java.io.*;
```

- javadoc

Встроенные типы (primitive types)

Тип	Размер	Минимальное значение	Максимальное значение	Тип-обертка
boolean	1-bit	—	—	Boolean
char	16-bit	Unicode 0	Unicode $2^{16}-1$	Character
byte	8-bit	-128	+127	Byte
short	16-bit	-2^{15}	$+2^{15}-1$	Short
int	32-bit	-2^{31}	$+2^{31}-1$	Integer
long	64-bit	-2^{63}	$+2^{63}-1$	Long
float	32-bit	IEEE754	IEEE754	Float
double	64-bit	IEEE754	IEEE754	Double

Встроенные типы

.Типы «по значению»: `boolean`, `char`, `byte`, `short`, `int`, `long`, `float`, `double` (без «`new`»)

- Хранятся на стеке

- Для хранения в куче производится «упаковка» (`boxing`)

.Типы «по ссылке»: все остальные

Значения по умолчанию

Тип	Значение по умолчанию
boolean	false
char	'\u0000' (null)
byte	(byte)0
short	(short)0
int	0
long	0L
float	0.0f
double	0.0d

Встроенные типы: boolean

- Представляется как int. 0 — false, 1 — true.
- Массив boolean — это массив byte.

Встроенные типы: char

- .16-битное беззнаковое целое

- .По семантике содержит символы в кодировке UTF-16

Встроенные типы: целые

.byte — знаковое, 8-битное

.short — знаковое, 16-битное

.int — знаковое, 32-битное

.long — знаковое, 64-битное

– На стеке JVM занимает два слота

Встроенные типы: float и double

.Специальные значения:

- NaN
- Положительный/отрицательный ноль
- Положительная/отрицательная бесконечность

Встроенные типы: Reference

- .Содержит ссылку на объект в куче (не сам объект)
- .Значение по умолчанию — null

Ссылочные типы (Reference)

List items;

//создается только ссылка, но не объект.
Ссылка указывает на null.

List items=new ArrayList<String>();

//создается объект (выделяется
память в куче и происходит
инициализация).

Items.Add(“элемент”);

//доступ к объекту по
ссылке, добавление
элемента

Ссылочные типы: Массивы

- Размер нельзя изменить
- Хранят только один тип
- Только int или char как индекс
- ArrayIndexOutOfBoundsException
- Гарантированная инициализация

Ex.: `String[] newMessages=new String[10];`

`int [][] arr = { {1,2,3}, {4,5,6} };`

Hello World!

Hello.java

```
package ru.mai.tmp.test;
/**
 * @author dbardin
 * Тестовый класс HelloWorld
 */
public class HelloWorld {
    /**
     * Точка входа. Печатает в консоль "Hello World!"
     * @param args параметры командной строки
     */
    public static void main ( String[] args ) {
        System.out.println( "Hello world!" );
    }
}
```

Вопросы?

Материалы доступны на

https://github.com/dbardin/tmp_course