

Java-технологии

- История и эволюция Java
- Особенности Java
- Разновидности Java
- Экосистема Java
- Язык Java
- Стандартный инструментарий 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 ТВ-приставок для кабельного телевидения

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

1996 Java Development Kit 1.0

1996 Java Development Kit 1.0

1997 JDK 1.1

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

2017 Java SE 9

2018 Java SE 10

Виртуальная машина и байткод

- **Подход C/C++:**

исходный код → машинный код → процессор

- программа работает только на той платформе, под которую скомпилирована

- **Подход Java:**

Исходный код → байткод виртуальной машины → виртуальная машина → процессор

- программа работает на любой платформе, где есть виртуальная машина Java

- “Write once, run anywhere!”

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 всего в 1.5-2 раза медленнее C, а в некоторых тестах не хуже или даже быстрее!

Сборка мусора

- Подход C/C++

выделил память → поработал → освободил память

- все управление памятью в руках программиста

- Подход Java

выделил память → поработал → забыл

- 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)

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

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

Реализация Java

- Oracle Java

<http://java.oracle.com/>

- Open JDK

<http://openjdk.java.net/>

- IcedTea

<http://icedtea.classpath.org/>

- IBM J9

Библиотеки

- Библиотеки общего назначения
Apache Commons Lang, Google Guava, Joda Time
- Библиотеки для логирования
Apache Log4J, SLF4J, LogBack,...
- Библиотеки для модульного тестирования
JUnit, TestNG, Mockito, JMock, Hamcrest,...
- Библиотеки для работы с байткодом
ASM, BCEL, Javassist,...

Системы сборки

- Apache Ant

Императивное описание сборки

- Apache Ivy

Управление зависимостями для Ant

- Apache Maven

Декларативное описание сборки и управление зависимостями

Среды разработки

- Eclipse IDE
 - IntelliJ IDEA
 - NetBeans IDE
-
- Подсветка синтаксиса
 - Автодополнение, гиперссылки
 - Рефакторинг
 - Интерактивный отладчик

Альтернативные языки

- Groovy, Kotlin, Scala
- Clojure
Диалект LISP
- Jruby
Реализация Ruby на JVM
- Jython
Реализация Python на JVM

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 "Hallo, world!" and terminates  
 * @author VBayrasheva  
 */  
public class HelloWorldWithComments {  
    /* Canonical example program since 1978,  
     many thanks to K&R */  
    public static void main(String[] args) {  
        // print our hello  
        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]);  
        }  
    }  
}
```

Спасибо за внимание!