



# Програмирование на Java

Лекция №1. Введение



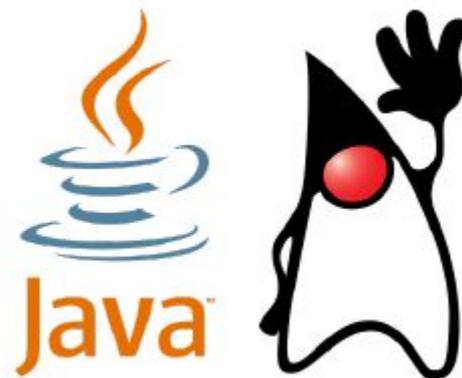
# План лекции

1. Что такое Java?
2. История и эволюция Java
3. Особенности Java
4. Инструменты разработчика

# Почему стоит изучать Java?

1. Это один из самых востребованных языков программирования. Занимает верхние строчки индекса TIOBE (см.

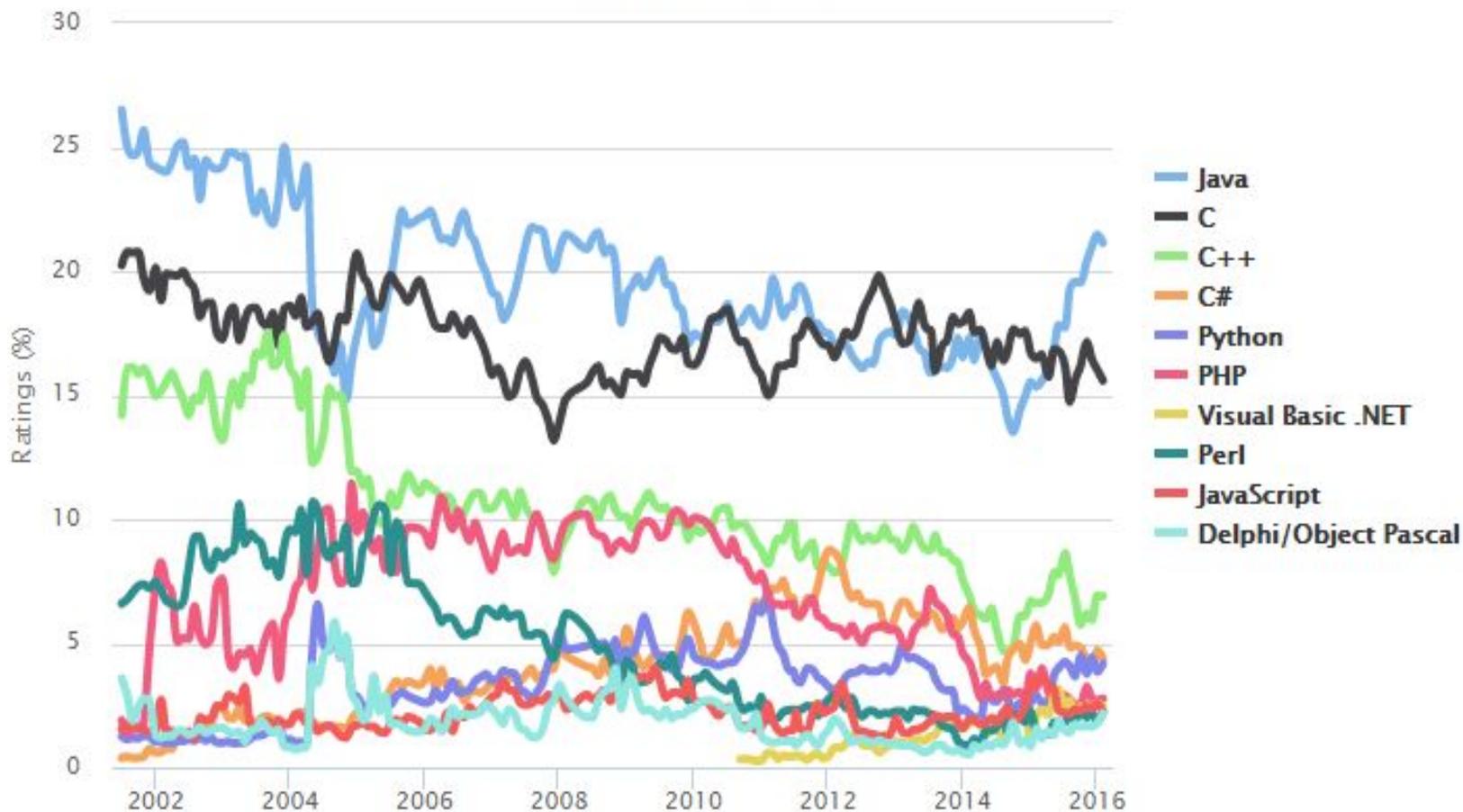
[http://www.tiobe.com/index.php/tiobe\\_index](http://www.tiobe.com/index.php/tiobe_index) )



# Почему стоит изучать Java?

## TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)



# Почему стоит изучать Java?

## 3. На Java пишут:

- Высоконагруженные системы (Одноклассники)
- Корпоративные приложения (Confluence, JIRA)
- Настольные приложения (Minecraft)
- Программы и игры для телефонов, в том числе под Android
- Апплеты для смарт-карт
- Интерактивный контент для Blu-ray

## 4. Язык развивается и совершенствуется

## 5. Java – это

- Язык программирования;
- Обширная стандартная библиотека;
- Инструменты разработки (сборка, тестирование);
- Методология ООП;
- Платформа для альтернативных языков (Clojure, Groovy, Jruby, Kotlin, Scala)

# История и эволюция

## Исторические предпосылки

*Конец 1980 гг. – начало 1990 гг. основное средство ОО программирования – C++*

- Поздняя стандартизация
- Сильная зависимость от реализации и платформы, отсутствие стандарта на бинарное представление
- Достаточно высокая сложность
- Отсутствие стандартной библиотеки, содержащей средства сетевого и меж процессного взаимодействия и построения графического интерфейса
- Классическая модель построения систем – компиляция, связывание, построение образа процесса

# История и эволюция

1991 внутренний проект Sun Microsystems по созданию платформы для

разработки встраиваемых систем — Green Project;

вместо C++ решили создать новый язык, названный Oak (дуб)

Язык программирования Oak (1991) James Gosling, Patric Naughton, Chris Warth, Ed Frank, Mike Sheridan, Sun Microsystems, Inc

- Разработка – 18 месяцев (первый компилятор – осень 1991г.)
- Платформенная независимость, переносимость, виртуальная машина
- Первоначальный акцент на «бытовую электронику», микроконтроллеры
- 1992 первое демонстрационное устройство на новой платформе — PDA Star7
- 1993 попытка занять нишу ТВ-приставок для кабельного телевидения
- Момент совпал с бурным развитием Internet, WWW
- 1993г. – идея использования в сети Internet



James Gosling

# История и эволюция

- 1994 фокус на разработке интерактивных приложений (апплетов) для
  - веб-страниц;
  - язык переименован в Java
- Java Development Kit 1.0 ( 26 августа 1996 года)
- 1997 JDK 1.1, JIT-компиляция
- 1998 J2SE 1.2, «Java 2», разделение на ME/SE/EE
- 2000 J2SE 1.3
- 2002 J2SE 1.4
- 2004 J2SE 5.0 (сентябрь 2004), изменение нумерации, перечислимые типы, аннотации, средства обобщенного программирования, foreach

# История и эволюция

- 2006 Java SE 6 (11 декабря 2006), уход от понятия «Java 2» добавлены коллекции, JConsole (графический мониторинг JVM, стала официально поддерживаемой утилитой), Java HTTP Server (позволяет создать полноценный HTTP сервер, с минимально необходимыми функциональными свойствами)
- 2011 Java SE 7 (28 июля 2011), Поддержка динамически-типизированных языков, Поддержка Unicode 6.0, Изменение синтаксиса языка Java (Project Coin)
- 2014 Java SE 8 (19 марта 2014), Поток (stream) для работы с коллекциями, Полноценная поддержка лямбда-выражений

# Особенности

## Традиционный подход:

- исходный код → машинный код → процессор
- программа работает только на той платформе, под которую она 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!”

# Особенности

## • 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 8 всего в 1.5–2 раза медленнее C, а в некоторых тестах не хуже или даже быстрее!

# Особенности

## Сборка мусора

### Подход C/C++:

- выделил память → поработал → освободил память
- всё управление памятью в руках программиста

### Подход Java:

- выделил память → поработал → молодец
- виртуальная машина считает ссылки на каждый объект
- освобождает память, когда ссылок больше нет

# Особенности

## 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.

# Особенности

## • 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)

# Особенности

- Простота
- Безопасность
- Переносимость, независимость от архитектуры
- ООП
- Многопоточность
- Высокая производительность
- Распределенность
- Динамичность (reflection)
- *Интерпретируемость (до версии 1.3)*

# Особенности

- Простота
- Безопасность
- Переносимость, независимость от архитектуры
- ООП
- Многопоточность
- Высокая производительность
- Распределенность

# Среды разработки

- **Eclipse** (IBM, Eclipse Foundation, Бесплатная среда с открытым исходным кодом. Может быть свободно загружена с сайта <http://www.eclipse.org/>. Является основой (платформой) для многих коммерческих сред разработки)
- **IntelliJ IDEA** (JetBrain, Коммерческая среда. Очень популярна в западных странах. Пробная версия может быть свободно загружена с сайта <http://www.jetbrains.com/idea/>.)
- **NetBeans** (Sun, [netbeans.org](http://netbeans.org), Бесплатная среда с открытым исходным кодом. Может быть свободно загружена с сайта <http://www.netbeans.org/>.)

# Среды разработки

- Jbuilder (Borland, CodeGear, Бесплатная редакция и пробные версии коммерческих могут быть свободно загружены с сайта <http://www.codegear.com/>.)
- Jdeveloper (Oracle, Бесплатная среда, ориентированная на работу с базами данных Oracle на основе технологий Java EE, но может использоваться как универсальная среда разработки для Java. Может быть свободно загружена с сайта <http://oss.oracle.com/jdeveloper.html>.)

**Подсветка синтаксиса, Автодополнение,  
гиперссылки, Рефакторинг,  
Интерактивный отладчик**

# Среды разработки

## Java Runtime Environment (JRE)

- виртуальная машина и стандартная библиотека классов для запуска скомпилированных программ

## Java Development Kit (JDK)

- набор инструментов для разработчиков, включает в себя JRE

# Среды разработки

Имеются три типа JDK:

Java ME — комплект Java Micro Edition, предназначенный для программирования "тонких аппаратных клиентов";

Java SE — комплект Java Standard Edition, предназначенный для программирования обычных компьютеров;

Java EE — комплект Java Enterprise Edition, предназначенный для написания серверного программного обеспечения.

# Среды разработки

## Инструментальные средства разработки JDK:

javac	компилятор
java	интерпретатор
jre	интерпретатор для конечных пользователей
jdb	отладчик
jvisualvm	профилировщик
javah	генератор файлов заголовков и исходных текстов на C
javap	дизассемблер классов
javadoc	генератор документации
appletviewer	программа просмотра апплетов
jar	программа, позволяющая упаковать множество классов в исполняемый архив
javaws	программа, осуществляющая загрузку и запуск приложений с удалённых web-серверов
jconsole	консоль, предназначенная для мониторинга и управления исполнением приложений
jmap	вывод карты памяти процесса
jps	вывод информации о запущенных процессах

# Среды разработки

keytool	программа, управляющая ключами и сертификатами
native2ascii	конвертер файлов в ASCII-формат
orbd	ORB-сервер
servertool	консоль администратора ORB-сервера
pack200	архиватор
unpack200	программа для распаковки архивов
policytool	программа для чтения и модификации policy-файлов
rmic	RMI-компилятор
rmid	RMI-сервер
rmiregistry	реестр RMI-объектов
schemagen	генератор XML-схем
serialver	вывод serialVersionUID классов в CLASSPATH
tnameserv	служба имён
wsgen	генератор stubs для веб-сервисов
wsimport	генератор ties для веб-сервисов
xjc	генератор классов на основе XML-схем

# Среды разработки

Примеры:

- `java -version`
- `javac Welcome.java`
- `java Welcome`