

Мультимедийный учебный курс  
**Программирование на JAVA**  
Часть 1

Лекция 1. 1

**ВВЕДЕНИЕ В ДИСЦИПЛИНУ**

## Лектор

- **Борисенко Виктор Петрович**
- к.т.н., доцент кафедры ЭВМ
- E-mail - [vborisenko07@gmail.com](mailto:vborisenko07@gmail.com)

# Литература

- **Шилдт, Герберт.** Java 8. Полное руководство; 9-е изд.: Пер. с англ. - М. : ООО "И.Д. Вильямс", 2015. - 1376 с.
- **Блинов И.Н., В.С. Романчик В.С.** Java – Промышленное программирование.- Минск, УниверсалПресс, 2007.- 727 с.
- **Бондарев В.М.** Учебное пособие по программированию на Java.- Харьков: ООО «Компания Смит», 2003. - 296 с.
- **Брюс Эккель.** Философия Java. 4-е издание.- СПб.: Питер, 2009.- 640с.
- **Хорстман К. С., Корелл Г.** Библиотека профессионала JAVA 2. Том 1. Основы: Пер. с англ. – М.: Издательский дом "Вильяме", 2007. - 896 с.:
- **Хорстман К. С., Корелл Г.** Библиотека профессионала JAVA 2. Том 2. Тонкости программирования.: Пер. с англ. – М.: Издательский дом "Вильямс", 2007. — 1168 с.;

# Internet - ресурсы

- [java.sun.com](http://java.sun.com)
- <http://www.oracle.com/technetwork/java/>
- <http://eclipse.org/>

# Состав курса

- ❖ Часть 1 - Основы программирования на Java (**core Java Standard Edition**)
- ❖ Часть 2 - Программирование клиент-серверных приложений Java (**core Java Enterprise Edition**)

# Применимость Java

- ❖ Разработчики ПО более 21 млн.
- ❖ Используется в более чем в 7 млрд. устройств. Среди НИХ:
  - ✓ более 800 млн. ПК;
  - ✓ 3,1 млрд. мобильных телефонов и других портативных устройств;
  - ✓ 4,5 млрд. смарт-карт;
  - ✓ телеприставки, принтеры, веб-камеры, игры, автомобильные системы навигации, игровые автоматы, медицинское оборудование, парковочные автоматы и т. д.

# Предложения на рынке труда

Адрес	Jav	C#	Delph	C++
jobserve.co.uk	154	102	116	156
rabota.com.ua	41	31	14	37
job.kiev.ua	169	109	84	140
jobpilot.de	66	119	4	29
jobs.com	>500	>500	35	>500
m	0	0	3	0



# История языка Java. "Зеленый" проект.

## Sun Microsystems

- Green Project -1991
- \*7 (1992) - устройство с 5" цветным (16 бит) сенсорным экраном
- Oak (создатели – Джеймс Гослинг, Патрик Ноутон, Крис Вирт, Эдом Франк, Майк Шеридан)



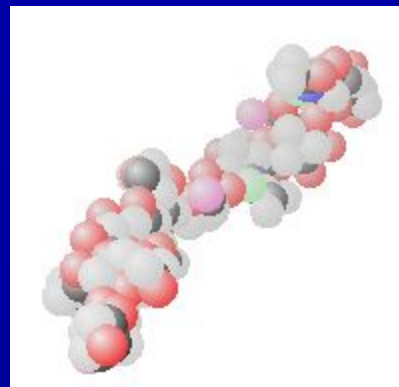
персонаж по имени Дьюк (Duke)





# Выход в Internet

- HotJava (WebRunner) – 1994
- Oak = Java (начало 1995)
- Первая презентация – 1995
- JDK (*Java Development Kit*) 1.0a2 – март 1995



# Инструментарий разработчика Java-приложений

*Java Development Kit* (JDK) – примитивная среда разработки:

- Библиотеки классов и интерфейсов
- Компилятор (**javac.exe**)
- Интерпретатор (**java.exe**)
- Отладчик (jdb.exe)
- Архиватор (**jar.exe**)
- Документатор (javadoc.exe) и др.

Начиная с версии JDK 1.2 три редакции

- J2SE (**Standard Edition**) - для программирования настольных приложений
- J2EE (**Enterprise Edition**) - для создания сложных, высоконадежных, клиент-серверных приложений
- J2ME (**Micro Edition**) – для программирования бытовой техники (например, сотовых телефонов)

# Новейшая история

- JDK 1.0 (Java12)– 1995
- J2SE 1.1 - 1996
- JDK 1.2 (Java 2)– 1997
- J2SE 1.3 - 2000
- J2SE 1.4 – 2003
- J2SE 1.5 (Tiger) – 2004
- 23. марта 2005 г. – 10 лет Java!
- J2SE 1.6 – 2010 год
- J2SE 1.7 - 2011 год
- J2EE 1.7 - 2013 год
- J2SE 1.8 UP-2014 год



# Характеристики языка Java

- объектно-ориентированный
- платформо-независимый
- простой
- устойчивый
- многопоточный
- интерпретируемый
- распределенный
- динамический

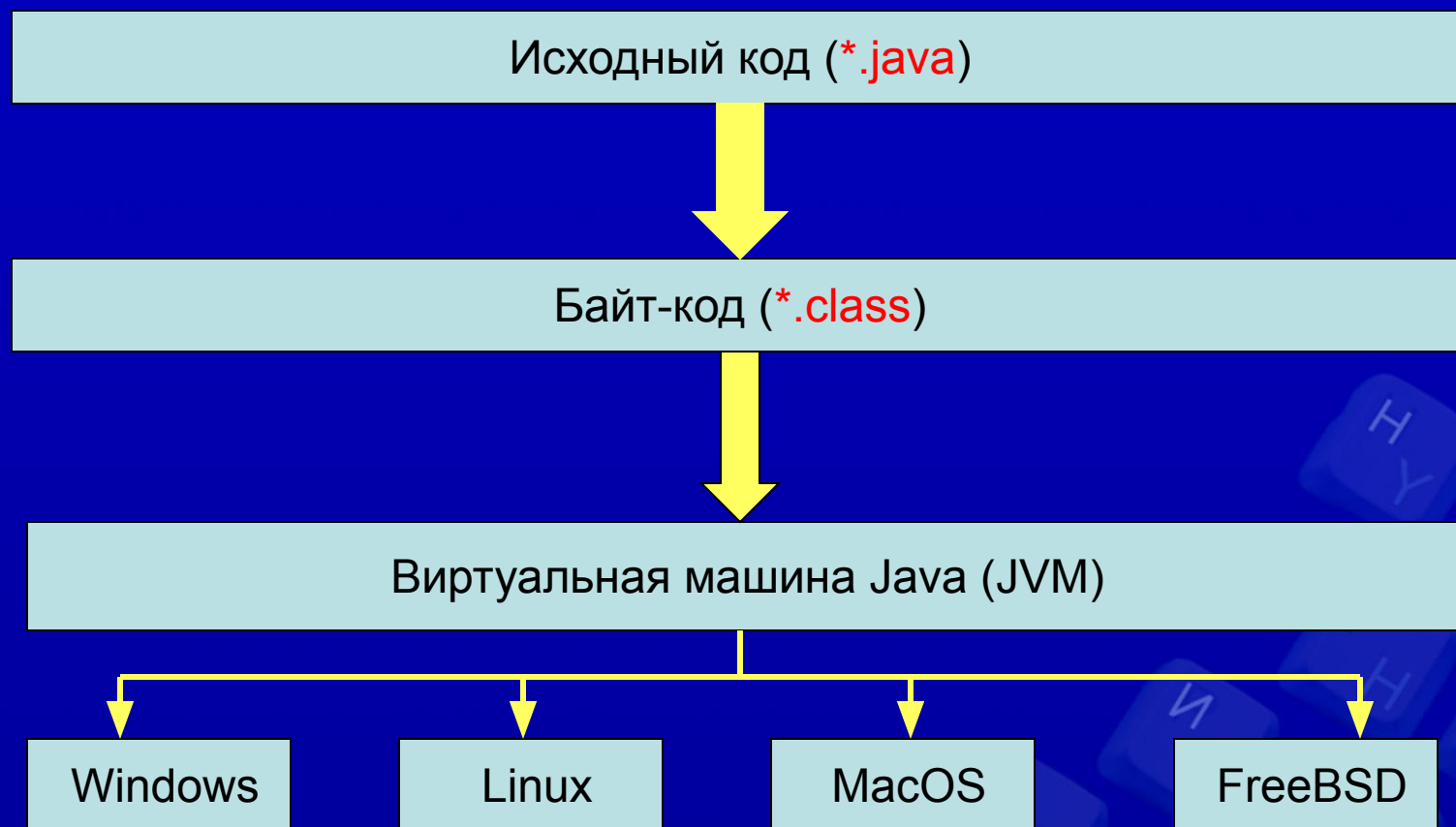
# Характеристики языка Java

- ❑ Объектно-ориентированный, в Java даже нет глобальных переменных или функций, есть только поля и методы классов
- ❑ Платформо-независимый т.е. не ориентирован на какую-то одну аппаратную или программную архитектуру
- ❑ Простой (нет указателей, нет операторов освобождения памяти, нет шаблонов, нет множественного наследования).
- ❑ Устойчивый (проверяет выход за границу массива, не только предоставляет аппарат исключений, но и требует от программиста их обязательной обработки).
- ❑ Многопоточный, средства работы с потоками встроены в язык.
- ❑ Интерпретируемый, выполнение программы происходит путем интерпретации частично откомпилированного байт-кода.
- ❑ Распределенный (реализует удаленные вызовы методов).
- ❑ Динамический (возможность динамически связывать код и обновлять байт-код на уровне исполнительной системы).





# Этапы программирования на языке Java



# Функции виртуальной машины Java

- интерпретация байт-кода
- проверка ограничений безопасности при выполнении байт-кода
- автоматическая "сборка мусора" (garbage collection)
- обеспечение многопоточного выполнения



# Виртуальная машина Java

- **Стек** – область памяти для хранения программ
- **Механизм сборки мусора** (из кучи)
- **Область методов** и **область констант**

————— 4 Гб —————

- **Регистры** для управления системным стеком

# Типы программ Java

- автономные настольные приложения
- апплеты (applets)
- Web-приложения
  - сервлеты
  - серверные страницы Java (JSP)
- компоненты EJB
- Web-сервисы
- мидлеты (MIDlets)
- JavaCard

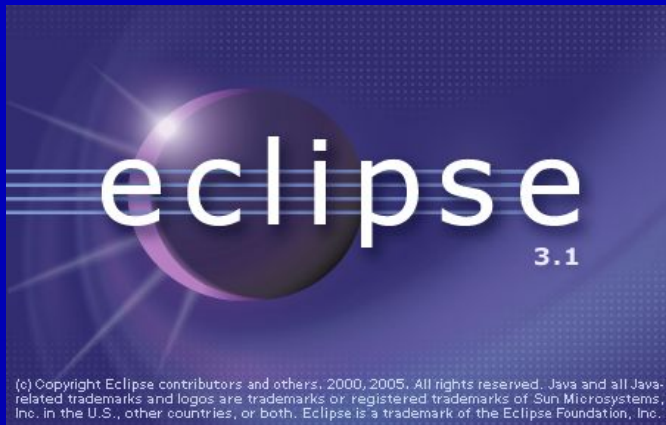
# Java Community Process (JCP)



- <http://www.jcp.org>
- **Исполнительный комитет** по выработке рекомендаций и управлению развитием платформ технологии Java
- **JCP** предполагает создание групп экспертов, которые сотрудничают в целях **определения новых спецификаций и усовершенствования уже существующих**. Система принятия решений с помощью голосования гарантирует, что **Java остается единой и общей платформой для всех** без каких-либо предпочтений
- J2SE/EE – Oracle (Sun Microsystems) + 15 компаний (Intel, IBM, Apple, SAP, Apache, Fujitsu, HP, Google и др.)
- J2ME – Oracle (Sun Microsystems) + 15 компаний (Nokia, Siemens, Motorola, Samsung и др.)



# Интегрированная среда разработки Eclipse



<http://www.eclipse.org>

- open-source
- подсветка синтаксиса в редакторе
- компиляция кода
- отладчик уровня исходного кода с поддержкой threads
- файловый менеджер и менеджер проектов
- поддержка модульного тестирования с помощью JUnit
- интеграция с инструментом сборки приложений Jakarta Ant
- возможность расширения (plugins)



# Первая программа- приложение

Пример программы (автономного приложения Java), которая выдает на экран сообщение

**Шаг 1.** Подготовим программу в текстовом файле **Hello.java**:

↓

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

**Шаг 2.**

Создадим рабочий каталог, сделаем его текущим и поместим туда файл с программой

**Шаг 3.** Откомпилируем приложение командой

**>javac Hello.java**

В результате в рабочем каталоге должен появиться файл Hello.class, который содержит нашу программу, откомпилированную в так называемый байт-код



**Шаг 4.** Выполним вызов на выполнение (интерпретацию) полученного байт-кода командой

```
>java Hello
```

**В результате на экран консоли будет выведено:**

```
My first Programm!
```