

# Прикладное программирование

# Кто я?

**Новосельцев Игорь Андреевич**

Выпускник **РФИКТ**,

Специальность: **Компьютерная безопасность**

Старший преподаватель,

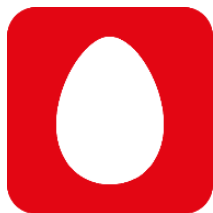
Кафедра телекоммуникаций и информационных технологий;

Магистр физико-математических наук.

## **Мои интересы:**

- Микроконтроллеры и микропроцессорная техника;
- Робототехника и протезирование;
- Разработка мобильных приложений.

# Как меня найти?



+ 375-29-8707234



@IgorPrestolov



inovoseltsev@outlook.com

# Чем так интересен курс?

В рамках этого курса, мы разберем:

- Ключевые особенности языка Java
- Основные возможности применения Java
- Что из себя представляют **принципы ООП**
- Как использовать **Git (Систему контроля версий)**
- **Паттерны проектирования**
- Основы **сетевых технологий**
- Многочисленные **WEB-технологии**
- Как разрабатывать **сетевые приложения** на Java

# Чем еще так интересен курс?

А еще расскажу:

- Как правильно отлаживать работу программ
- Из чего состоят **пользовательские интерфейсы**
- Как **взломать** пользовательские приложения
- И (!) как **обеспечить их безопасность**
- Как самостоятельно написать **мессенджер**
- И как развернуть свой собственный **сервер**
- Как разрабатывать **многопоточные приложения**
- Как работать с **базами данных**, и зачем они нужны
- И многое другое...

# Лекции

## Объем курса:

- **40** часов лекций (**20 + 20**),
- **96** часов лабораторных (**48 + 48**).

## Форма отчетности:

- Отчет по всем лабораторным работам.
- **Зачет** в каждом семестре.

## Расписание лекций:

### **Суббота, аудитория 115**

- 1я лекция: **10.30 – 11.50**
- 2я лекция: **12.00 – 13.20**

# Лабораторные работы

Выбор уровня сложности лабораторных работ:

	<b>А</b>	<b>В</b>	<b>С</b>
Рекомендации по выбору варианта	<p><b>Оценка 4-6</b> по программированию за 1-ый курс.</p> <p>Кто хочет пройти курс с минимальными затратами времени и усилий.</p>	<p><b>Оценка 7-8</b> по программированию за 1-ый курс.</p> <p>Кого интересует программирование и кто готов учиться.</p>	<p><b>Оценка 8-10</b> по программированию за 1-ый курс.</p> <p>Кто хочет связать свою профессию с разработкой ПО.</p> <p>Кто готов и умеет работать самостоятельно.</p>


**p.s. Расписание лабораторных занятий согласовывается с руководителем практики.**

**ОБОЖАЮ ПРИНИМАТЬ ЛАБЫ УРОВНЯ «С»**



**А потом не зачет, а целый ЗАЧЕТИЩЕ!**



A full-page background image featuring the character Thanos from the Marvel Cinematic Universe. He is shown from the waist up, in a powerful, forward-leaning pose with his right arm extended and his left hand clenched into a fist. He is wearing his iconic purple and gold armor. The background is a bright, hazy orange and yellow, suggesting a fiery or explosive environment. The text is overlaid on the right side of the image.

Когда выстраиваешь  
баланс в  
Университете,  
обычно не до  
веселья...

Но ЗАЧЕТ...  
Это веский повод  
для улыбки 😊

# Дополнительная лекция

- Лекция на **свободные темы.**
- **1 раз в семестр по желанию.**

Как правило, лекция проходит в форме **дискуссии.**

На таких занятиях обсуждаем такие важные темы, как:

- Как написать крепкое **Резюме (CV)**
- Как выбрать подходящее **направление** в IT
- Как организованы рабочие процессы в компаниях
- Как проходить **собеседования** на курсы / на работу
- **Актуальные технологии** и т.д.

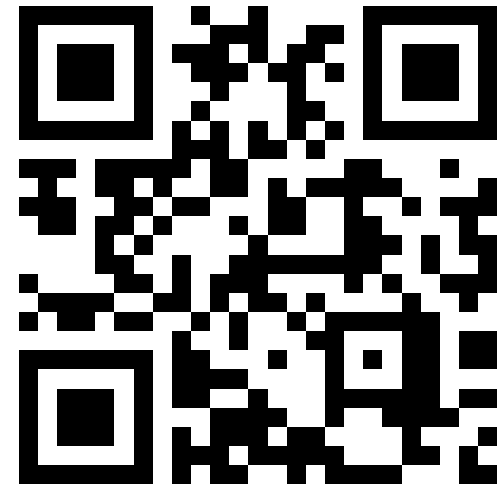
# Где найти лекции?

Канал «**Прикладное программирование (RF&CT)**» в Telegram:

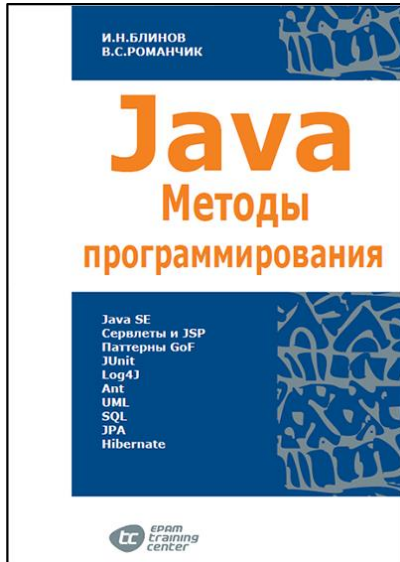
- [https://t.me/ASP\\_RFCT](https://t.me/ASP_RFCT)

**Что еще можно найти на канале?**

- Лекции (*пополняются еженедельно*),
- Расписание занятий;
- Новости;
- Информация по курсам;
- Актуальные вакансии;
- Полезные материалы.



# Какую литературу по Java выбрать?



## «Java. Методы программирования», Блинов, Романчик

Книга написана практикующими специалистами компании Eram.

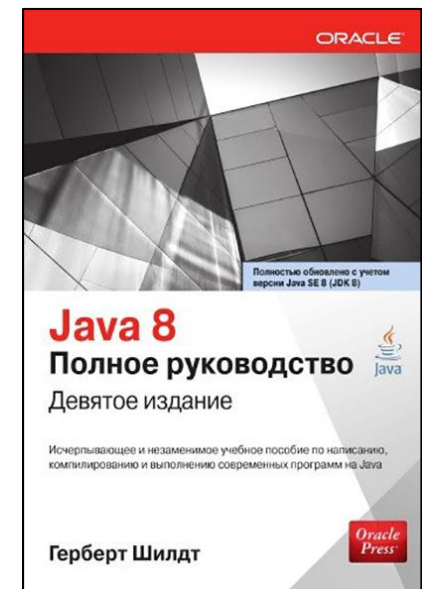
Она содержит большое количество практических задачи и примеров, а также в ней подробно расписаны технологии Hibernate для разработки распределённых систем, а также основы работы с HTML, XML, JavaScript.

## «Java 8. Полное руководство», Герберт Шилдт

В данной книге Вы найдете все, что только может пригодиться программисту на первых порах.

Она состоит из 17 глав, каждая из которых ознакомит вас с основами языка, его основными принципами и возможностями.

Также в книге затрагиваются принципы работы с библиотекой Swing и платформой JavaFX.





# Какую литературу по Java выбрать?



## «Effective Java», Джошуа Блох

Эта книга обязательна к прочтению всем, кто хочет взять Java на вооружение.

Книга содержит свод полезнейших правил, из которых станет очевидно, как сделать код максимально эффективным и чистым. Каждое правило сопровождается подробными примерами.

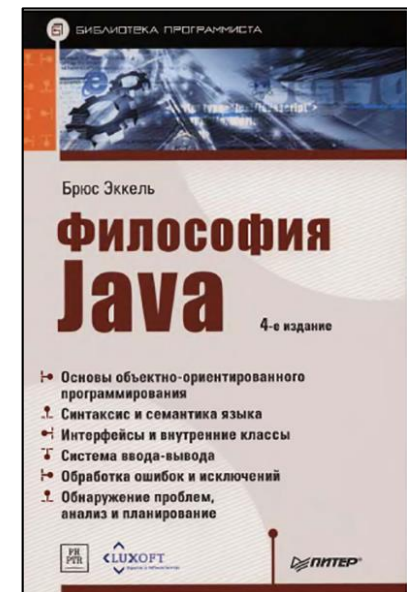
Также тут вы можете найти советы опытных разработчиков.

## «Философия Java», Брюс Эккель

Название этой книги говорит само за себя. Это неувядающая классика книг о программировании.

В книге присутствует глубокое изложение принципов Java от практикующего специалиста, а также приводится множество сравнений с C++.

Книга научит Вас как писать качественный объектно-ориентированный код.



# Документация Oracle

Обязательно загляните в **техническую документацию Oracle!**  
Это **лучший справочник по технологии**, который можно найти.

- <https://docs.oracle.com/javase/8/docs/>
- <https://docs.oracle.com/javase/tutorial/>
- <https://docs.oracle.com/en/java/javase/12/>





После окончания данного курса  
у Вас есть **высокие шансы получить работу.**

# Лекция №1. Введение в Java

## Сравним Java и C++:

1. Отсутствие указателей
2. Отсутствует множественное наследование
3. Heap & Stack
4. JVM (Java Virtual Machine)
5. Работа с памятью

Также Java не поддерживает:

- Перегрузку операторов (operator overloading)
- Беззнаковые целые числа

В Java существуют **конструкторы**, но отсутствуют **деструкторы**.



# Что такое JVM (JAVA VIRTUAL MACHINE)?

**Java Virtual Machine (JVM)** это основная часть исполняющей системы **Java Runtime Environment (JRE)**.

**JVM** является **ключевым компонентом** платформы Java.

Виртуальная машина Java исполняет **байт-код**, предварительно созданный компилятором из исходного текста Java-программы.

Т.к. **JVM** доступна для большинства аппаратных и программных платформ, **JVM** может также использоваться для выполнения программ, написанных на других языках программирования.

# JVM (JAVA VIRTUAL MACHINE)

Виртуальная машина Java определяет:

- Набор инструкций для виртуального CPU;
- Набор регистров процессора;
- Формат файла с байт-кодом (class-файла);
- Формат стека;
- Механизм сборки мусора;
- Способ организации памяти.

**JVM** может быть реализована как программно, так и аппаратно.

# Heap и Stack

В Java все объекты программы расположены в **динамической памяти - куче данных (heap)** и доступны по объектным ссылкам, которые, в свою очередь, хранятся **в стеке (stack)**.

Всегда, когда объявляются новые переменные, создаются объекты или вызываются новые методы, **JVM** выделяет память для этих операций в **стеке** или в **куче**.

Это решение **исключило непосредственный доступ к памяти**, но усложнило работу с элементами массивов и сделало ее менее эффективной по сравнению с программами на C++.

# Stack

**Stack** работает по схеме **LIFO** (последним вошел, первым вышел).

- Когда **вызывается новый метод**, содержащий примитивные значения или ссылки на объекты, то на вершине стека под них **выделяется блок памяти**.
- Когда **метод завершает выполнение**, то **блок памяти очищается**, и пространство становится доступным для следующего метода.

## Ключевые особенности:

- Стек заполняется и освобождается по мере вызова и завершения новых методов;
- Переменные в стеке существуют до тех пор, пока выполняется метод, в котором они были созданы;
- Если память стека будет заполнена, Java бросит исключение **java.lang.StackOverflowError**;
- Доступ к **стеку** осуществляется быстрее, чем к **куче**
- **Стек – потокобезопасный**, т.к. для каждого потока создается отдельный стек.

# Heap

**Heap** используется для **объектов** и **классов**.

Новые объекты **всегда создаются в куче**, а **ссылки на них хранятся в стеке**.

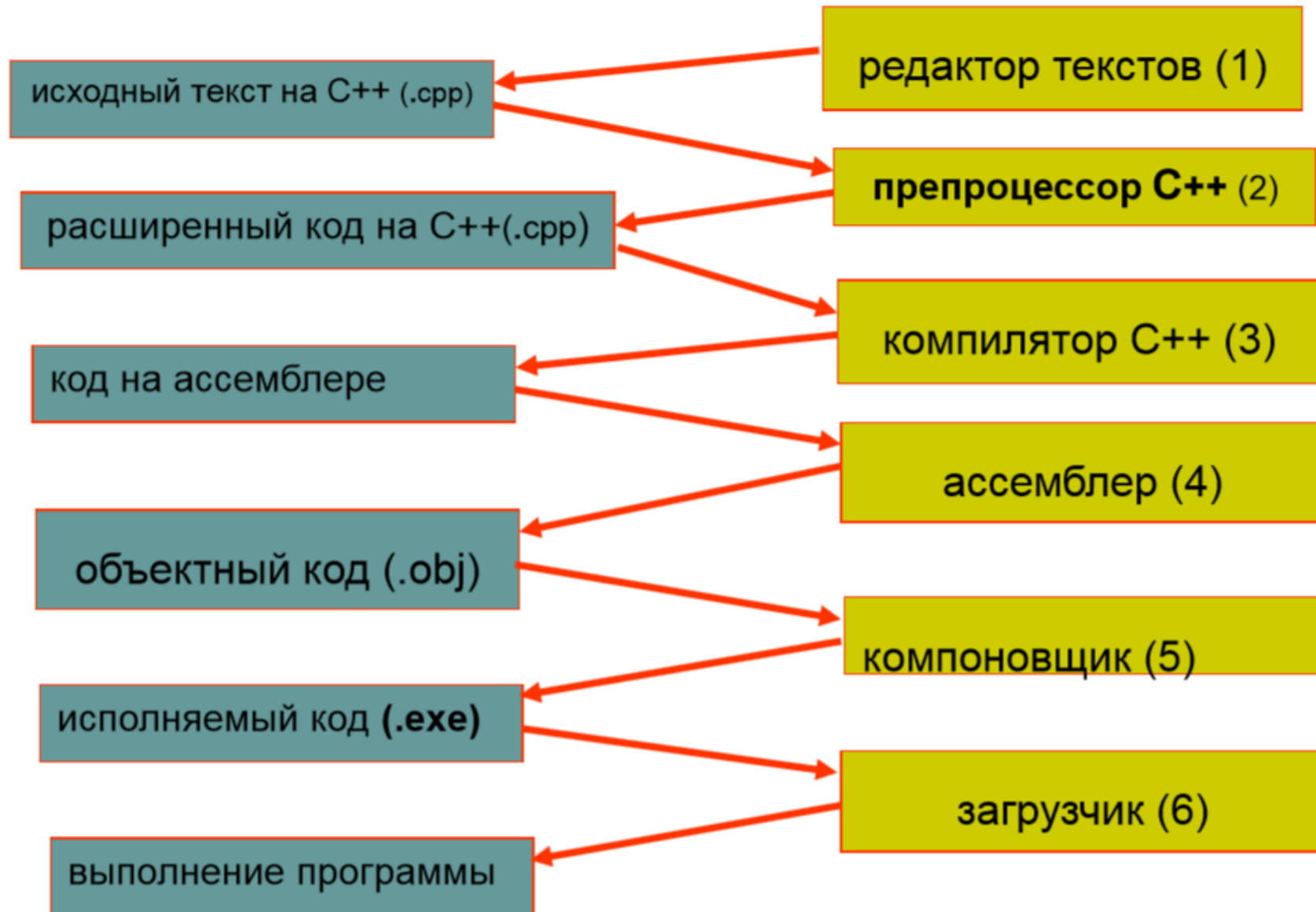
Эти объекты имеют **глобальный доступ** и могут быть получены **из любого места программы**.

## Ключевые особенности:

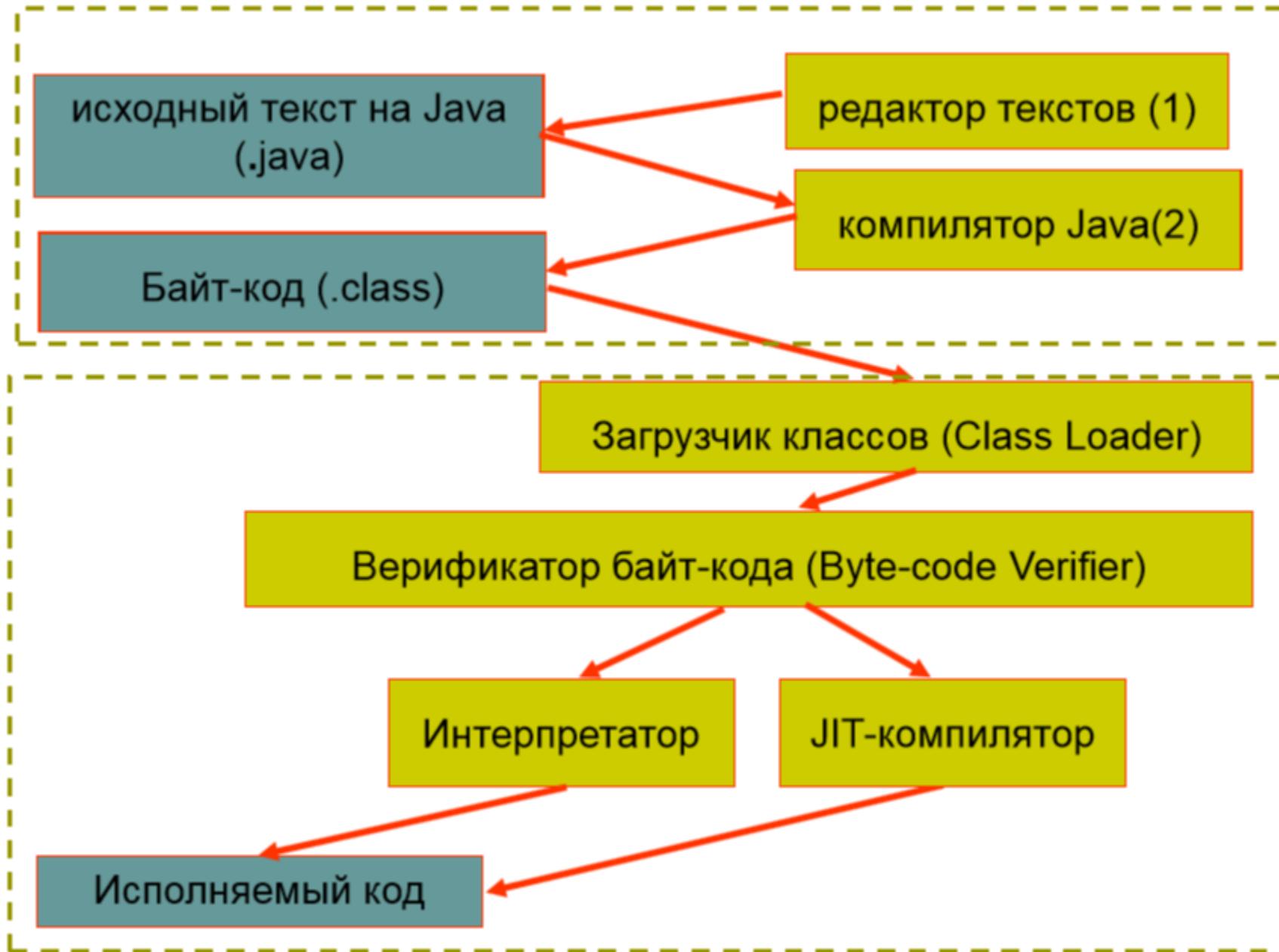
- Когда память кучи заполняется, Java бросает **`java.lang.OutOfMemoryError`**;
- Доступ к **куче** значительно медленнее, чем к **стеку**;
- **Куча**, в отличие от стека, **автоматически не освобождается**. Для сбора неиспользуемых объектов используется **сборщик мусора**.
- В отличие от стека, **куча** не является **потокобезопасной**, поэтому необходимо правильно синхронизировать код для ее контроля.

**JAVA DEVELOPERS NEVER RIP,  
THEY JUST GET GARBAGE COLLECTED.**

# Этапы написания, компиляции и исполнения программы в C++



## Этапы написания, компиляции и исполнения программы в Java





# Особенности платформы Java

## 1. Платформенная независимость

Программы Java переносимы, как на уровне исходных кодов, так и на уровне скомпилированных исполняемых кодов – байт-кода.

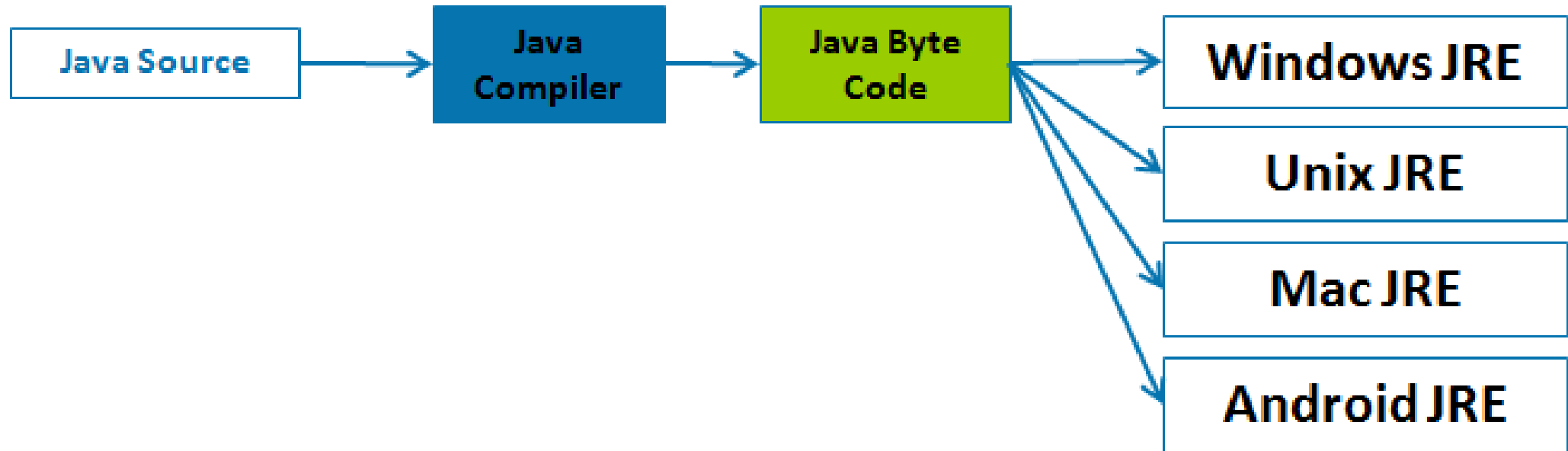
Байт-код является платформонезависимым, так как не содержит инструкций процессора конкретного компьютера.

Он интерпретируется виртуальной **Java-машиной** (Java Virtual Machine – JVM).

Приложения Java могут выполняться на любой платформе, где имеется среда выполнения Java (**Java Runtime Environment – JRE**), содержащая JVM.

# Особенности платформы Java

## 1. Платформенная независимость



# Особенности платформы Java

## 2. Ориентированность на работу с объектами

Java-приложение представляет собой некоторую композицию классов, каждый из которых обычно определен в отдельном файле (в файл включаются и определение, и реализация класса) исходного кода.

## 3. Автоматическое управление памятью

Программисту в Java никогда не приходится уничтожать объект или освобождать память.

Когда всем переменным, хранящим ссылки на объект, присваиваются другие значения или когда они выходят из области определения, объект помечается виртуальной машиной Java как неиспользуемый, и система асинхронно ликвидирует все такие объекты.

Этот процесс называется сбором мусора.

# Особенности платформы Java

## 4. Надежность

Среда Java автоматически проверяет, что исполняемый код:

- соответствует спецификациям платформы,
- не нарушает целостность системы,
- не приводит к переполнению стека,
- аргументы методов имеют правильный тип,
- не происходит недопустимых преобразований типов (например, целое к ссылке).

## 5. Защищенность

Приложения Java выполняются в ограниченном окружении и перечень доступных операций задается с помощью политики безопасности.

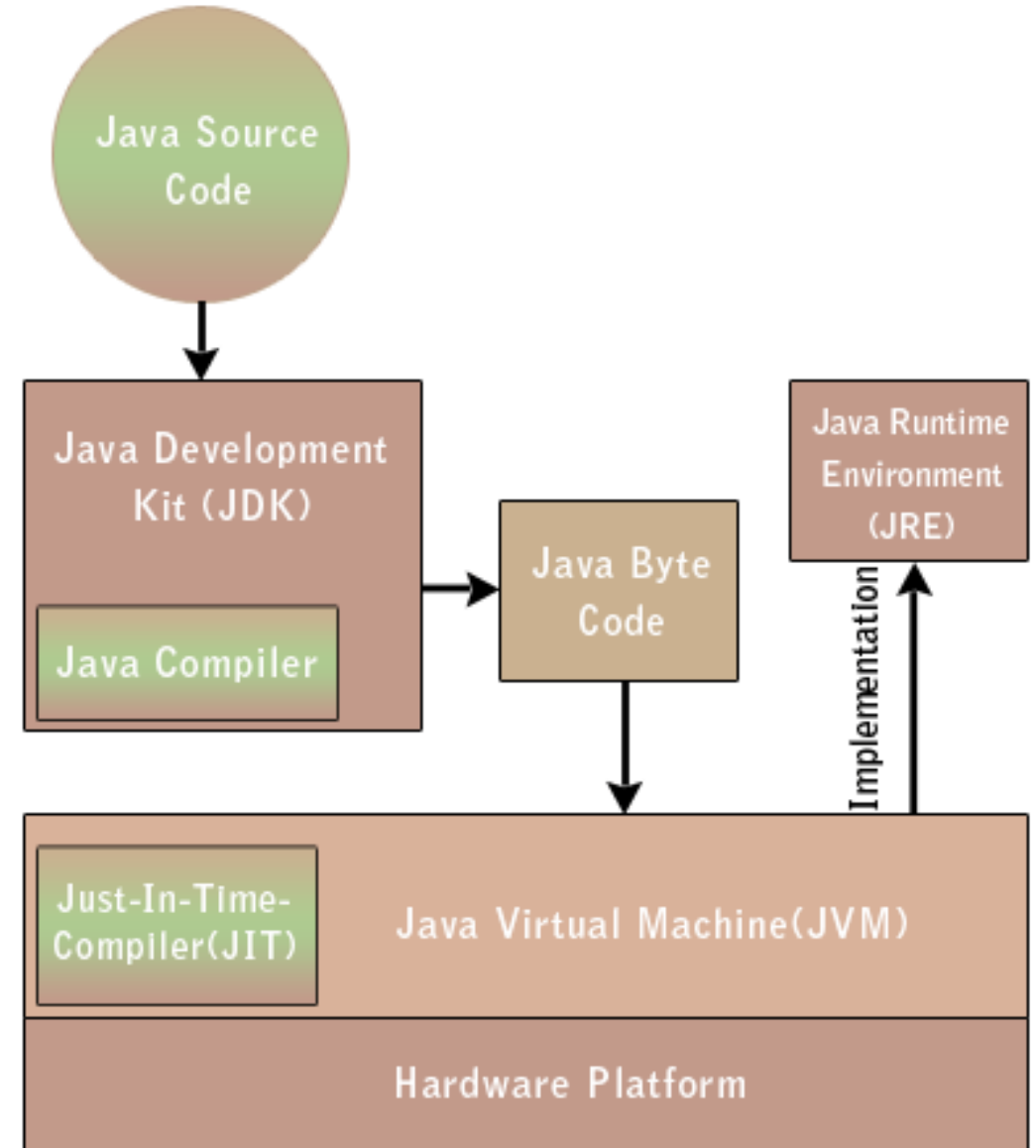
# Особенности платформы Java

## 6. Производительность

Интерпретация байт-кода – медленная операция и является узким местом системы.

Для повышения скорости работы приложений применяется JIT-компиляция (Just-In-Time) – компиляция байт-кода в код конкретной платформы в момент выполнения программы.

Она позволяет ускорить работу программ за счет замены интерпретации байт-кода на выполнение скомпилированного кода.



# Особенности платформы Java

## 7. Многопоточность

Среда Java изначально разрабатывалась для создания многопоточных приложений, поэтому обладает средствами управления и синхронизации параллельно работающих потоков.

## 8. Встроенная поддержка сетевых взаимодействий

Среда Java разрабатывалась для создания сетевых приложений, поэтому программирование сетевых взаимодействий существенно облегчено и для этого предложен целый ряд встроенных средств.

# Структура платформы Java

JSE (Standard)



JME (Micro)



JEE (Enterprise)

# Структура платформы Java

- **Java Platform Standard Edition (Java SE)** – данный тип является базовым и необходим как для разработки, так и для запуска Java приложений. Он включает библиотеки API, компилятор байт-кода, виртуальную машину Java (JVM) с интерпретатором, а также ряд других утилит.
- **Java Runtime Environment, Standard Edition (Java RE)** – является урезанной версией JSE. Не содержит компилятора и других утилит, но по-прежнему включает библиотеки API и интерпретатор. Данная версия платформы может быть установлена на систему, предназначенную только для исполнения Java-приложений.
- **Java Platform, Enterprise Edition (Java EE)** – является одновременно и платформой и для упрощения создания высоко масштабируемых промышленных приложений. J2EE требует для своей работы как JSE, так и различные программные продукты сторонних разработчиков, обеспечивающих среду выполнения.
- **Java Platform, Micro Edition (Java ME)** – является уменьшенной и оптимизированной версией JSE, предназначенной для создания приложений, способных выполняться на изделиях бытовой электроники, таких как смарт-карты, сотовые телефоны и КПК.



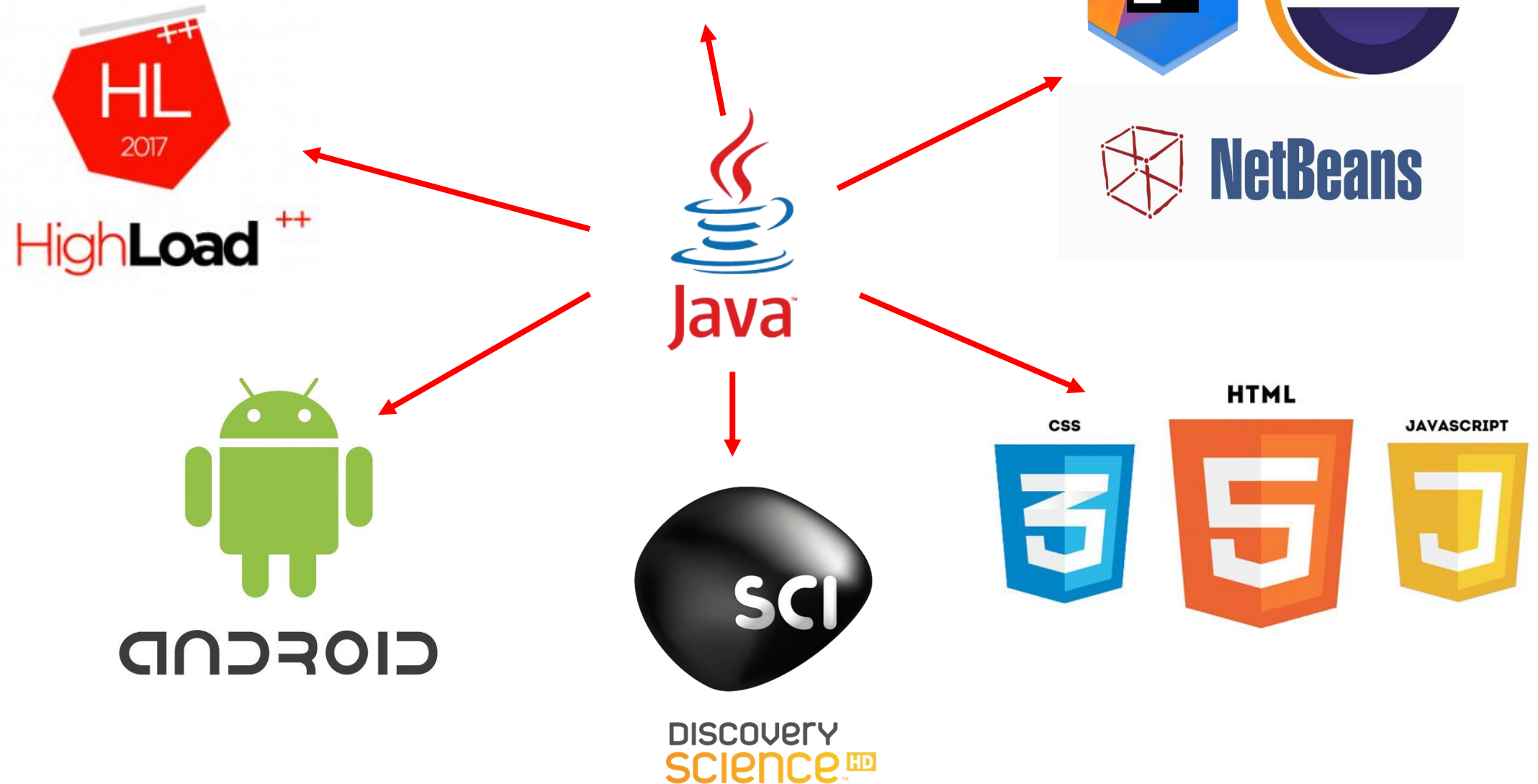
# Категории программ Java

Существуют следующие типы платформы Java:

- [Приложение \(Application\)](#) – аналог “обычной” прикладной программы, которая может быть запущена в операционной системе.
- [Апплет \(Applet\)](#) – специализированная программа, работающая в окне браузера и запускаемая из веб-документа.
- [Сервлет \(Servlet\)](#) – специализированная программа, работающая внутри специализированного программного контейнера (HTTP сервера).
- [Мидлет \(Midlet\)](#) – специализированная программа, работающая на мобильном устройстве.
- [Компонент EJB \(Enterprise JavaBean\)](#) – самостоятельный компонент, реализующий определённые функции, предназначен для многократного использования серверными приложениями Java, работает внутри специализированного программного контейнера (сервера приложений).
- [Библиотека](#) – предназначена для многократного использования программами классов Java.

# GLOBALTRADE

Market Research | Trade Consulting | International Business



# Выбор среды разработки


- IntelliJ Idea
- NetBeans (от Oracle)
- Eclipse
- Android Studio
- JDeveloper
- DrJava
- MyEclipse
- JEdit



**NetBeans**



# Сертификация OSCP



**ORACLE®**  
Certified Professional

CERTIFICATE OF RECOGNITION

**Aleksandr Nesterov**  
Oracle Certified Professional, Java SE 8 Programmer

THIS CERTIFIES THAT ABOVE NAMED IS  
RECOGNIZED BY ORACLE CORPORATION AS AN  
**ORACLE CERTIFIED PROFESSIONAL**

June 17, 2016  
DATE


*D. Carey*  
DAMIEN CAREY  
SENIOR VICE PRESIDENT, ORACLE UNIVERSITY

Java SE - Certification Path

Associate Level Certification Professional Level Certification Master Level Certification **Upgrades**

**Proudly Wear Your Oracle Certification  
You've Earned It!**


Get a FREE Oracle Certification T-shirt  
with any NEW Java Certification(s) granted.



OCJP or SCJP (any version)  
P or SCJP (any version)  
versions)

Exam Preparation (optional)

- [Java SE 8 New Features](#)



**ORACLE®**  
**Certified Professional**

# Полезные ресурсы

- Узнать больше о **высоконагруженных системах** вы можете в блоге **Олега Бунина Highload++**:  
<https://www.highload.ru/>
- Портал **Android Developers** от компании **Google**:  
<https://developer.android.com/>
- Отличные **базовые уроки по Java**, а также по колоссальному стеку **веб-технологий** можно найти тут:  
<https://www.w3schools.com/java/default.asp>





Java.  
Туда и обратно...



# Увидимся на следующей лекции!

На следующей лекции мы поговорим **об объектных возможностях Java, наследовании, интерфейсах и некоторых стандартных классах.**