



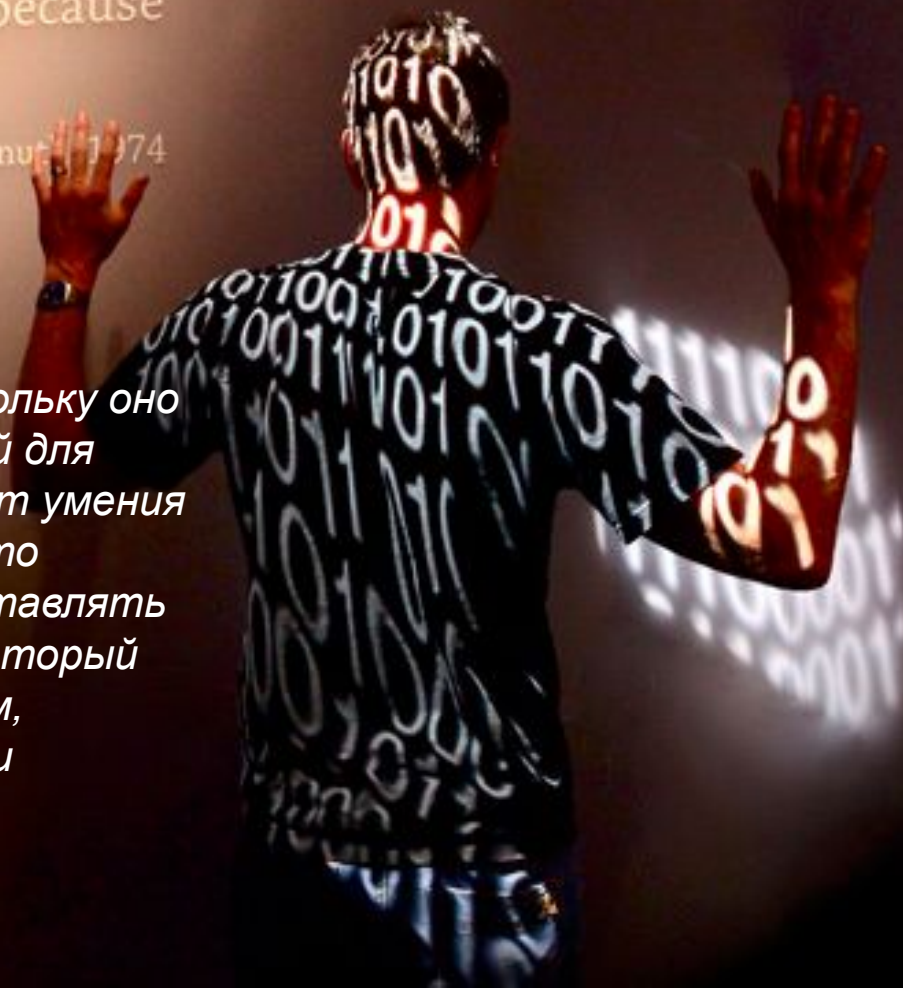
Java™

"Computer programming is an art, because it applies accumulated knowledge to the world, because it requires skill and ingenuity, and especially because it produces objects of beauty."

Donald Knuth, 1974

Программирование — это искусство, поскольку оно является применением накопленных знаний для практических целей, поскольку оно требует умения и мастерства, и в особенности потому, что продукты программирования могут представлять эстетическую ценность. Программист, который бессознательно ощущает себя художником, получает удовольствие от своей работы и справляется с ней лучше.

Дональд Кнут



Программирование — это...

- это искусство.
- это процесс написания программ на языке понятном компьютеру.
- это не сложно.

Язык программирования

- инструмент программиста для создания программ.
- писать программу в виде чисел очень сложно, поэтому люди придумали языки программирования и компиляторы.

Компилятор – это специальная программа, которая переводит текст программы, написанный на языке программирования, в набор машинных кодов.

Почему именно Java ?

- 1 Java — один из самых доступных языков для изучения.
 2. Большой спрос на рынке труда. И самые высокие зарплаты по отрасли.
 4. Миллионы готовых модулей и решений.
 5. Java это кроссплатформенность.
 6. Изучив язык вы с легкостью сможете овладеть другими языками программирования.
- Чем больше вы будите программировать на Java, тем больше будите находить ответов на вопрос – «Почему именно Java?»

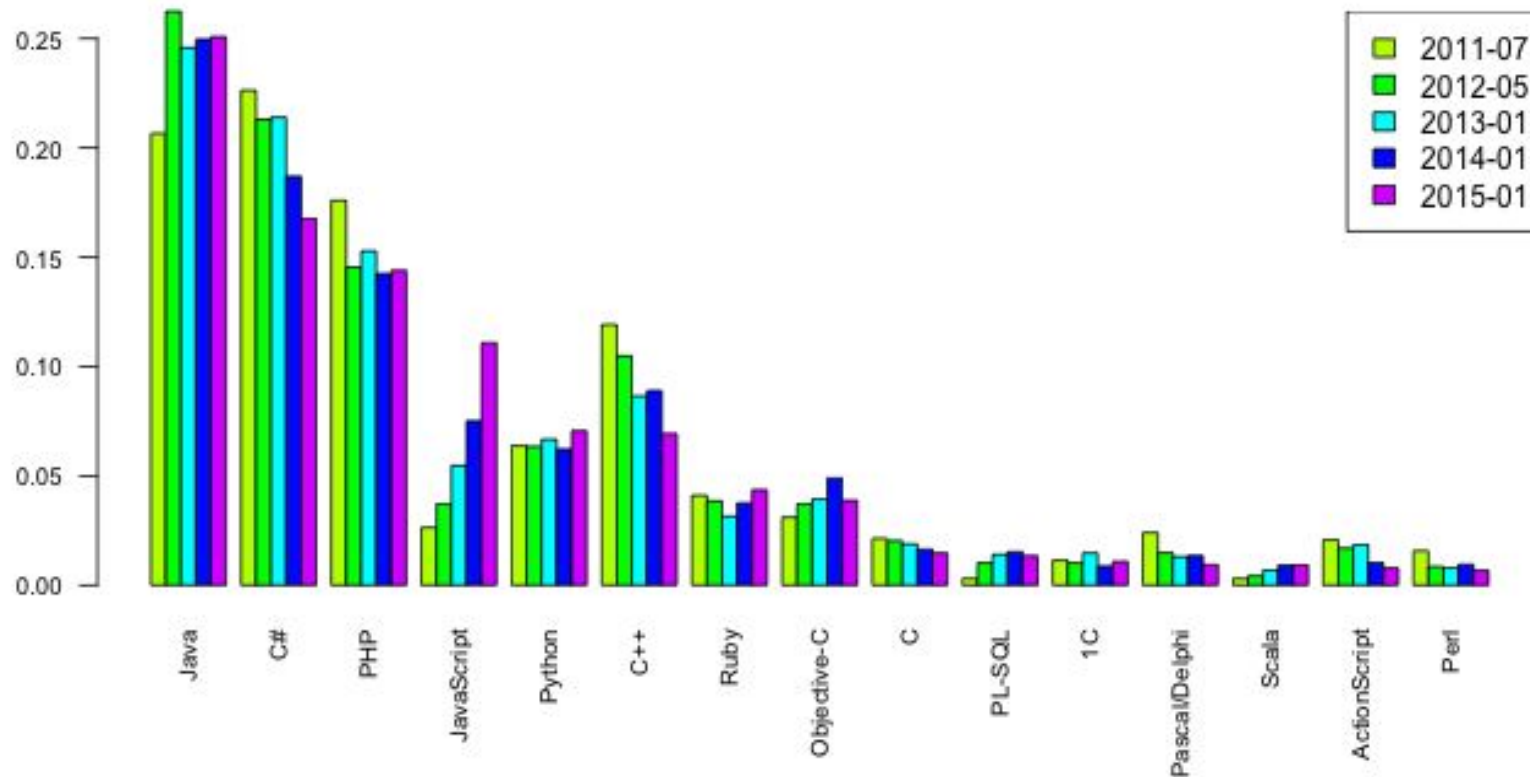
Где используется Java

- 1) Enterprise: тяжелые серверные приложения для банков, корпораций, инвестфондов и т.д.
- 2) Mobile: мобильная разработка (телефоны, планшеты), благодаря Android.
- 3) Web: лидирует PHP, но и Java держит солидный кусок рынка.
- 4) Big Data: распределенные вычисления в кластерах из тысяч серверов.
- 5) Smart Devices: программы для умного дома, электроники, холодильников с выходом в интернет

Самые популярные языки программирования (в Украине 😊)

<http://dou.ua/lenta/articles/language-rating-jan-2015/>

На каком языке вы пишете для работы сейчас



В рамках курса будут рассмотрены следующие темы:

1. "Основные понятия" java: Введение в платформу Java. JVM, JRE, JDK. Компиляция и запуск из консоли. Classpath. Eclipse IDE.
2. "Процедурная" java: циклы, локальные переменные, вызовы методов, рекурсия, ссылки. Все то, что позволяет непосредственно делать полезную работу, вычисления.
3. Объективно-ориентированная java: классы, интерфейсы, наследование, полиморфизм, инкапсуляция, области видимости, переопределение. Все то, что делает Java объектно-ориентированным языком.
4. Исключения в java: базовая иерархия (Throwable, Error, Exception, RuntimeException) checked/unchecked исключения, stack trace, обработка исключений (try, catch, finally, throw, throws).
5. Java I/O: механизмы Java ориентированные на работу с потоками ввода/вывода
6. Коллекции в java: архитектура пакета коллекций (Collection, List, Set, Map, SortedSet, SortedMap, Iterable, Iterator, Comparable, Comparator).
7. Потоки в java: Thread, Runnable, synchronized, volatile, wait()/notify()/notifyAll(). Все те механизмы, которые позволяют создавать многопоточные программы, использующие процессор "на все 100".

Технологии Java

Внутри Java существуют три основных семейства технологий:

Java EE — Java Enterprise Edition, для создания программного обеспечения уровня предприятия;

Java SE — Java Standard Edition, для создания пользовательских приложений, в первую очередь — для настольных систем;

Java ME или Java Micro Edition, для использования в устройствах, ограниченных по вычислительной мощности.

Java SE

В составе Java SE содержится два основных продукта: Java Runtime Environment (JRE) и Java Development Kit (JDK). JRE представляет собой библиотеки, виртуальную Java-машину (JVM) и другие компоненты для исполнения приложений, разработанных на языке Java.

JDK включает в себя все, что содержится в JRE, а также компиляторы языка Java, отладчик и дополнительные библиотеки..

Java Virtual Machine (сокращенно Java VM, JVM) — виртуальная машина Java — основная часть исполняющей системы Java.

Виртуальная машина Java **интерпретирует** Байт-код Java, предварительно созданный из исходного текста Java-программы компилятором Java (javac).

JIT-компилятор (JIT – Just In Time). Этот механизм позволяет преобразовать байт-код в машинный код при первом проходе программы и в дальнейшем выполнять просто готовый машинный код.

Установка Java SE 8

Для того, чтобы начать программировать на языке Java, первоначально нужно установить программную среду на свой компьютер. Для этого необходимо скачать установочный файл с сайта. На сегодня местонахождения файлов для скачивания JDK находится по адресу

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

В таблице Java SE Development Kit 8u111 отмечаем согласие с условиями использования, выбрав пункт Accept License Agreement

Technology, and subscribe to Java.

- [Java Developer Day hands-on workshops \(free\) and other events](#)
- [Java Magazine](#)

JDK 8u111 Checksum
JDK 8u112 Checksum

Java SE Development Kit 8u111

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Accept License Agreement Decline License Agreement

File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.78 MB	jdk-8u111-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.73 MB	jdk-8u111-linux-arm64-vfp-hflt.tar.gz
Linux x86	160.35 MB	jdk-8u111-linux-i586.rpm
Linux x86	175.04 MB	jdk-8u111-linux-i586.tar.gz
Linux x64	158.35 MB	jdk-8u111-linux-x64.rpm
Linux x64	173.04 MB	jdk-8u111-linux-x64.tar.gz
Mac OS X	227.39 MB	jdk-8u111-macosx-x64.dmg
Solaris SPARC 64-bit	131.92 MB	jdk-8u111-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	93.02 MB	jdk-8u111-solaris-sparcv9.tar.gz
Solaris x64	140.38 MB	jdk-8u111-solaris-x64.tar.Z
Solaris x64	96.82 MB	jdk-8u111-solaris-x64.tar.gz
Windows x86	189.22 MB	jdk-8u111-windows-i586.exe
Windows x64	194.64 MB	jdk-8u111-windows-x64.exe

Дальнейшая установка будет рассмотрена на примере Windows. Если вы не знаете тип вашей Windows (x86 или x64) откройте Панель управления -> Система там будет указан Тип системы. Если тип вашей системы — 32 разрядная, то нужно устанавливать версию для x86, если 64 разрядная, то x64.

Для того, чтобы установить Java на компьютер, запустите скаченный файл, после чего начнется процесс установки



Настройка переменных сред.

На следующем шаге необходимо задать пути к выполняемым файлам, а именно задать системные переменные PATH, CLASSPATH и JAVA_HOME .

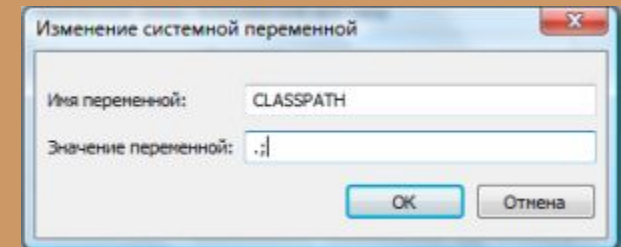
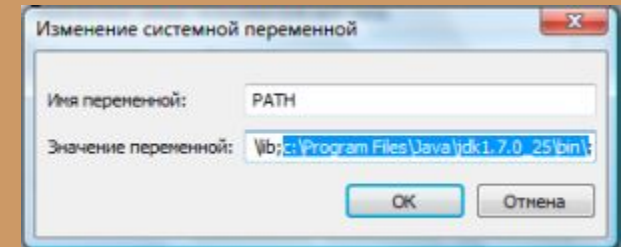
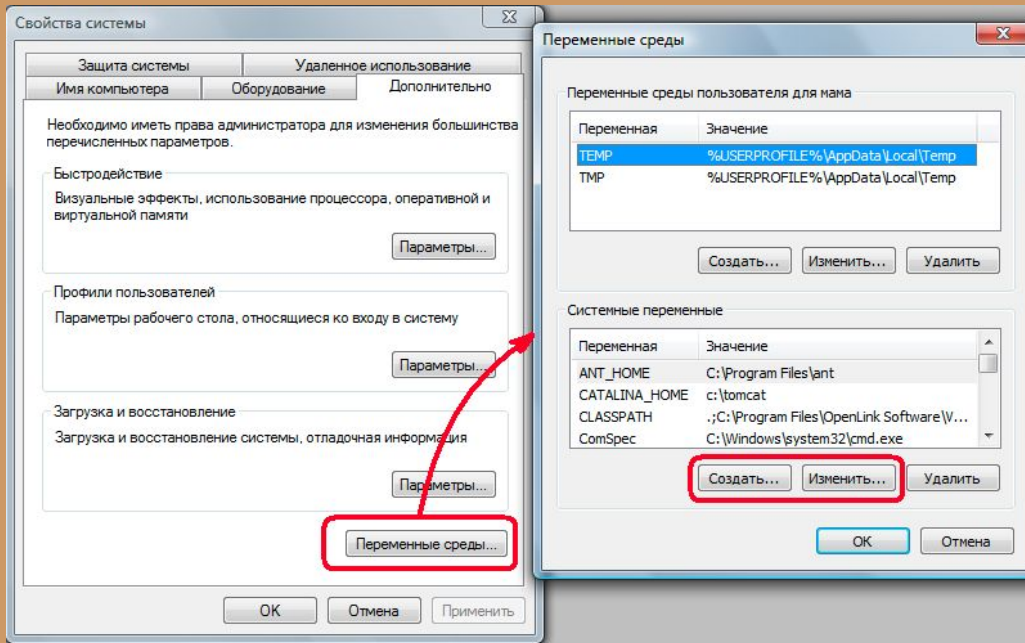
Для этого откройте Панель управления -> Система, выберете Дополнительные параметры системы. В появившемся окне нажмите кнопку Переменные среды.

В открывшемся окне в блоке Системные переменные нужно проверить существует ли переменная PATH, если переменная не существует, то ее нужно создать нажав на кнопку Создать, если переменная PATH уже существует, то нужно установить Значение переменной — путь к каталогу, в который была установлена Java.

Если переменной PATH уже присвоены другие значения, то новое значение добавляется через точку с запятой «;» в самом конце строки.

Также необходимо создать переменную CLASSPATH, если она еще не создана. В качестве ее значения необходимо указать «.»

Подобным образом создается/изменяется переменная JAVA_HOME. Значение этой переменной — путь к каталогу, в который была установлена Java (как для PATH), но уже без поддиректории bin.



Пример:

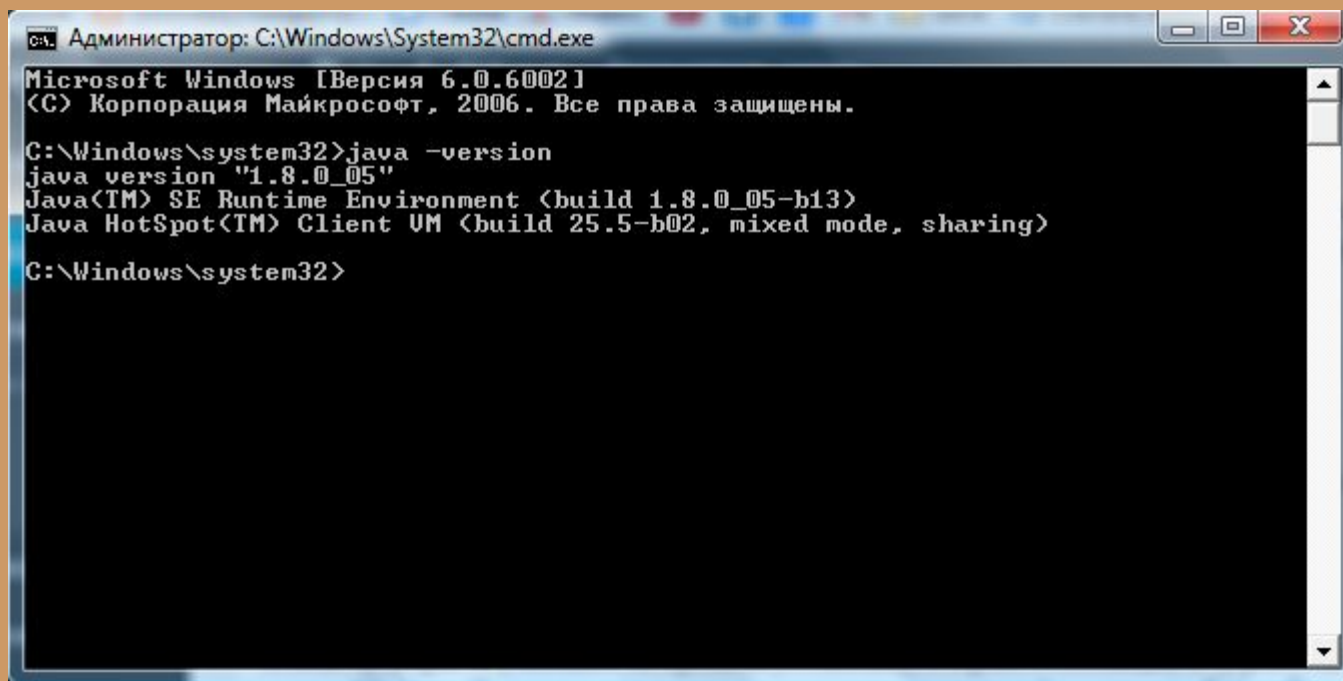
Значение в PATH: C:\Program Files\Java\jdk1.7.0_25\bin

Значение в JAVA_HOME: C:\Program Files\Java\jdk1.7.0_25

После того, как были прописаны переменные, установку Java можно считать завершенной!

Чтобы проверить, успешно ли прошла установка, откройте командную строку. Для этого в меню Windows Пуск в строке поиска введите команду **cmd** и нажмите Enter. После этого откроется командное окно, в котором нужно ввести следующую команду.

java -version



```
Администратор: C:\Windows\System32\cmd.exe
Microsoft Windows [Версия 6.0.6002]
(C) Корпорация Майкрософт, 2006. Все права защищены.

C:\Windows\system32>java -version
java version "1.8.0_05"
Java(TM) SE Runtime Environment (build 1.8.0_05-b13)
Java HotSpot(TM) Client VM (build 25.5-b02, mixed mode, sharing)

C:\Windows\system32>
```



Кратко об установке.

Создание нового проекта.

Создание нового класса.

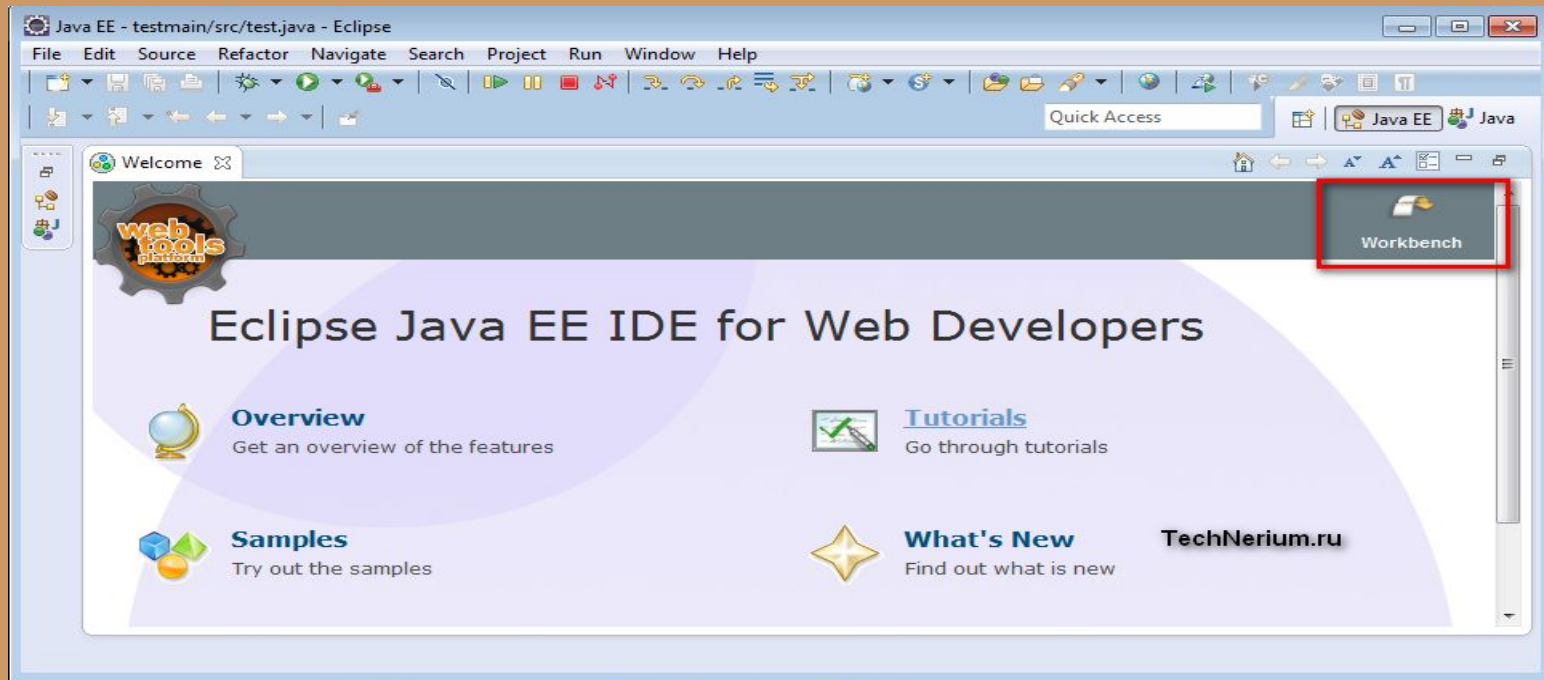
Редактирование кода программы – Hello!

Автоматическое форматирование кода.

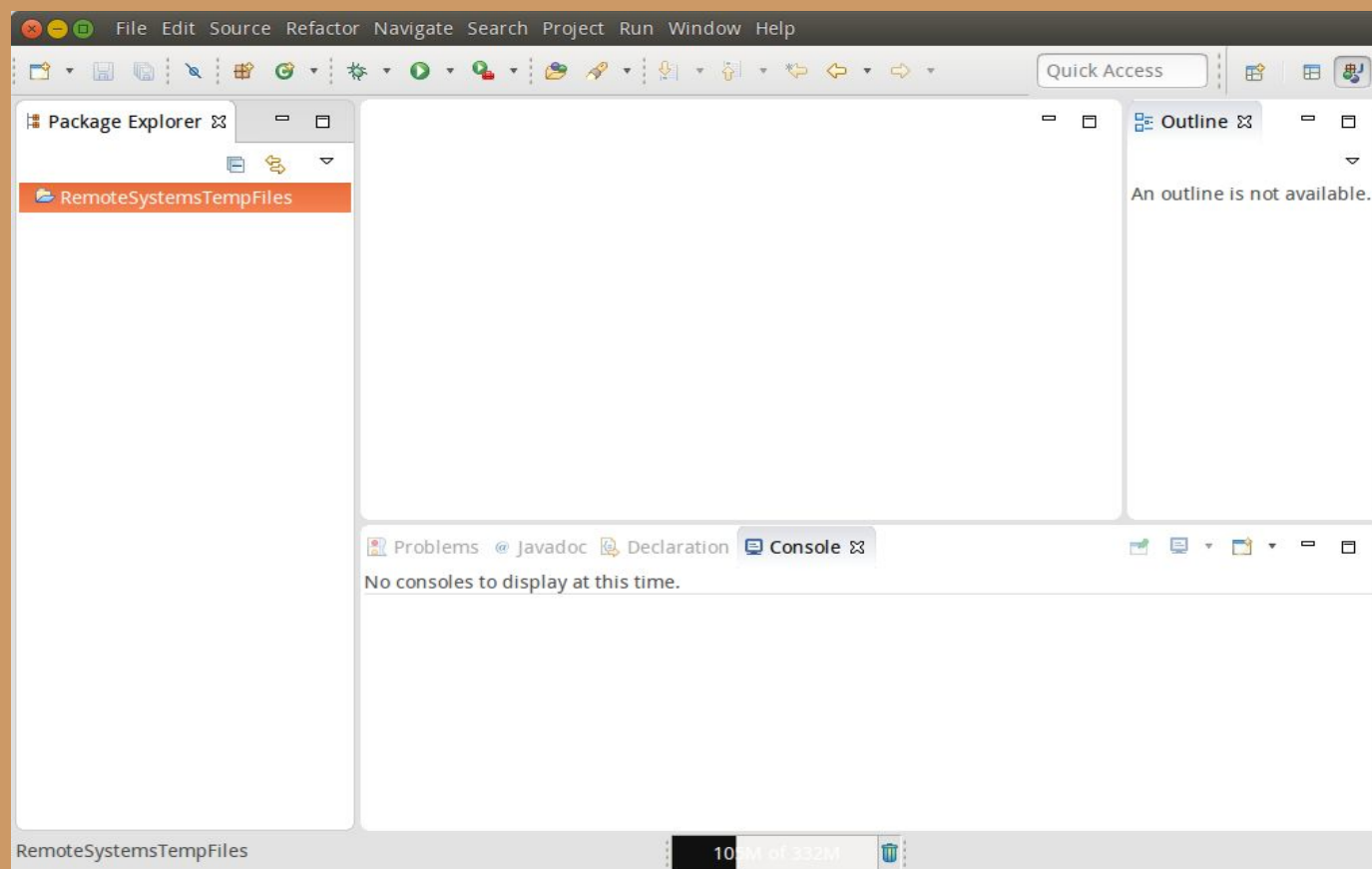
Запуск на выполнение.

Окно “Console”.

1. Открываем страницу [Eclipse Indigo](#). Здесь выбираем ссылку на загрузку в соответствии с используемой операционной системой. Скачиваем архив. Распаковываем в удобное место: Eclipse не устанавливается через инсталлятор, он portable.
2. Откройте папку bin в папке установки Eclipse. Найдите в ней файл eclipse.exe. Запустите его двойным кликом. Можно создать ярлык для этого файла и скопировать его на рабочий стол. После того как Eclipse запустится, кликните по иконке Workbench как показано на картинке ниже.

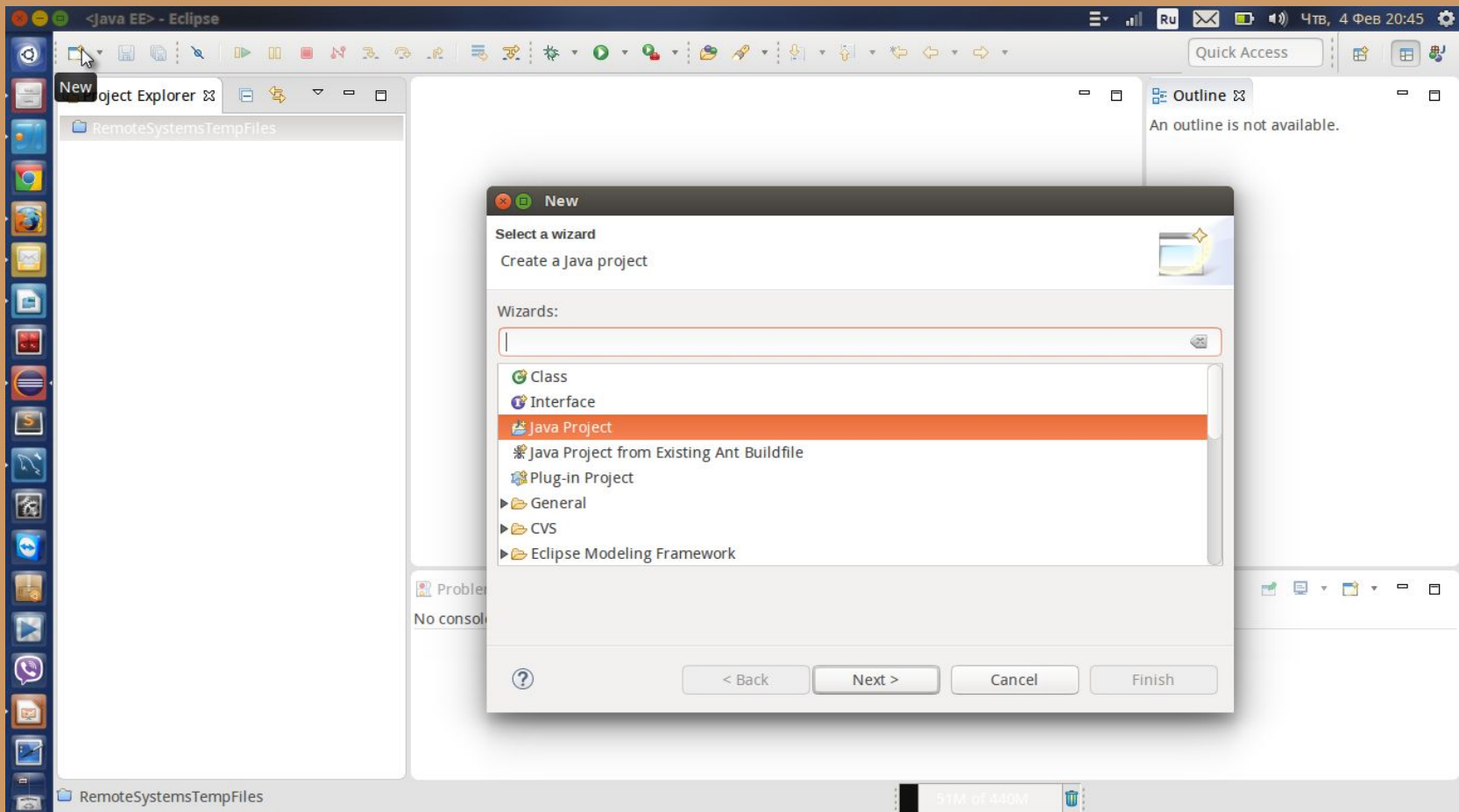


1 шаг. Запустим Eclipse SDK



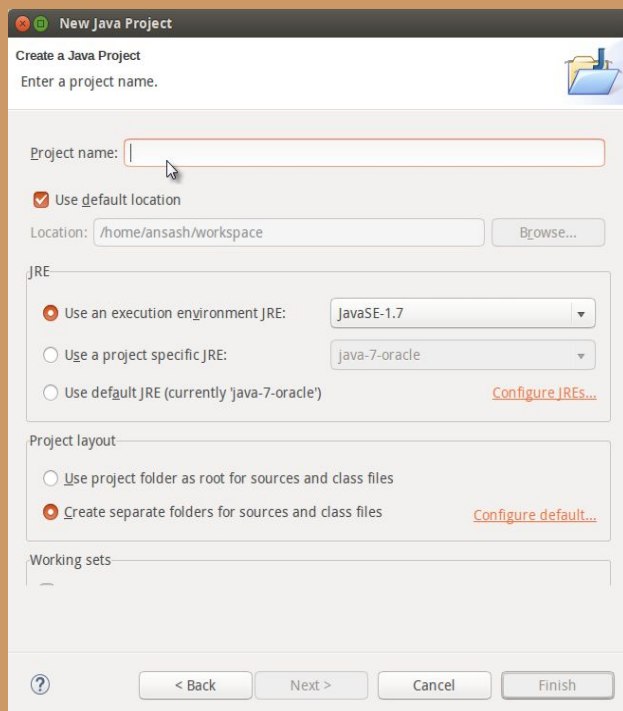
2 шаг. Создание Java проекта

Заходим в главное меню "File" -> "New" -> "Java Project".



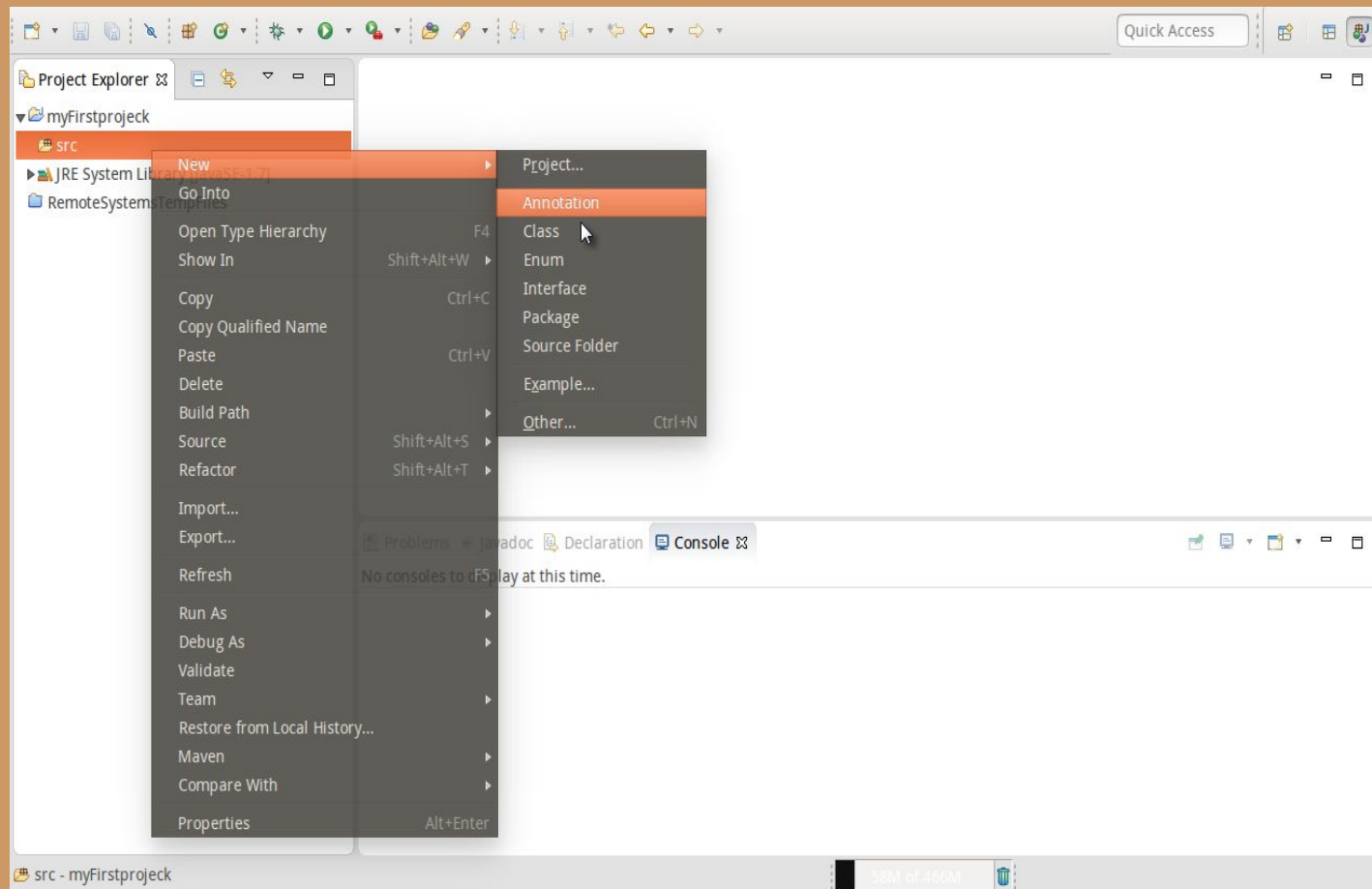
3 шаг. В открывшемся диалоговом окне введем имя проекта. Располагаться проект будет в директории, установленной как *Workspace* при настройке.

Другие установки оставим как есть, при этом для выполнения проекта будет использоваться установленная по умолчанию в системе JRE и исходные файлы будут помещены в отдельную папку "src".



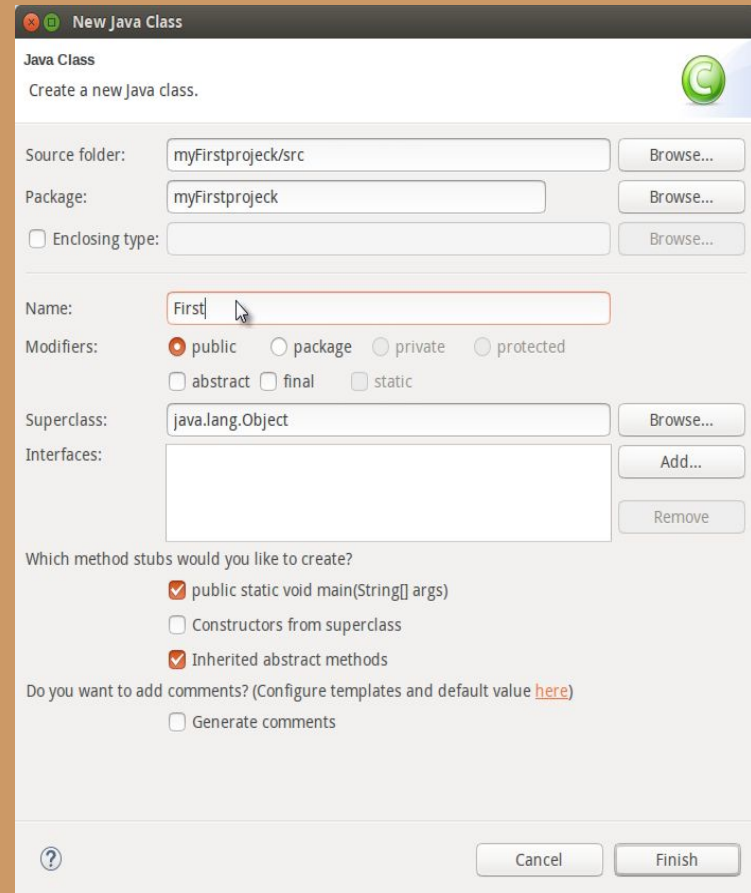
". Будет создан пустой проект.

4 шаг. Создадим класс через контекстное меню. Кликнем правой кнопкой мыши на src и выберем "New" -> "Class". То же можно проделать, нажав кнопку "New Java Class" на панели инструментов.



В диалоговом окне создания класса введём его имя, оно может быть любым, согласно правилам именования классов в Java.

Отметим опцию "public static void main(String[] args)", тем самым укажем IDE создать для нас одноимённую функцию.



New Java Class

Java Class
Create a new Java class.

Source folder: myFirstproject/src Browse...

Package: myFirstproject Browse...

Enclosing type: Browse...

Name: First

Modifiers: public package private protected
 abstract final static

Superclass: java.lang.Object Browse...

Interfaces: Add...
Remove

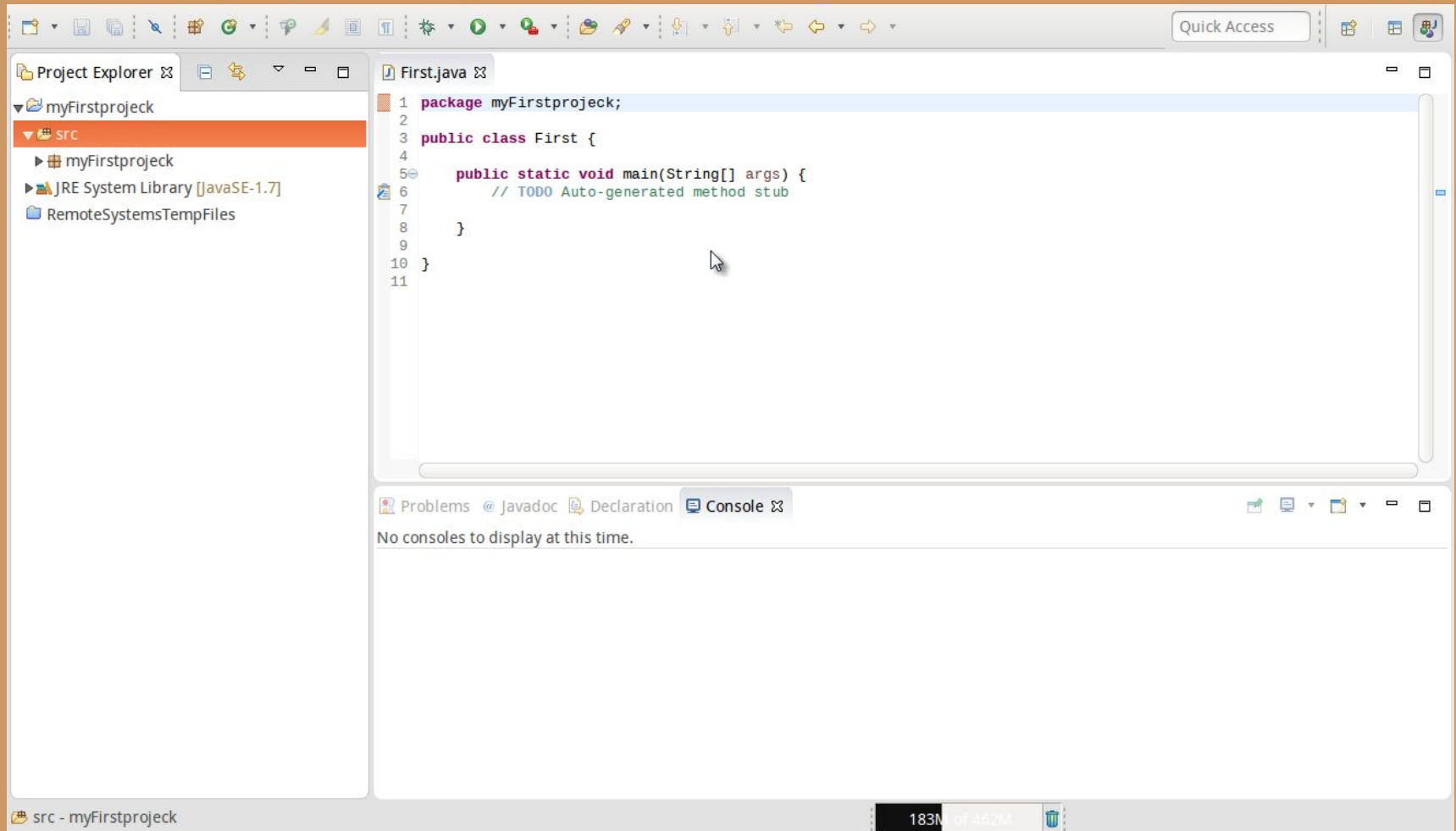
Which method stubs would you like to create?

public static void main(String[] args)
 Constructors from superclass
 Inherited abstract methods

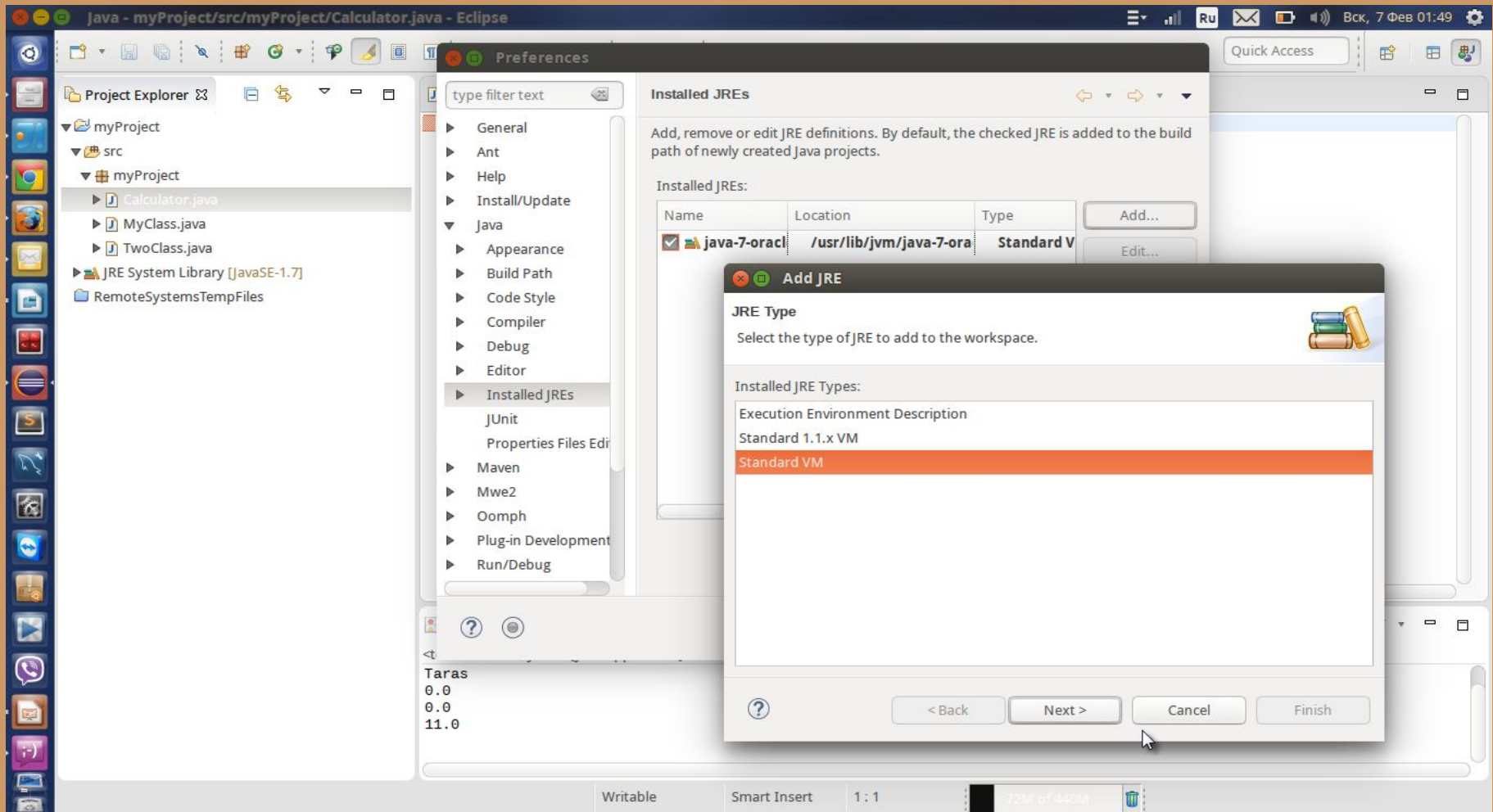
Do you want to add comments? (Configure templates and default value [here](#))
 Generate comments

? Cancel Finish

Теперь справа мы видим структуру проекта и наш файл класса с расширением JAVA. По центру - исходный код класса, а справа - браузер классов, показывающий структуру пакетов и классов в виде дерева. Проект в левой части экрана и должен в себе содержать элемент JRE System Library



Если этого элемента нет, то его необходимо добавить вручную! Для этого выберем Windows -> Preferences, в открывшемся окне Preferences слева выберем Java -> Installed JREs, нажмем кнопку Add... справа. В открывшемся окне выберем Standard VM и нажмем кнопку Next. В открывшемся окне Add JRE, укажем директорию, в которой установлена Java и нажмем Finish.

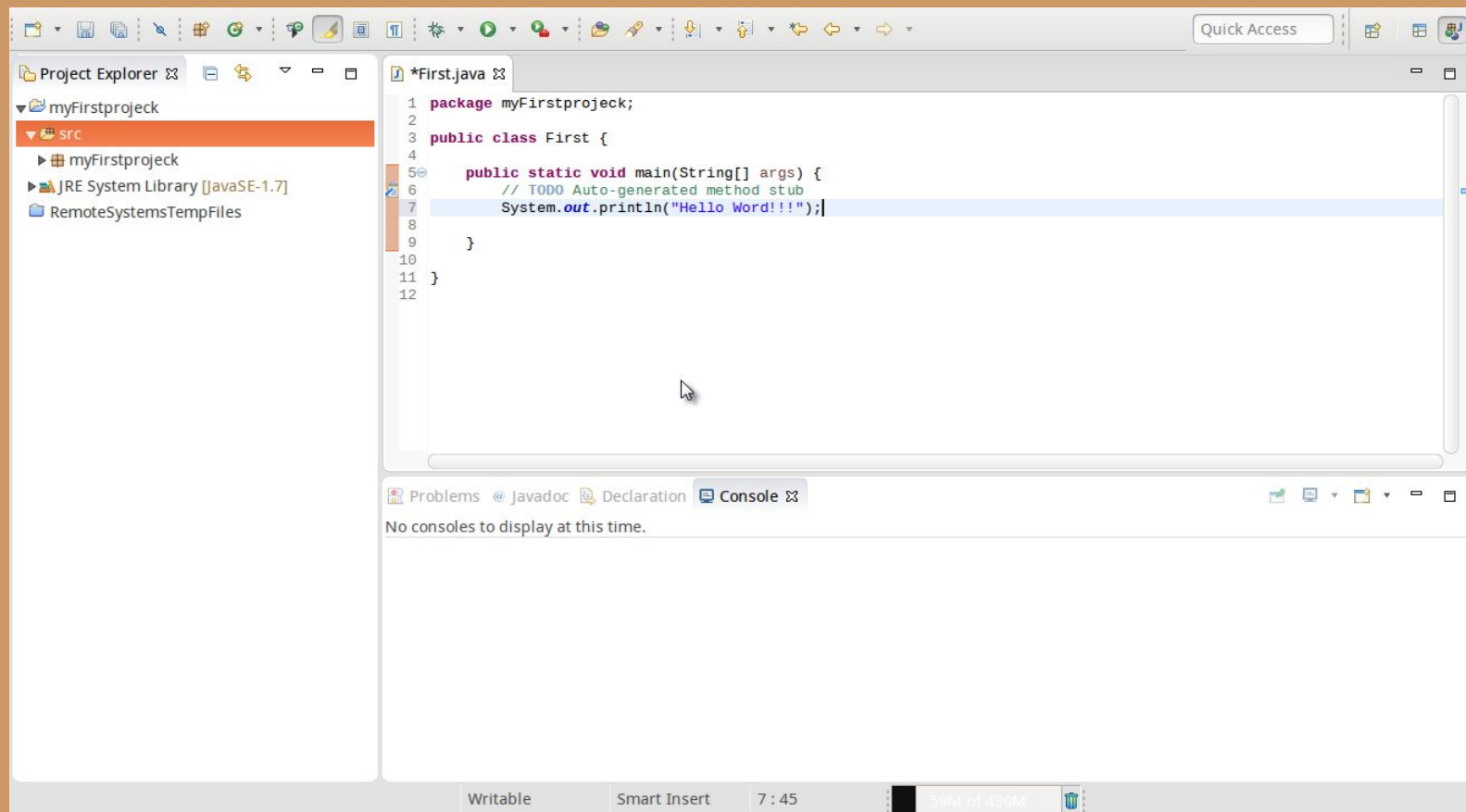


Простейшая программа на Java

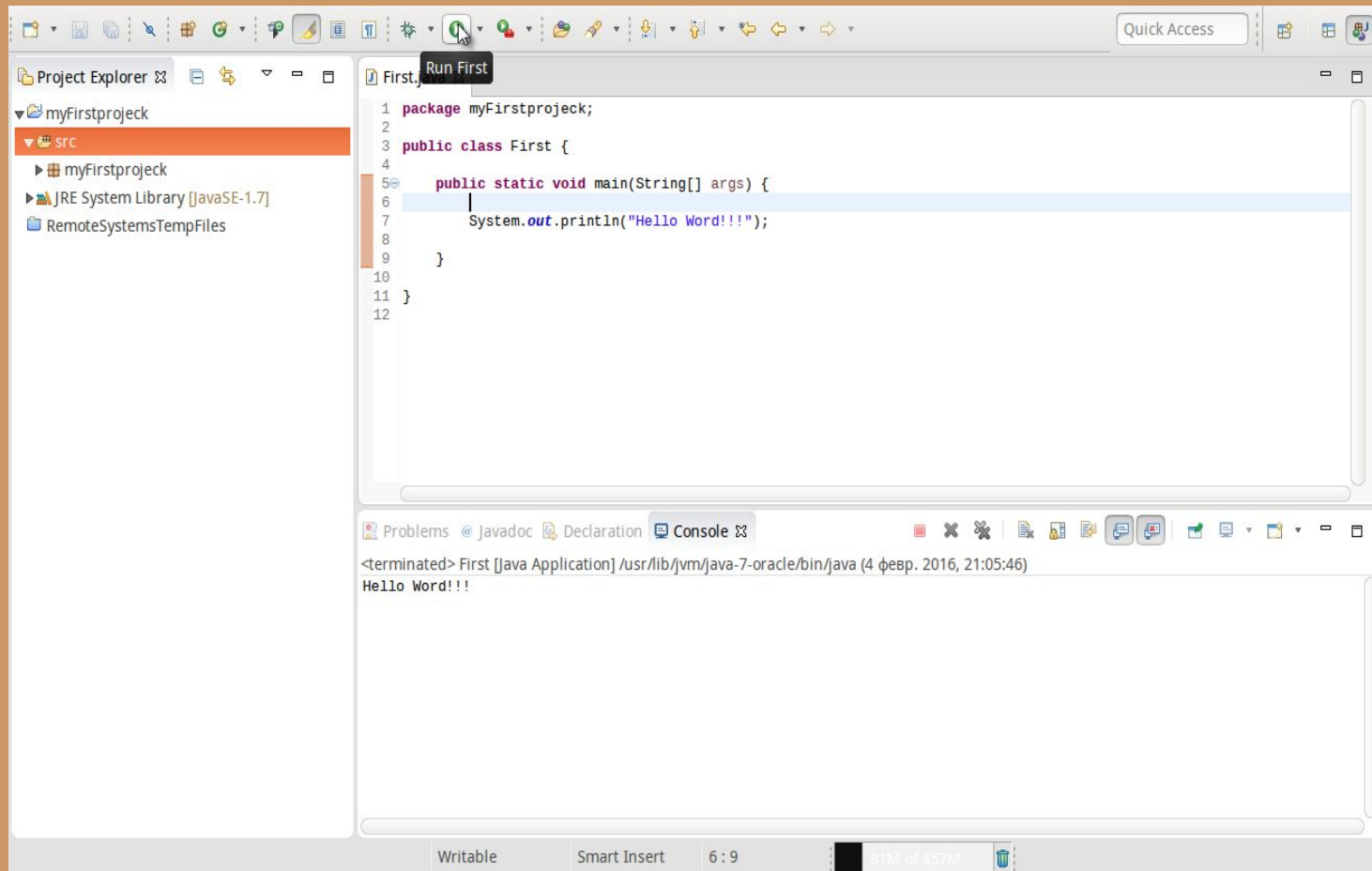
```
public class Main {  
  
    public static void main(String[] args) {  
  
        System.out.println("Hello World");  
    }  
}
```


Запуск Java проекта в Eclipse

Чтобы проверить работоспособность нашей программы, нажмём кнопку "Run" на панели инструментов или через главное меню.



Наше первое консольное Java приложение будет скомпилировано и выполнено. В открывшемся представлении "Console" в нижней панели главного окна IDE мы увидим вывод программы, а именно - нашу строку.



Компиляция исходного файла из консоли

Для того, чтобы скомпилировать исходный файл, открываем командную строку. Для этого в меню Windows Пуск в строке поиска вводим команду cmd и жмем Enter. После этого откроется командное окно.

Теперь в нем нужно изменить текущий каталог на тот, в котором находится наш исходный файл (например C:\studyjava\). Для этого вводим следующую команду (Change Directory):

```
cd C:\studyjava\
```

и нажимаем Enter. После того, как директория изменилась, вводим команду компиляции

```
javac HelloWorld.java
```

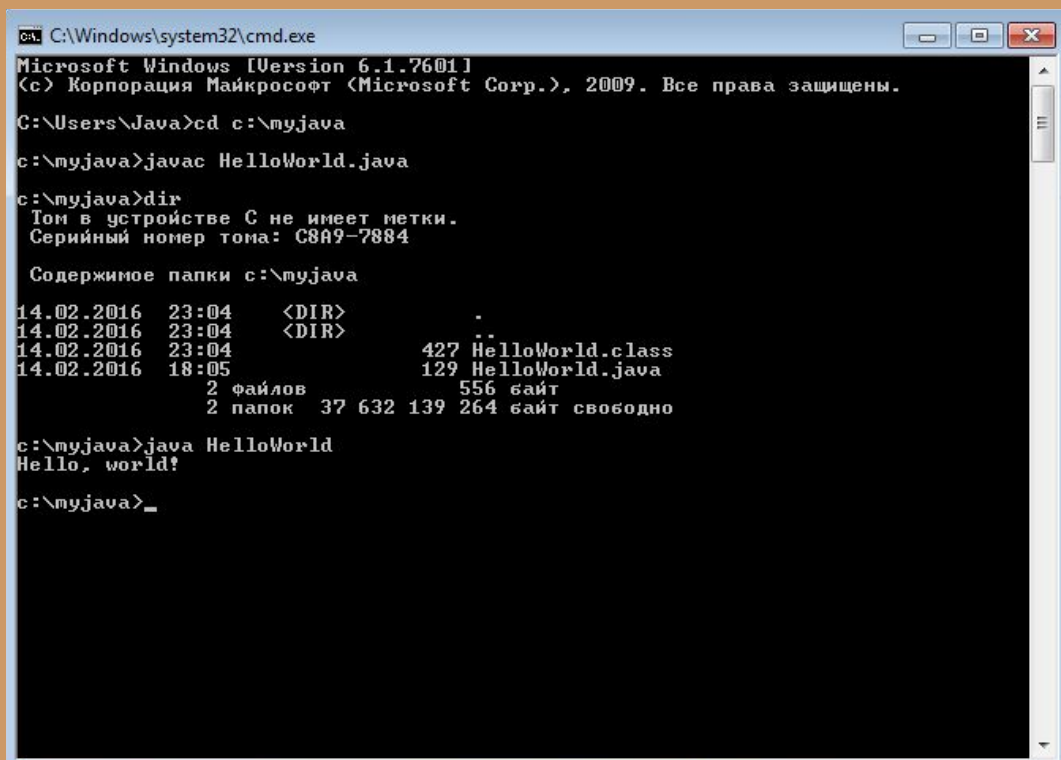
В папке с нашим исходным файлом, должен появиться файл HelloWorld.class. Это можно проверить с помощью команды `dir`. Эта команда выводит на экран список всех файлов, находящихся в выбранной директории или проверить через проводник. Если файл HelloWorld.class присутствует в этом списке, то это значит, что программа скомпилировалась успешно.

Если в коде программы есть ошибка, то компилятор Java при компиляции нам об этом сообщает.

Для запуска нашей программы введем команду

java HelloWorld

и если все перед этим было сделано правильно, то получаем результат — вывод сообщения «Hello World!»



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.

C:\Users\Java>cd c:\myjava
c:\myjava>javac HelloWorld.java
c:\myjava>dir
Том в устройстве C не имеет метки.
Серийный номер тома: C8A9-7884

Содержимое папки c:\myjava

14.02.2016  23:04    <DIR>          .
14.02.2016  23:04    <DIR>          ..
14.02.2016  23:04                427 HelloWorld.class
14.02.2016  18:05                129 HelloWorld.java
                2 файлов          556 байт
                2 папок    37 632 139 264 байт свободно

c:\myjava>java HelloWorld
Hello, world!

c:\myjava>_
```

Домашнее задание

1. Создать новый проект в Eclipse (задать название проекта
ФамилияПервая буква имени на английском языке)
 2. Создать класс отметим опцию "public static void main(String[] args)"
 3. Добавить строку
`System.out.println("Hello World");`
смотри пример из лекции
 4. Заменить Hello World на «Любую фразу на английском языке»
 5. Запустить программу на выполнение.
- Дополнительное задание:
6. Найти в папке с проектом файл вашего класса с расширением java.
 7. Скопировать файл в отдельную папку, попробовать скомпилировать через консоль убедиться, что программа скомпилировалась и запустить на выполнение.