ORACLE®

ORACLE®

Платформа Java SE: введение

Андрей Дмитриев
andrei.dmitriev@oracle.com

Программа

- **Платформа JavaSE.**
- **Язык Java.**
- **С чего начать?**

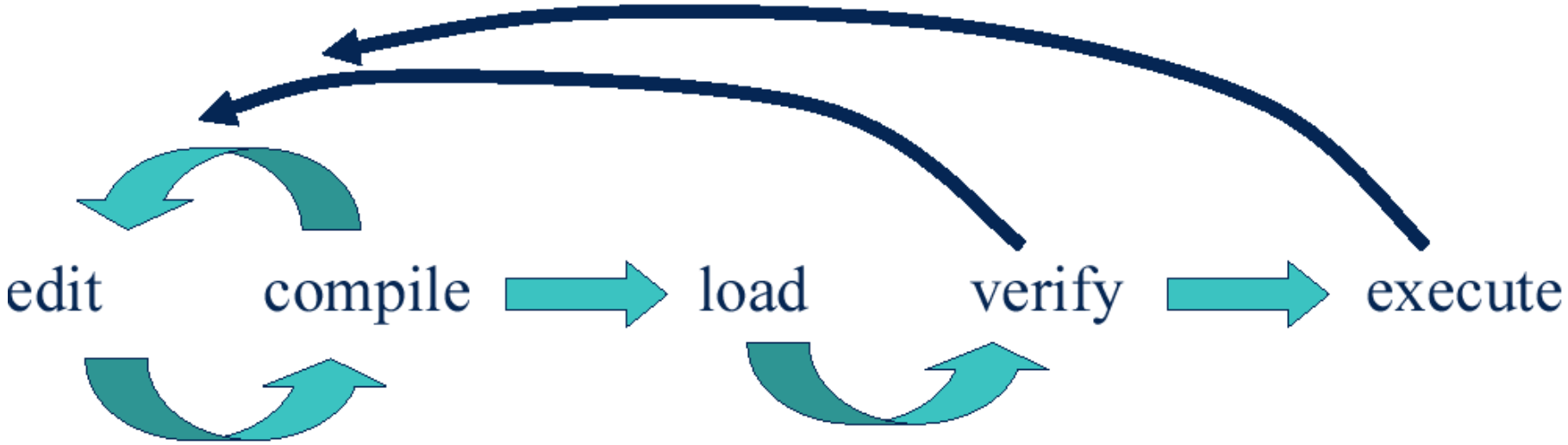
- **Вопросы и ответы.**

Платформа Java SE

- Почему платформа?
- Варианты исполнения:
  - Java SE
  - Java EE
  - Java ME
- Операционные системы:
  - Windows
  - Linux
  - Solaris
  - MacOS
  - и др.

Разработка Java программы

Что значит, что Java – интерпретируемый язык?



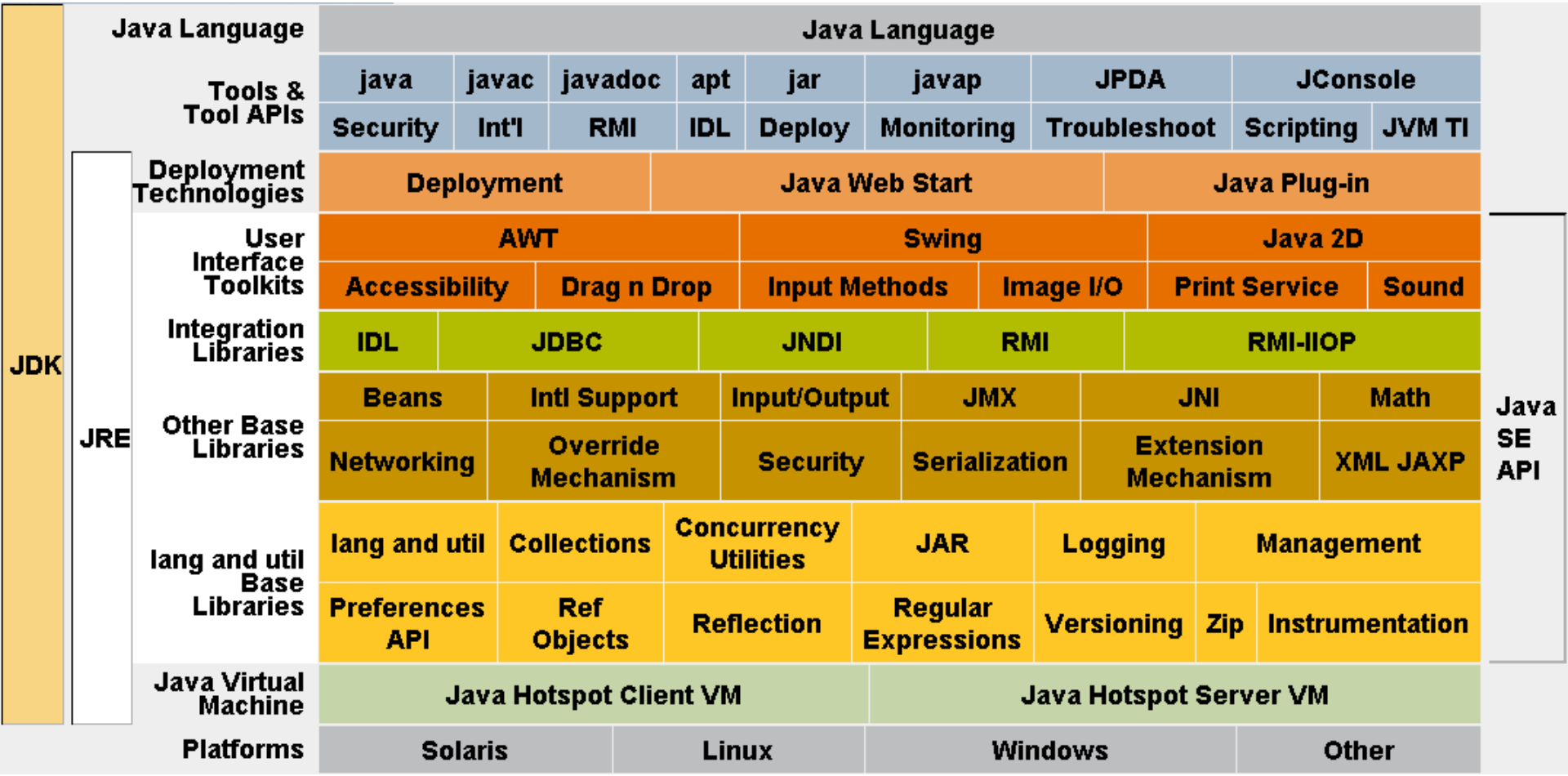
Язык Java

- Простой
- Объектно-ориентированный
- Распределенный
- Интерпретируемый
- Надежный
- Безопасный
- Архитектурно-нейтральный
- Переносимый
- Многопоточный

Платформа Java

- Средства разработки существуют для большинства аппаратных платформ.
- Виртуальная машина Java (Java Virtual Machine, JVM) гарантирует единообразие интерфейса с операционной системой.
- Переносимость: «Write once, run everywhere».
- Поставляется с богатой библиотекой классов JDK (Java Development Kit).
- JRE (Java Runtime Environment) – среда, позволяющая запустить программу, написанную на языке Java.

Общий вид платформы Java SE



Отличия от C++

- Перегрузка операторов
- Множественное наследование
- Автоматическое согласование типов
- Адресная арифметика
- Деструкторы
- …

Google: “java c++ differences”

Переносимый код Java

- Программы распространяются в виде класс-файлов или jar-пакетов.
- Класс-файл содержит промежуточный код (байт-код).
- Байт-код – это набор данных и последовательность инструкций для виртуальной машины Java.
- Класс-файлы выполняются виртуальной машиной (JVM).
- Структура класс-файлов может меняться параллельно с изменениями в JVM.

Инструментарий

В состав JDK входит набор утилит для создания Java приложений.

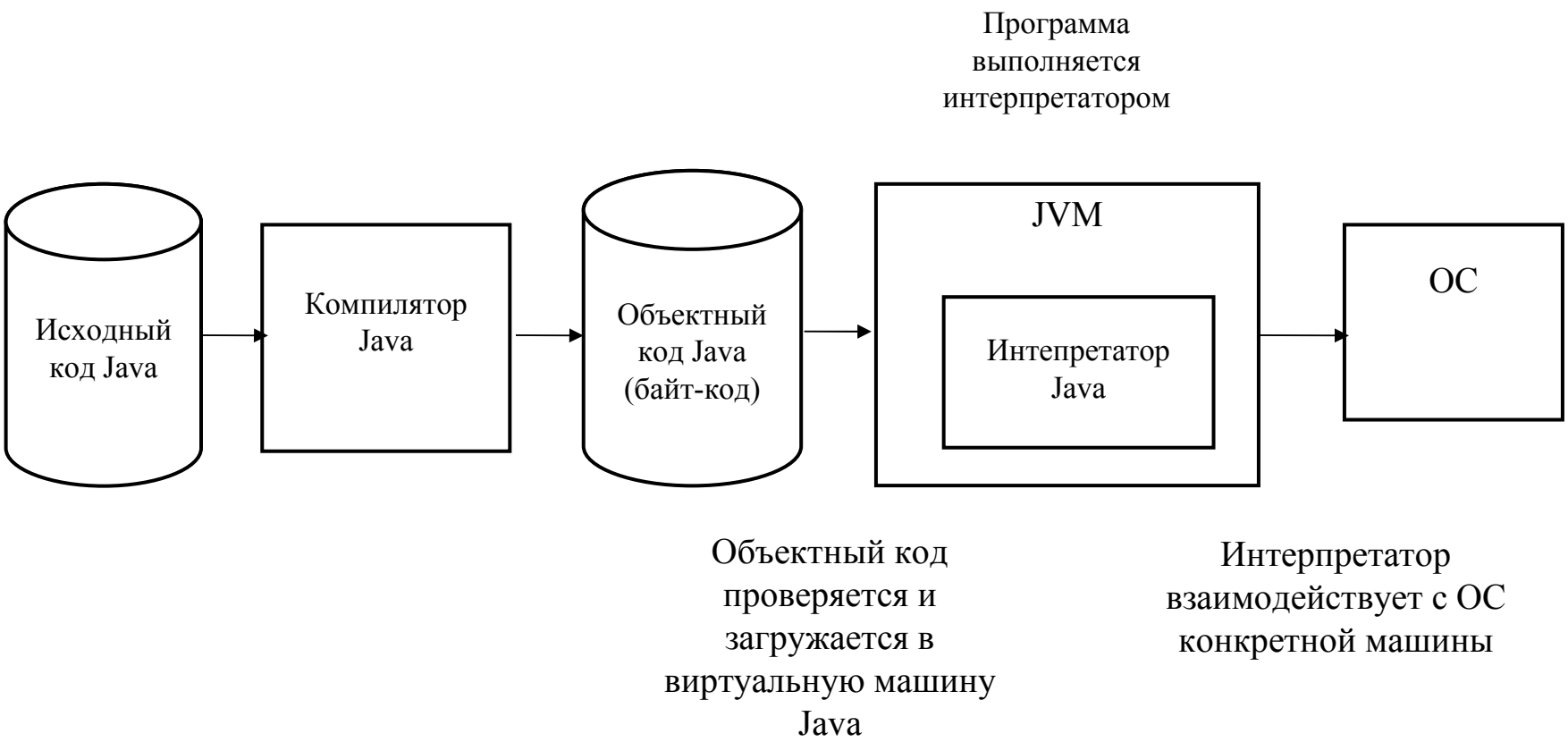
javac	Компилятор языка Java. Преобразует исходный код в промежуточный байт-код.
java	Интерпретатор байт-кода. Запускает класс на выполнение.
javadoc	Утилита формирования стандартной документации JavaDoc.
javah	утилита создания заголовочные файлов для интеграции.
jar	Утилита для создания дистрибутивов Java программ.
javap	Дизассемблер

Среда разработки

Среда разработки предлагает редактор, компилятор и набор вспомогательных программ, упрощающих разработку ПО.

- Eclipse http://www.eclipse.org/
- NetBeans http://www.netbeans.org/
- IntelliJ Idea http://www.jetbrains.com/
- JBuilder, CodeGear http://www.borland.com/

Создание и запуск приложения



Создание и запуск приложения (cont.)

- В текстовом редакторе создать файл с исходным кодом программы на языке Java: Hello.java

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

- Скомпилировать исходный код в промежуточный код командой
- В результате получится файл Hello.class
- Запустить приложение командой

Создание и запуск приложения (cont.)

Если запускаемый класс принадлежит именованному пакету (пространство имен, которому принадлежит класс, package), например,

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

то…

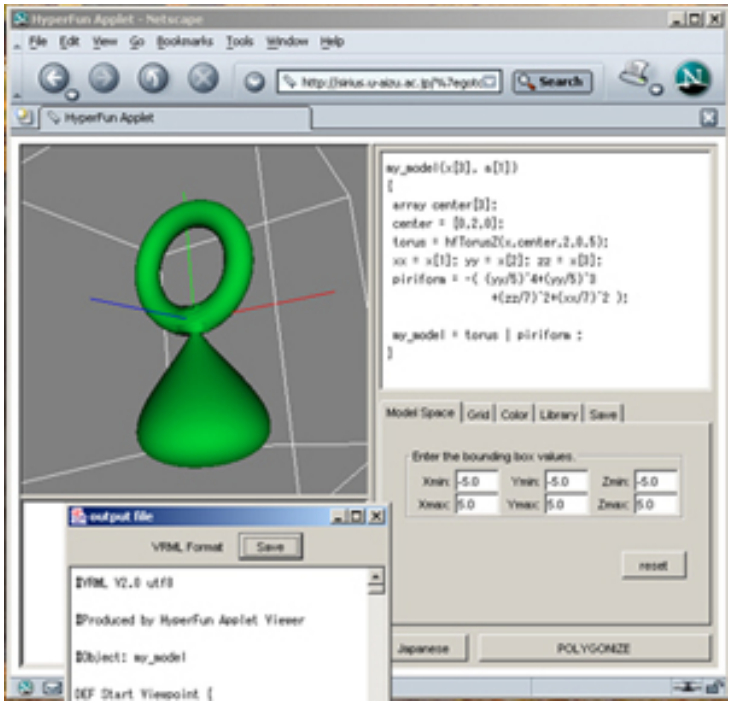
Создание и запуск приложения (cont.)

...то скомпилированный класс необходимо поместить в папку, имя которой совпадает с именем указанного в классе пакета (если пакеты вложены друг в друга, то в несколько вложенных каталогов).

Запуск программы командой:

Типы приложений

- Самостоятельное приложение работает под управлением виртуальной машины
  - Сюда же: jnlp протокол
- Аплет встраивается в другое приложение и находится с его адресном пространстве



Параметры компиляции

Параметры компиляции можно изменять при помощи ключей компилятора javac:

-cp –classpath	указать путь, по которому можно найти классы, необходимые для компиляции (переменная CLASSPATH)
-bootclasspath	указать путь, по которому можно найти классы, необходимые для запуска JVM
-source	указать версию исходного кода
-target	указать версию JVM, для которой создается класс-файл
-version	вывести версию компилятора
-help	перечень разрешенных опций компилятора
-J	свойство, передаваемое в JVM. Виртуальная машина может изменять свое поведение в зависимости от переданных параметров.
-X	дополнительные опции

Параметры запуска

Параметры исполнения можно изменять при помощи ключей интерпретатора, передаваемых java:

-cp(-classpath)	указание пути, по которому содержатся классы, необходимые для запуска
-D	установка системного свойства
-X	расширенные параметры
-agentlib	загрузка отладочного агента
-javaagent	загрузка Java агента
-client/-server	выбор клиентской или серверной модификаций JVM

Исполняемые пакеты

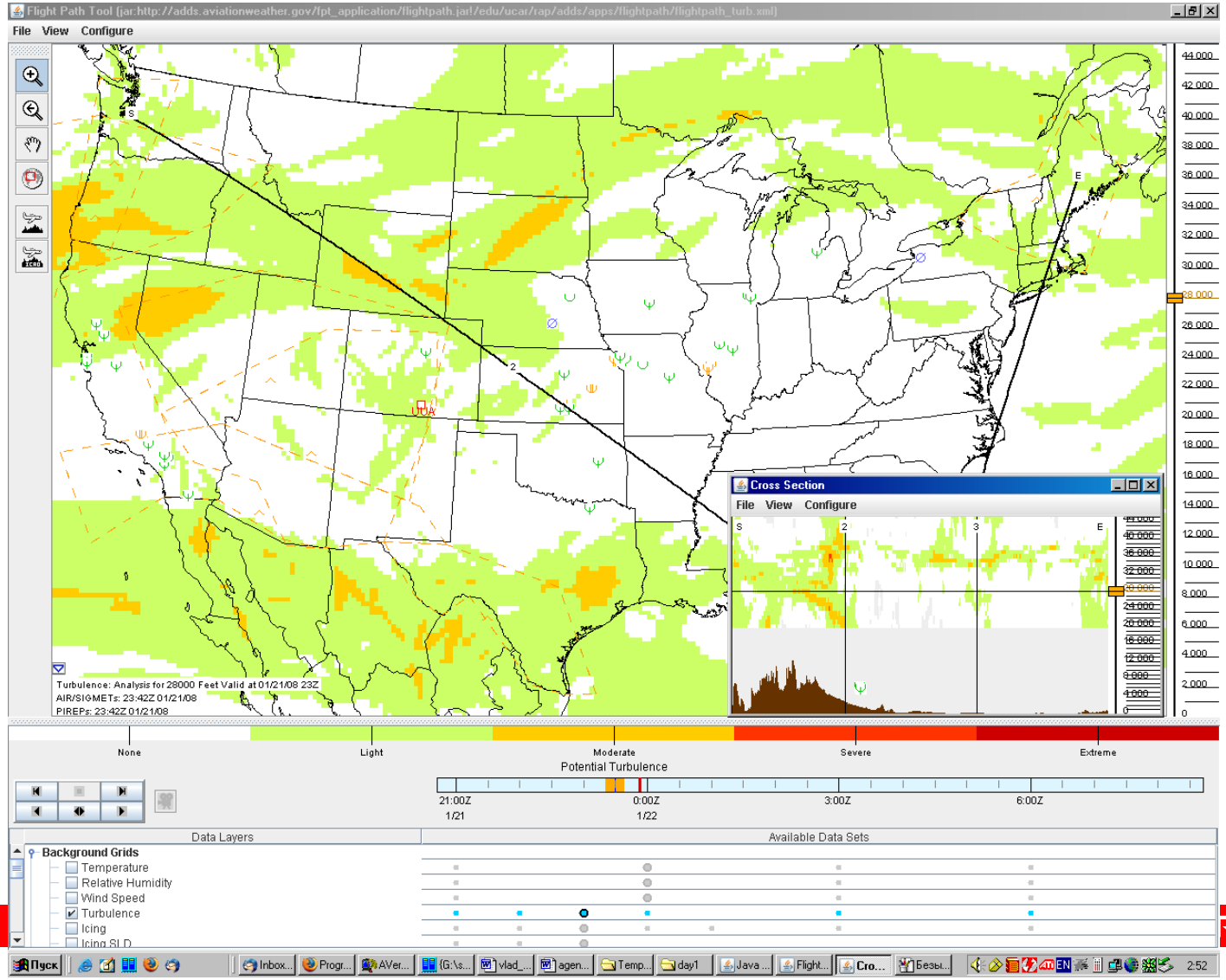
- Классы можно объединять в jar-пакеты.
- Запуск программы из такого пакета происходит быстрее.
- Пакет может содержать ресурсы для приложения.
- Создание архива утилитой jar:

```
jar cf Hello.jar Hello.class
 Hello.java
```

Запуск приложения, запакованного в jar фаил:

```
java -jar Hello.jar
```

Avia Fligh Path tool



Правда ли что…

- Формат class-файла неизменен?
- Программа на языке Java будет работать на всех машинах, снабженных JDK?
- С помощью JRE невозможно написать и запустить программу на Java?
- Утилита jar использует тот же алгоритм компрессии что и zip?
- Программа на Java сначала компилируется, а потом интерпретируется?

С чего начать?

- Домашняя страница Java:
  - http://java.sun.com
- Java JDK 6:
  - http://java.sun.com/javase
- Java Tutorial:
  - http://java.sun.com/docs/books/tutorial/index.html
- Java API:
  - http://java.sun.com/javase/6/docs/api/
- Форум:
  - http://forums.java.net/
- Статистика использования языков:
  - http://www.langpop.com/

Questions & Answers

ORACLE®