

Павел Степанов
Кафедра компьютерной математики и
программирования СПб ГУАП

ЯЗЫК
ПРОГРАММИРОВАНИЯ
JAVA

Тема 1. Введение в Java

- Обзор
- Основные концепции платформы
- Состав платформы
- Hello, World!

1.1 Что такое Java

- Java – прослойка между приложением и операционной системой (middleware), состоит из виртуальной машины, исполняющей код, и библиотек
- Java поддерживает большое количество платформ без перекомпиляции (Windows, Linux, Solaris, MacOS)
- Java может использоваться как в серверной части приложений, так и в клиентской
 - Java удерживает порядка 25% всего рынка ПО, используется более чем половиной компаний - крупнейших разработчиков ПО и большим количеством крупных банков. Освоив Java, вы всегда найдете высокооплачиваемую работу.
 - Большое количество компиляторов и интерпретаторов других языков построено поверх виртуальной машины Java, в том числе с использованием JSR292 (“динамический вызов”) – Scala, Closure, Ruby, Python, Ada, JS, Croovy
- Текущая версия Java – Java 7. В работе находится Java 8, ориентировочная дата выхода – март 2014.
 - JCP
 - Open JDK community

1.2 Где взять Java

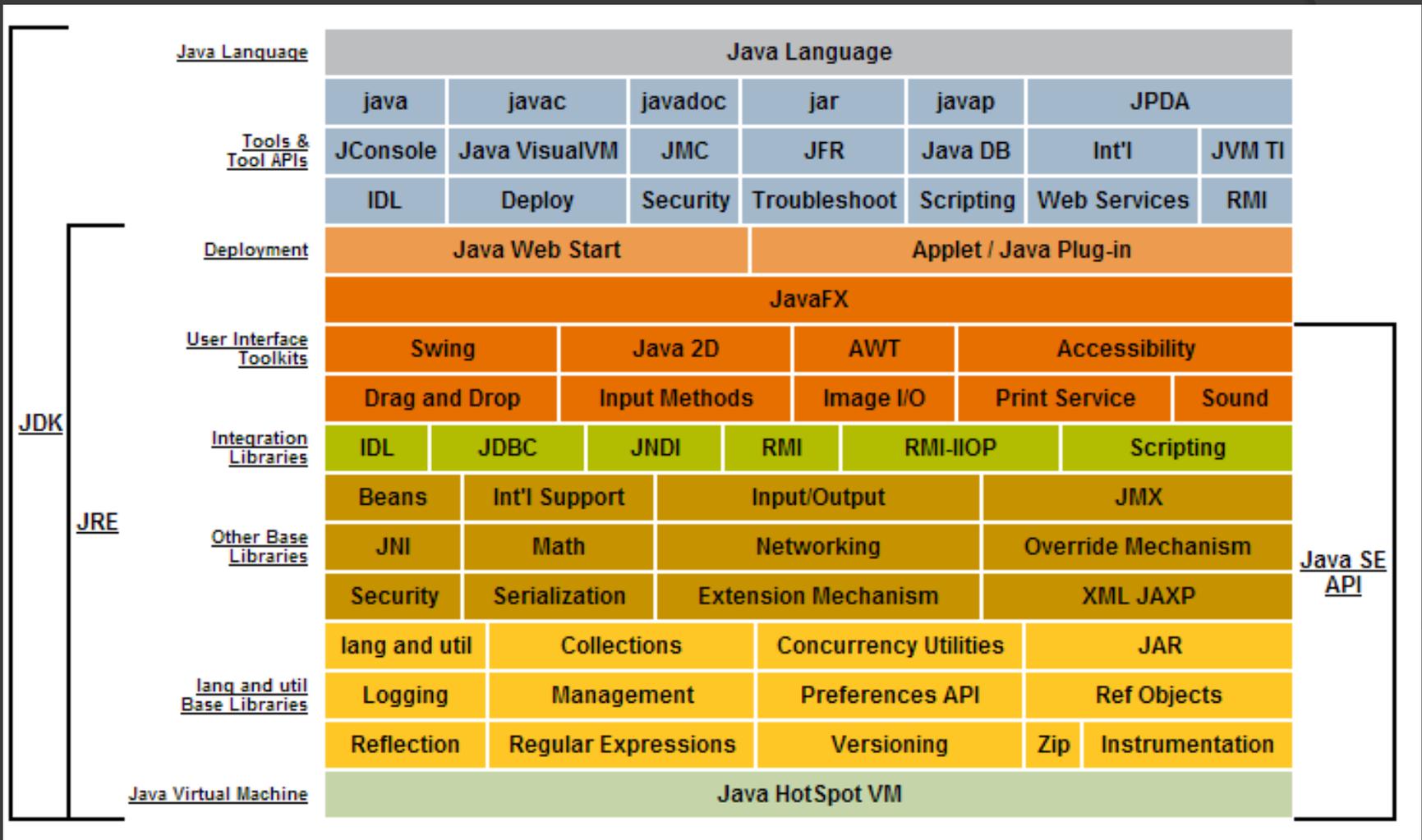
- Дистрибутивы
 - Java.com
 - OTN (java.oracle.com)
- IDE
 - Netbeans
 - Eclipse
 - Idea
- Учебники
 - Брюс Эккель “Философия Java”
 - Джошуа Блох “Java. Эффективное программирование”
 - Кен Арнольд, Джеймс Гослинг “Язык программирования Java”
 - Java Tutorials (англ.)

<http://www.oracle.com/technetwork/java/javase/documentation/index.html>

1.3 Базовые концепции

- Виртуальная машина
 - Java Compiler генерирует промежуточный байт-код, который может исполняться виртуальной машиной
- JIT – Just in time compiler. Часть виртуальной машины, определяющая “горячие” участки кода и транслирующие их в машинный код платформы на лету.
 - То, что в режиме компиляции Java всегда работает быстрее, - популярное заблуждение. Никогда не используйте `-Xcomp` для форсирования JIT – все будет работать со скоростью улитки
 - Популярно мнение, что Java приложения всегда тормозит. Для того, чтобы они работали быстро, необходимо уметь их правильно готовить.
- Автоматическое управление памятью
 - Garbage Collector автоматически освобождает память, занятую неиспользуемыми объектами. Нет деструкторов, нет прямого доступа к памяти(*), нет арифметики указателей.

1.4 Состав платформы



1.5 JDK и JRE

- Java Runtime Environment – набор средств, необходимых для исполнения программ на Java
 - Java VM
 - Основные библиотеки
 - Интеграционные библиотеки
 - Графический интерфейс пользователя
 - Deployment (средства развертывания)
- Java Development Kit – набор средств, необходимых для разработки программ на Java, включает в себя JRE
 - Инструменты (компилятор и пр.)

1.6 Виртуальная машина

- Java VM – исполняет байт-код (обычно содержится в файлах с расширением .class)
 - Oracle выпускает две виртуальные машины – HotSpot и JRockit
 - А вообще их десятки
- Виртуальная машина HotSpot поддерживает несколько сотен параметров, предназначенных для повышения производительности, оптимизации использования ресурсов, отладки, профилирования и пр.
- Just In Time Compiler (компилятор “на лету”)
- Garbage Collector (сборщик мусора), в состав входит несколько различных сборщиков, таких как Parallel, Serial, ConcMarkSweep и G1.
- Средства автоматического определения аппаратного обеспечения и использования его возможностей

1.7 Базовые библиотеки

- **Lang and util** – базовые средства языка. Библиотека lang включается в приложение всегда.
- **Collections** – средства работы с коллекциями (множествами, списками, отображениями)
- **Concurrency utilities** – средства поддержки многопоточности
- **JAR, ZIP** – средства архивации
- **Logging** – средства ведения журнала событий
- **Management** – мониторинг и управление виртуальной машиной
- **Preferences API** – управление настройками приложения
- **Ref Objects** – нежесткие ссылки на объекты
- **Reflection** – средства работы с классами, методами и атрибутами по их именам во время исполнения кода
- **Regular Expressions** – регулярные выражения
- **Versioning**
- **Instrumentation** – средства модификации исполняемого кода на лету (“инструментации”)

1.8 Другие базовые библиотеки

- **Beans** – поддержка java beans
- **Int'l support** – поддержка локализации для разных стран и языков
- **Input/Output** – библиотеки ввода-вывода
- **JMX** – стандартный интерфейс мониторинга и управления ресурсами
- **JNI** – поддержка внешнего кода, специфичного для платформы, например, на языке C.
- **Math** – поддержка математических вычислений, в том числе с неограниченной точностью
- **Networking** – сетевое взаимодействие
- **Override Mechanism** – поддержка полиморфизма
- **Security** – протоколы безопасности и шифрования
- **Serialization** – сериализация и десериализация
- **Extension Mechanism**
- **XML JAXP** – поддержка XML (на основе JAXP - Java API for XML Processing)

1.9 Интеграционные библиотеки

- **IDL** – язык описания удаленных интерфейсов (CORBA и т.д.)
- **JDBC** – Интерфейс взаимодействия с базами данных
- **JNDI** – Naming and Directory Interface (LDAP, CORBA, RMI registry, DNS)
- **RMI** – удаленный вызов процедур
- **RMI-IIOP** – поддержка RMI через протокол IIOP (CORBA)
- **Scripting** – Поддержка JavaScript (JDK 6 и ниже – RHINO, JDK 7 и выше – NASHORN)

1.10 Графический интерфейс

- **Java 2D** – работа с двумерной графикой и печатью
- **AWT** – стандартная оконная библиотека
- **SWING** – оконная библиотека с расширенными возможностями
- **Java FX** – новый графический стек, сделанный на замену SWING
- **Drag and Drop** – поддержка перетаскивания графических элементов мышью
- **Printing** – Поддержка печати на принтере
- **Accessibility** – Поддержка людей с ограниченными возможностями (экранный лупа, чтение элементов экрана голосом и пр.)
- **Input methods** – поддержка различных методов ввода (в том числе методов, не связанных непосредственно с печатью на клавиатуре)
- **Image I/O** – поддержка ввода и вывода изображений
- **Sound** – поддержка звука

1.11 Технологии развертывания

- **Java Web Start** – распространение приложений через web.
- **Applet/Java Plug-in** – поддержка Java-кода, внедряемого в веб-страницы (апплетов)

- Вопреки распространенному мнению, широко используются в промышленности.

1.12 инструменты

- **java** – собственно исполняющая система
- **Javac** – компилятор
- **Javadoc** – генератор документации
- **Jar** – архиватор в формате jar (zip с метаданными)
- **Javap** – дизассемблер класс-файлов (в мнемонику java байткода)
- **JPDA** – дебаггер (Java Platform Debugger Architecture). Включает технологии
 - **JVMTI** – JVM Tool Interface, интерфейс доступа к ресурсам виртуальной машины, позволяет узнать информацию о состоянии и контролировать выполнение программы
 - **JDWP** – Java Debug Wire Protocol
- **Инструменты мониторинга виртуальной машины**
 - **Java Visual VM (jvisualvm)**
 - **JMC** – Java Mission Control
 - **JFR** – Java Flight Recorder
 - **Jconsole**
 - **Jstack** – мониторинг потоков
 - **Jstat, jstatd** – мониторинг статистики производительности

1.13 инструменты (cont.)

- Java DB – Встроенная база данных, на основе Derby
- Int'l – средства интернационализации, native2ascii
- Средства удаленного вызова процедур
 - IDL – язык определения интерфейсов idlj, orbd
 - RMI – сервер удаленного вызова, tnameserv
- Deploy
 - Javafxpackager средства развертывания javafx программ
 - Pack200 архиватор
 - Unpack200 деархиватор
- Security
 - поддержка сертификатов (jarsigner, keytool)
 - policytool
 - kerberos (kinit,klist,ktab)
- Troubleshoot – сборное название средств диагностики
- Scripting (jjs) - джаваскрипт
- Web Services
 - поддержка WSDL (schemagen, xjc,wsgen,wsimport)

1.14 Hello, world!

```
package sample; //это - пакет
```

```
public class Sample { //это - имя класса, класс имеет публичный доступ
```

```
public static void main(String[] args) { //точка входа в приложение
```

```
    System.out.println("Hello, World!"); //Вывод на экран
```

```
    //System - класс, представляющий систему
```

```
    //out - член класса, ссылка на поток вывода в консоль
```

```
    //(объект класса PrintWriter), в данном случае - на экран
```

```
    //println - метод класса PrintWriter
```

```
    }
```

```
}
```

1.15 Как запустить Hello World?

- ⦿ Класс `Sample` должен быть в файле `Sample.java` в каталоге `sample`
 - В каждом `java` файле только один `public` класс
- ⦿ `Javac -classpath . sample/Sample.java`
- ⦿ `Java -classpath . sample.Sample`

1.16 Q & A