

Java Core Introduction

IT Academy

Agenda

- Introduction to Java Platform. Java SDK
- Environment and Development Tools for Java applications
- Eclipse IDE Tips & Tricks
- The Entry Point into the Program
- Java Packages. Introduction
- My first program



Introduction to Java Platform

- James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991.
- The language was initially called **Oak** after an oak tree. Now, Java it is the name of a **platform** and **language**.
- Java applications are typically compiled to bytecode (class file) that can run on any Java Virtual Machine (**JVM**) regardless of computer architecture.
- The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1995 (May 23, 1995, **Java 1.0**).
- With the advent of **Java 2** (JDK 1.2, released initially as J2SE 1.2 in December 1998–1999), new versions had **multiple configurations** built for different types of platforms.

Introduction to Java Platform

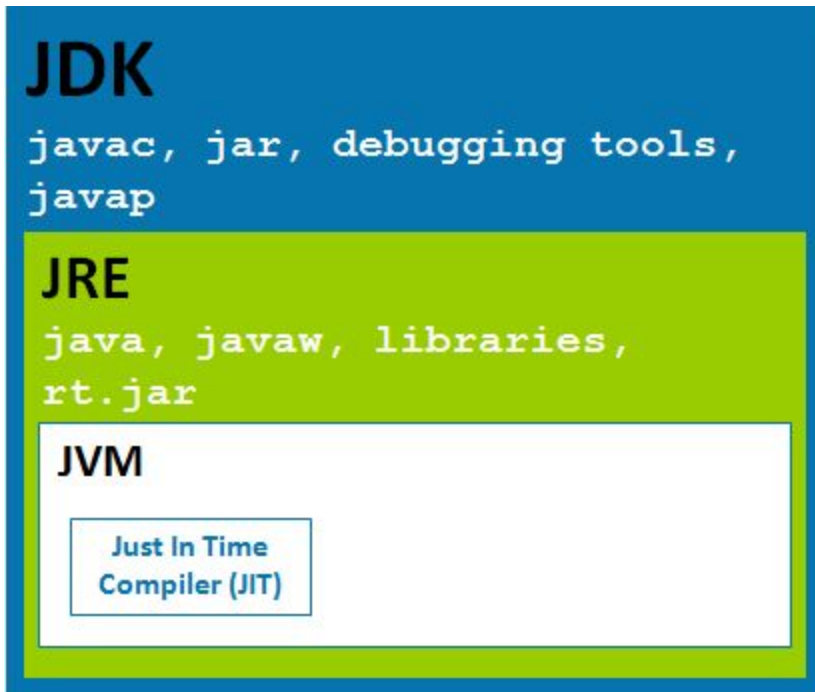
- Sun distinguishes between its Software Development Kit (**SDK**) and Runtime Environment (**JRE**) (a subset of the SDK); the primary distinction involves the JRE's lack of the compiler, utility programs, and header files.
- Sun also distributes a superset of the JRE called the Java Development Kit (commonly known as the **JDK**), which includes development tools such as the Java compiler, Javadoc, Jar, and debugger.
- The JDK forms an extended subset of a SDK. In the descriptions which accompany its recent releases for **Java SE**, **EE**, and **ME**, Sun acknowledges that under its terminology, the JDK forms the subset of the SDK which has the responsibility for the writing and running of Java programs.

Introduction to Java Platform

- Specification **Java 2** (JDK 1.2) – 1998/1999 (strictfp; collection support; Swing; GUI; drag-and-drop; CORBA; Unicode; JIT compiler). Sun renamed new J2 versions as Java **EE**, Java **ME**, and Java **SE**.
- Specification **Java 5** (JDK 1.5) – September 2004 (enum type is class; **generics** mechanism, templates analog of C++; foreach; Autoboxing/Unboxing: for example, scalar and wrapper type `int` – `Integer`; Javadoc; etc.)



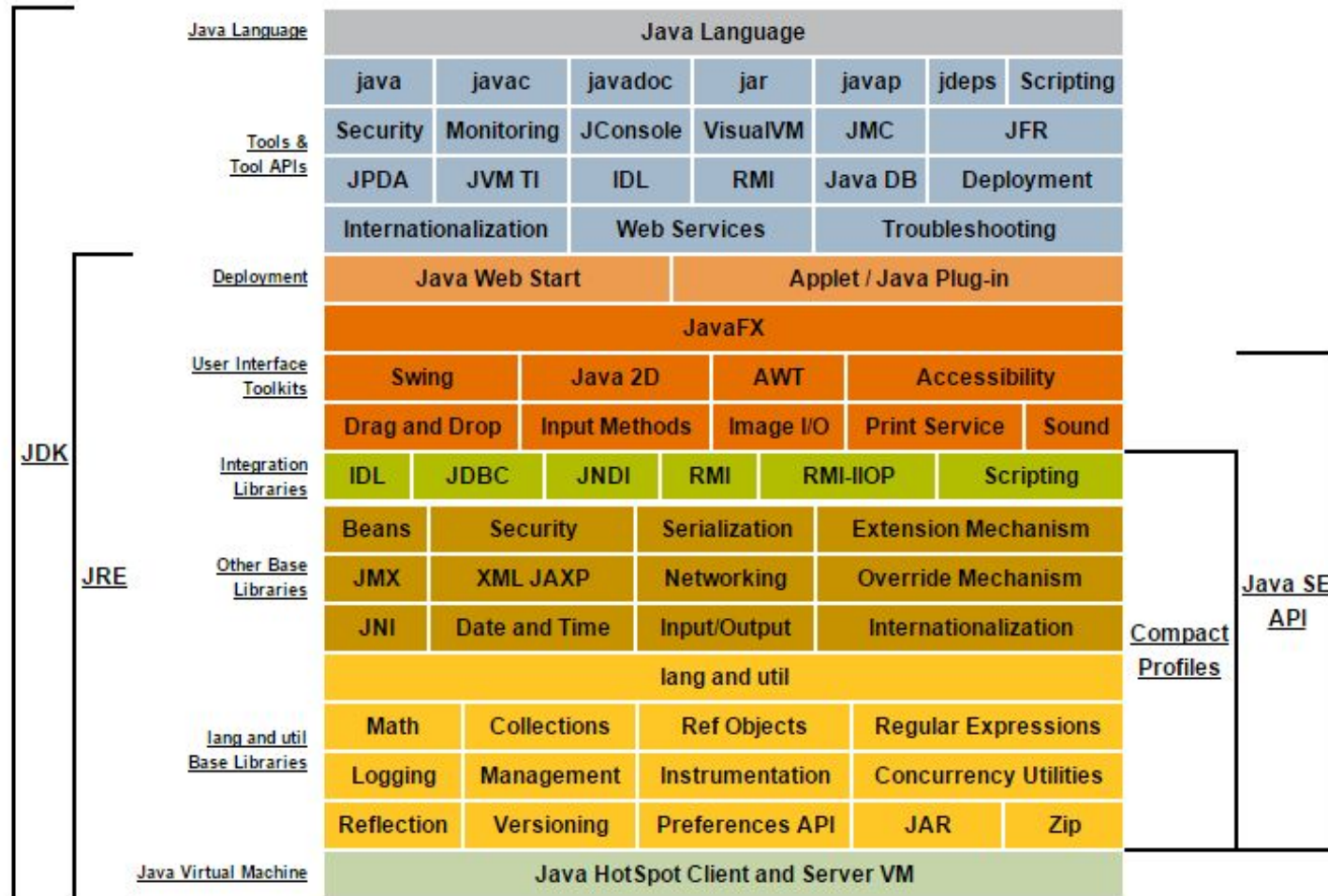
Platform Components



- **JDK = JRE + Development/debugging tools**
- **JRE = JVM + Java Packages Classes (like util, math, lang, awt, swing etc) + runtime libraries.**
- **JVM = Java Virtual Machine**

Detailed Diagram of Java Components

Description of Java Conceptual Diagram



Java JDK

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

ORACLE


Sign In/Register Help Country ▾ Communities ▾ I am a... ▾ I want to... ▾ Search

Products Solutions Downloads Store Support Training Partners About OTN

Oracle Technology Network > Java > Java SE > Downloads


Overview Downloads Documentation Community Technologies Training

Java SE Downloads



DOWNLOAD ↓

Java Platform (JDK) 8u5



DOWNLOAD ↓

JDK 8u5 & NetBeans 8.0

Java Platform, Standard Edition

Java SE 8u5
This release includes important security fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release.
[Learn more](#) ▶

Java SDKs and Tools

- Java SE
- Java EE and Glassfish
- Java ME
- Java Card
- NetBeans IDE
- Java Mission Control

Java Resources

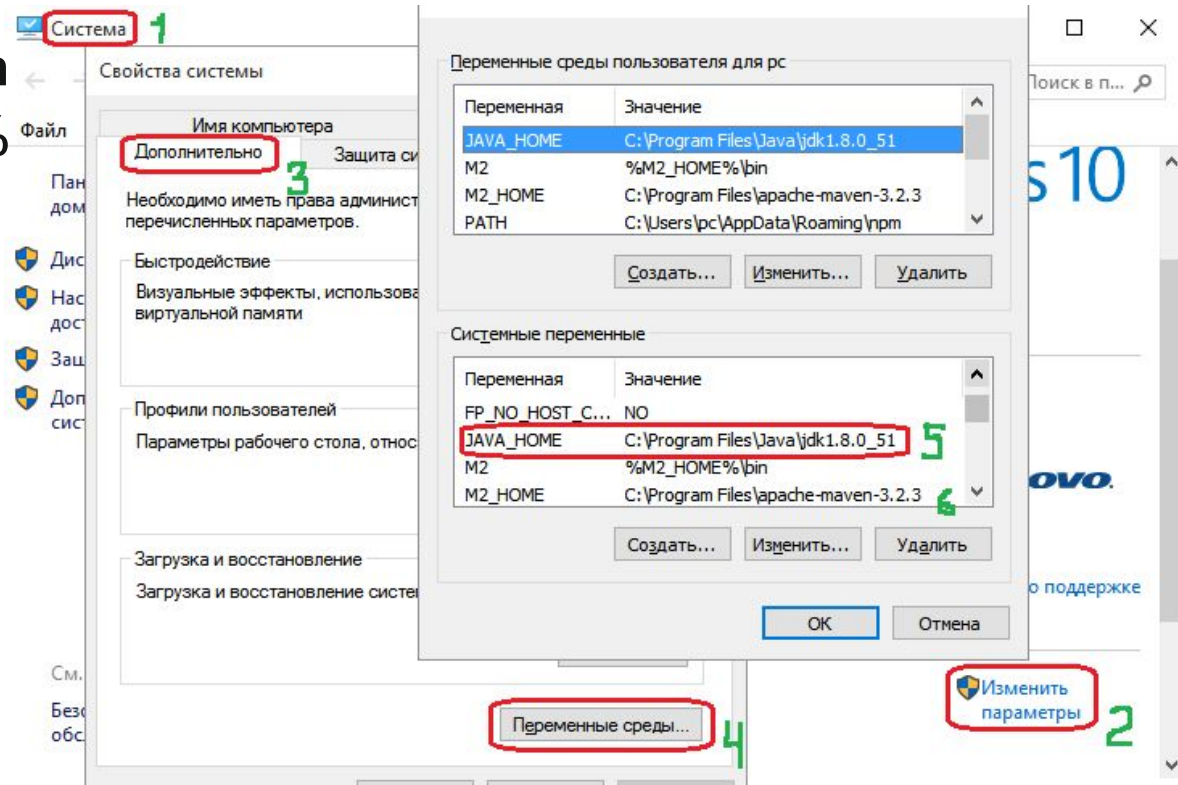
- Java APIs
- Technical Articles
- Demos and Videos
- Forums
- Java Magazine
- Java.net

Java SE
Java EE
Java ME
Java SE Support
Java SE Advanced & Suite
Java Embedded
Java DB
Web Tier
Java Card
Java TV
New to Java
Community
Java Magazine

Install Java

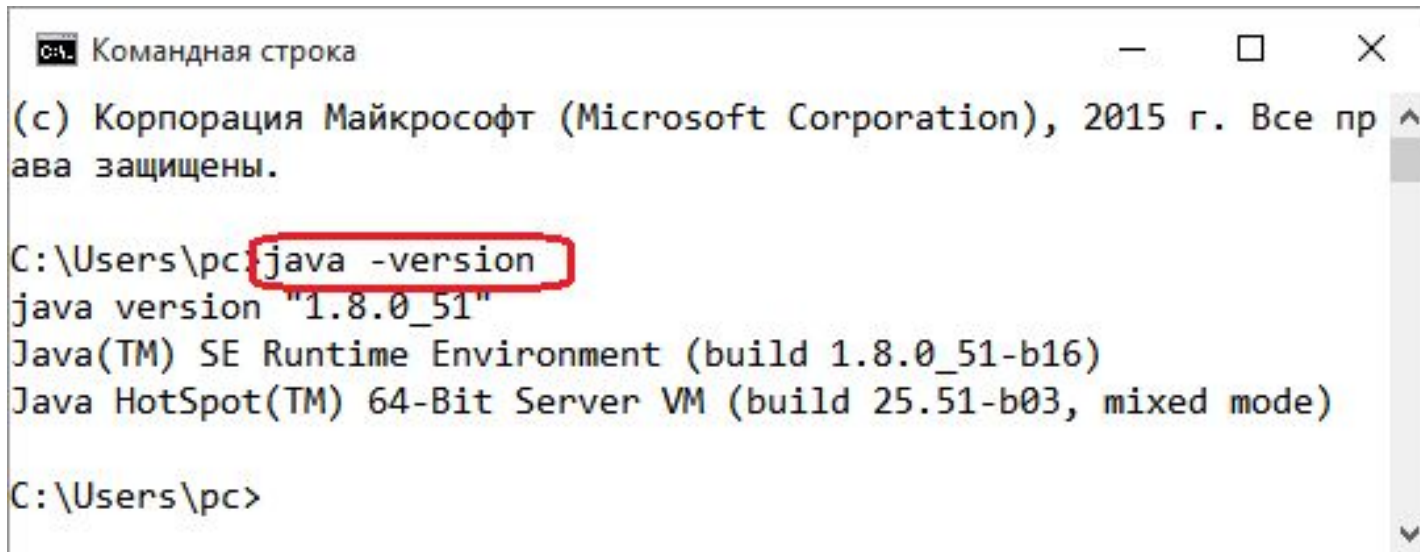
In System Variables

- enter the variable **name** as **JAVA_HOME** and the variable value as the installation path for the Java Development Kit
- add to the variable Path values %JAVA_HOME% and %JAVA_HOME%\bin



Install Java

```
java -version
```



```
Командная строка
(с) Корпорация Майкрософт (Microsoft Corporation), 2015 г. Все права защищены.

C:\Users\pc> java -version
java version "1.8.0_51"
Java(TM) SE Runtime Environment (build 1.8.0_51-b16)
Java HotSpot(TM) 64-Bit Server VM (build 25.51-b03, mixed mode)

C:\Users\pc>
```

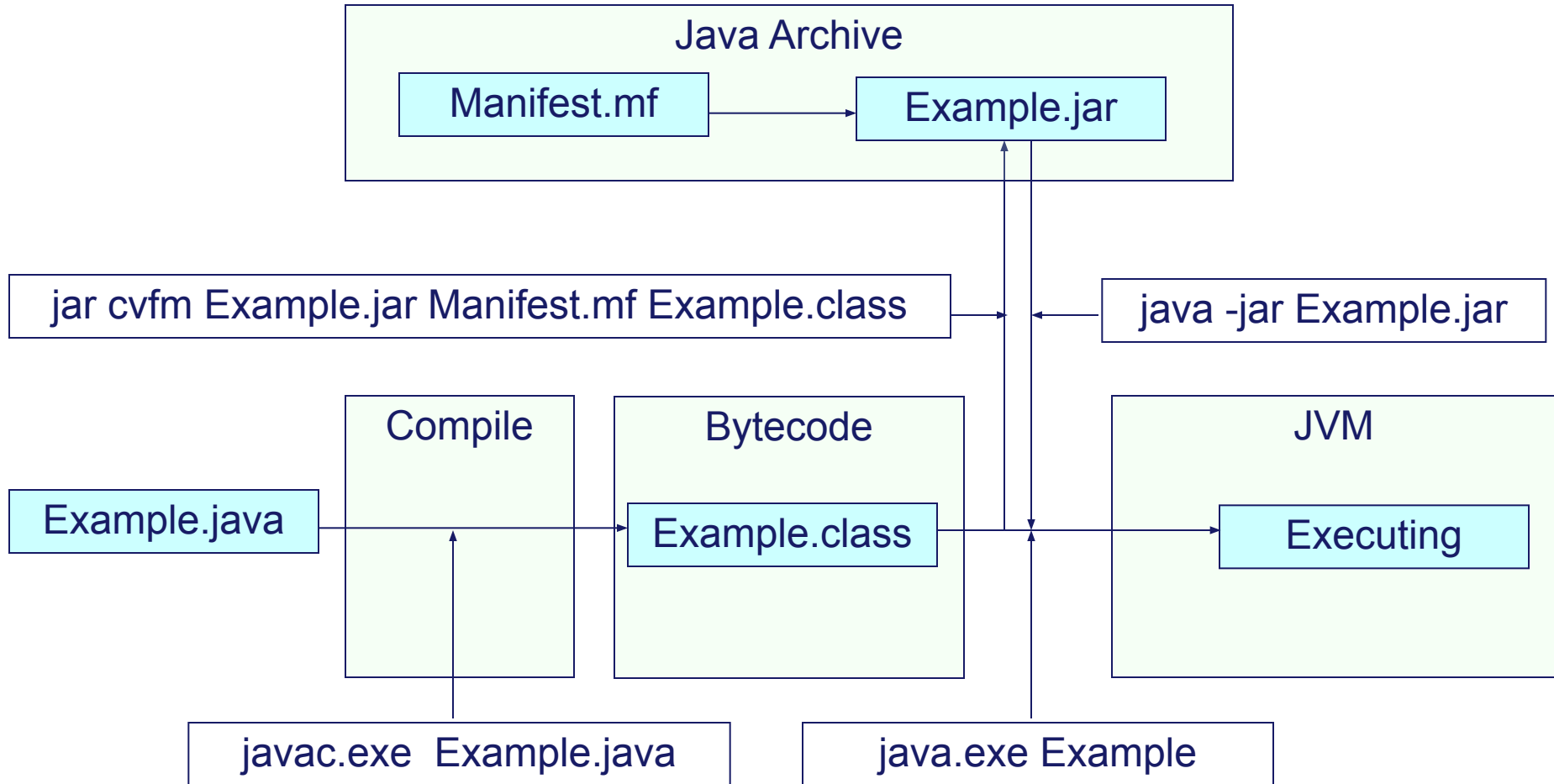
Java JDK. Entry Point

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

To display information on the screen it's used

```
System.out.println("Text1");  
System.out.println("Text 1 " + "Text 2");
```

Java JDK



Java JDK

For Compile:

```
javac Example.java
```

Output file: ***Example.class***

For Running:

```
java Example
```

Exception in thread "main"

java.lang.NoClassDefFoundError: Example

```
java -cp "Example.class;" Example
```

For Creating Java archive:

```
jar cvfm Example.jar Manifest.mf Example.class
```

```
java -jar Example.jar
```

Manifest.mf

Manifest-Version: 1.0

Created-By: 1.8.0_5 (Sun Microsystems Inc.)

Main-Class: Example

Program Entry Point

```
public class ShowArgs {  
    public static void main(String[ ] args) {  
        for (int i=0; i<args.length; i++) {  
            System.out.println("Arg " + i + " is " + args[i]);  
        }  
    }  
}
```

or

```
public class ShowArgs2 {  
    public static void main(String[ ] args) {  
        for (String arg: args) {  
            System.out.println("Command line arg: " + arg);  
        }  
    }  
}
```

Some types of data

Type	Size	Examples
int	4 bytes	-5; 0; 100
double	8 bytes	-6.4; 0.0; 123.56
boolean	≤1 byte	true, false
String		"Student", "Soft Serve", ""

Online Java compiler

- http://www.tutorialspoint.com/compile_java_online.php
- <https://ideone.com/>
- <https://www.compilejava.net/>
- <http://www.browxy.com/>
- <https://www.jdoodle.com/online-java-compiler>

Environment, Development Tools

NetBeans began in 1996 as Xelfi (word play on Delphi). In 1997 Roman Staněk (Charles University in Prague) formed a company around the project and produced commercial versions of the NetBeans IDE until it was bought by Sun Microsystems in 1999. Sun open-sourced the NetBeans IDE

IntelliJ IDEA is a commercial Java IDE by JetBrains. It is often simply referred to as 'IDEA' or 'IntelliJ'. A 30-day fully functional trial of the IntelliJ IDEA commercial edition for various platforms can be freely downloaded. Also available is an open source Community Edition. The first version of IntelliJ IDEA appeared in January 2001. Community Edition is open-source

Environment, Development Tools

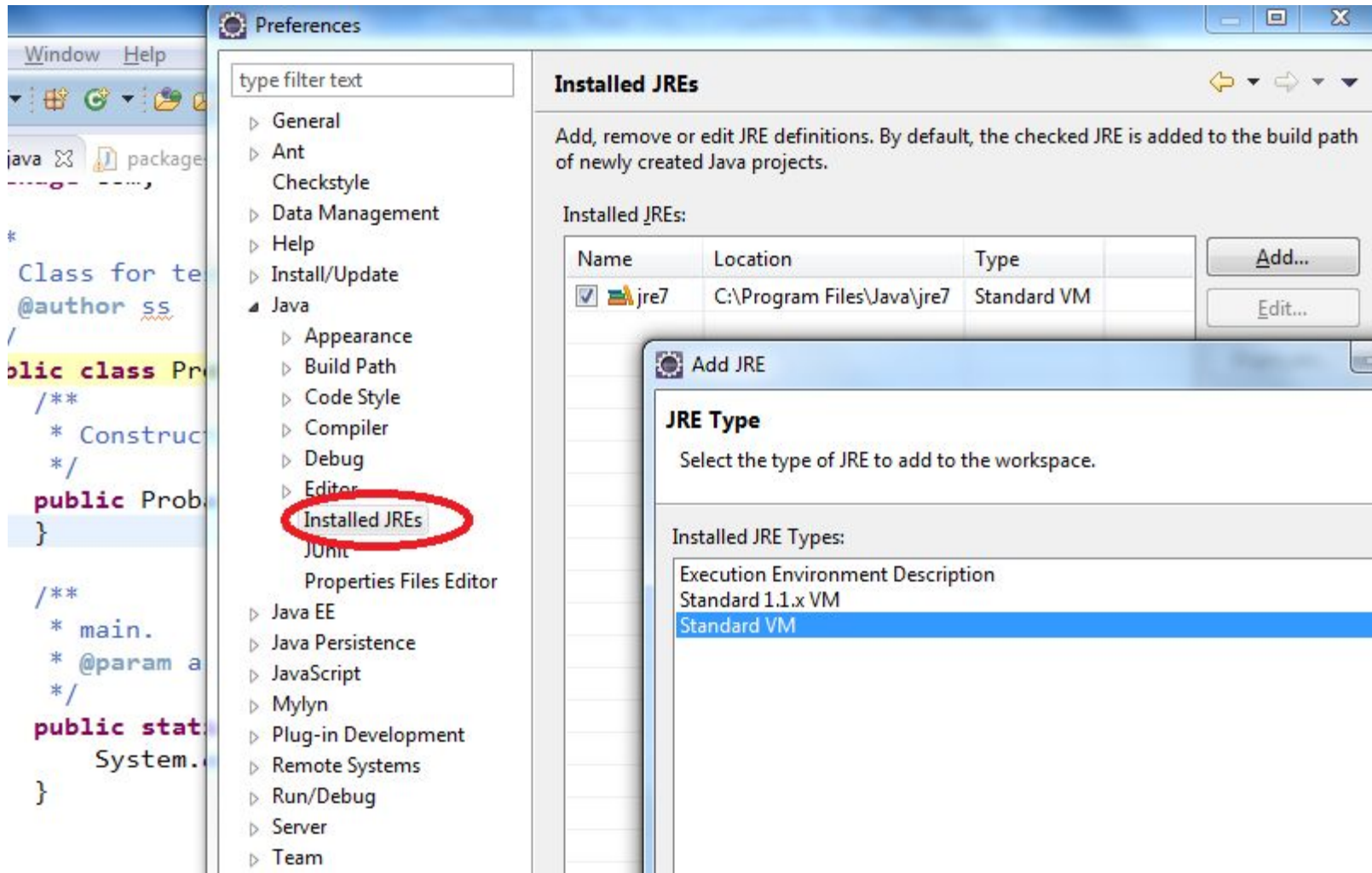
JCreator is a Java IDE created by Xinox Software. Its interface is similar to that of Microsoft's Visual Studio. JCreator has two editions: Lite Edition (LE) is freeware and Pro Edition (Pro, 30-days trial). JCreator is only available on the Windows Operating System.

Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an ***extensible plug-in system***. It is written primarily in **Java** and can be used to develop applications in Java and, by means of various **plug-ins**, other languages including C, C++, COBOL, Python, Perl, PHP, Scala, Scheme and **Ruby** (including Ruby on Rails framework)

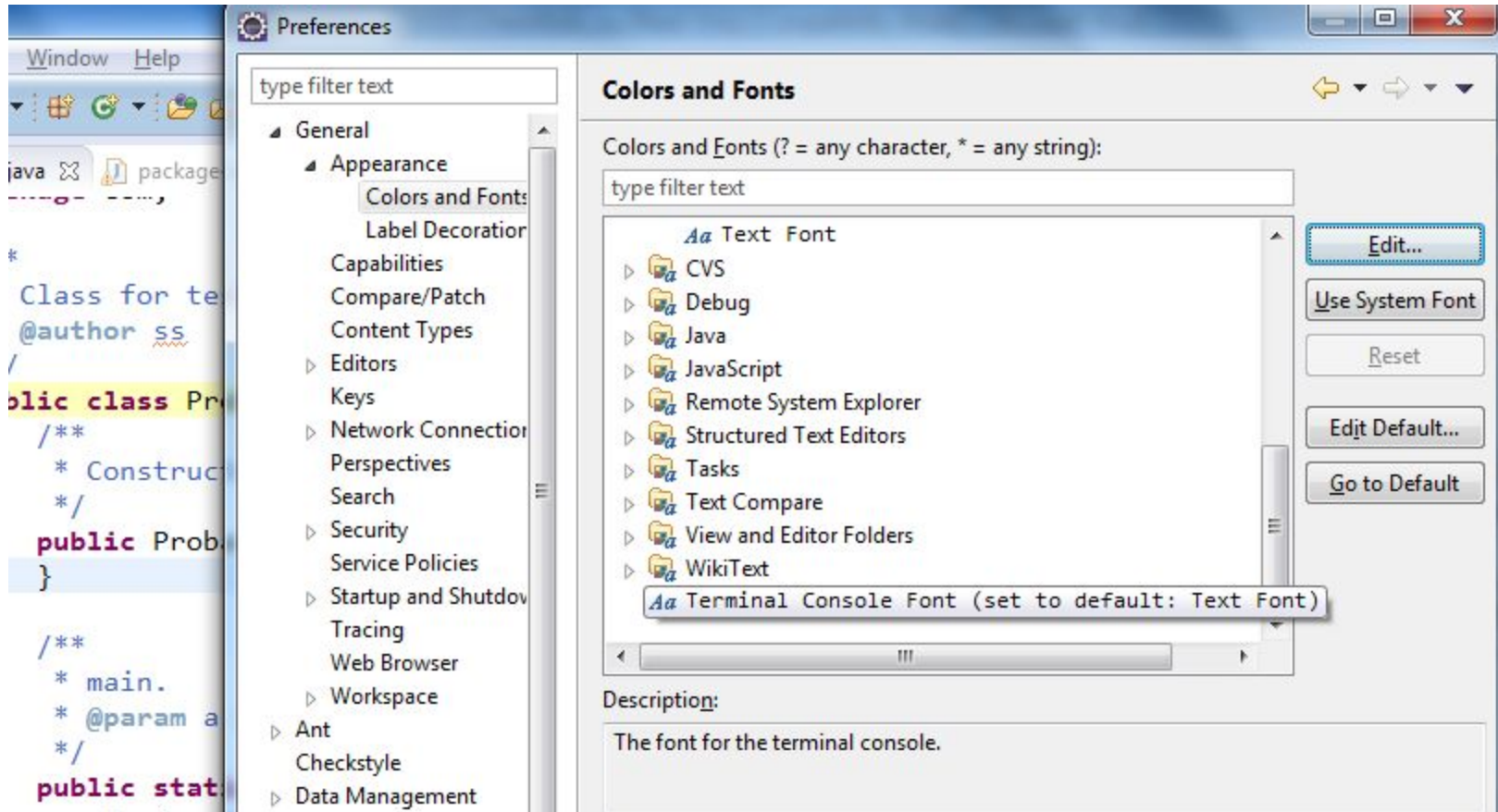
<http://www.eclipse.org/downloads/>



Eclipse



Eclipse



Packages

Java package is a mechanism for organizing Java classes into **namespaces**.

Programmers also typically use packages to organize classes belonging to the same category or providing similar functionality.

- A package provides a **unique** namespace for the types it contains.
- Classes in the same package can access each other's package-access members.

```
package java.awt.event;
```

Packages

imports **all** classes from the package

```
import java.awt.event.*;
```

imports **only** the `ActionEvent` class from the package

```
import java.awt.event.ActionEvent;
```

```
...
```

```
ActionEvent myEvent = new ActionEvent();
```

Classes can also be used directly without an import declaration

```
java.awt.event.ActionEvent myEvent =  
    new java.awt.event.ActionEvent();
```

Packages. Java

`java.lang` – basic language functionality and fundamental types

`java.util` – collection data structure classes

`java.io` – file operations

`java.math` – multiprecision arithmetics

`java.awt` – basic hierarchy of packages for native GUI components

`javax.swing` – hierarchy of packages for platform-independent rich GUI components

The `java.lang.*` package is available **without** the use of an import statement.

Packages

Package names and type names usually differ.

```
package Vector;
public class Mosquito {
    int capacity;
}
//-----
package strange.example;
import java.util.Vector;
import Vector.Mosquito;
class Test {
    public static void main(String[ ] args) {
        System.out.println(new Vector().getClass());
        System.out.println(new Mosquito().getClass());
    }
}
```


Input data

To input value during the run time you can use

```
BufferedReader br = new BufferedReader(  
    new InputStreamReader(System.in));
```

```
String text = br.readLine();  
int age = Integer.parseInt(br.readLine());  
double t = Double.parseDouble(br.readLine());
```

Or

```
Scanner sc = new Scanner(System.in);  
name = sc.nextLine();
```

My first program

```
package com.edu;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(
            new InputStreamReader(System.in));

        System.out.println("Hello. What is your name?");
        String name = br.readLine();
        System.out.println("How old are you?");
        int age = Integer.parseInt(br.readLine());

        System.out.println("Hello " + name);
        System.out.println("You are " + age);
    }
}
```

Tasks

Create Console Application project in Java.

In method `main()` write code for solving next tasks:

1) Define integer variables *a* and *b*. Read values *a* and *b* from Console and calculate: $a + b$, $a - b$, $a * b$, a / b .

Output obtained results.

1) Output question "How are you?". Define string variable *answer*. Read the value *answer* and output: "You are (*answer*)".

HomeWork (online course)

- Please register on UDEMY course "Java Tutorial for Complete Beginners": <https://www.udemy.com/java-tutorial/>
- Complete lessons 1-7:

Section: 1
Introduction

- ✓ 1. Introduction and Installation
- ✓ 2. What Java Is and How It Works
- ✓ 3. Getting a Job and What to Study After Completing Basic Java
- ✓ 4. How To Get The Most Out Of This Course

Section: 2
Programming Core Java

- ✓ 5. A Hello World Program
 - [Java for Complete Beginners \(Video\), Part 1: A Hello World Program | Cave of Programming](#)
- ✓ 6. Using Variables
 - [Java for Complete Beginners \(Video\), Part 2: Using Variables | Cave of Programming](#)
- ✓ 7. Strings: Working With Text
 - [Learn Java Tutorial for Beginners \(Video\), Part 3: Strings](#)

HomeWork

Install JDK and Eclipse.

Create Java project.

Create console application. In method `main()` write code for solving next tasks:

- 1) Flower bed is shaped like a circle. Calculate the perimeter and area by entering the radius. Output obtained results.
- 2) Define string variable *name* and integer value *age*. Output question "What is your name?" Read the value *name* and output next question: "Where are you live, (name)?". Read address and write whole information.
- 3) Phone calls from three different countries are c_1 , c_2 and c_3 standard units per minute. Talks continued t_1 , t_2 and t_3 minutes. How much computer will count for each call separately and all talk together? Input all source data from console, make calculations and output to the screen.

The end

USA HQ

Toll Free: 866-687-3588

Tel: +1-512-516-8880

Ukraine HQ

Tel: +380-32-240-9090

Bulgaria

Tel: +359-2-902-3760