

JAVA PROGRAMMING BASICS

Module 1: Java Overview

Training program

1. Java Fundamentals
2. Start programming with Java, create simple console application
3. Classification of Data Types
4. Primitive types in java
5. Control Flow Statements
6. Arrays

Module contents

- Start programming with Java, create simple console application
 - JDK tools and utilities
 - Overview of IDE
 - Create simple console application
 - Structure of simple console application
 - The main() method
 - String output and input to console
 - Java Code Conventions

Module contents

- Start programming with Java, create simple console application
 - JDK tools and utilities
 - Overview of IDE
 - Create simple console application
 - Structure of simple console application
 - The main() method
 - String output and input to console
 - Java Code Conventions

Standard JDK Tools and Utilities

→ ↺ 🏠 docs.oracle.com/javase/8/docs/technotes/tools/

Standard JDK Tools and Utilities

- **Basic Tools** (appletviewer, extcheck, jar, java, javac, javadoc, javah, javap, jdb, jdeps)
- **Security Tools** (keytool, jarsigner, policytool, kinit, klist, ktab)
- **Internationalization Tools** (native2ascii)
- **Remote Method Invocation (RMI) Tools** (rmic, rmiregistry, rmid, serialver)
- **Java IDL and RMI-IIOP Tools** (tnameserv, idlj, orbd, servertool)
- **Java Deployment Tools** (javapackager, pack200, unpack200)
- **Java Web Start Tools** (javaws)
- **Java Troubleshooting, Profiling, Monitoring and Management Tools** (jcmd, jconsole, jmc, jvisualvm)
- **Java Web Services Tools** (schemagen, wsgen, wsimport, xjc)

1. Installing Java JDK

<https://www.oracle.com/java/technologies/downloads/>

2. Set Environment Variables

3. Check JDK installation

Module contents

- Start programming with Java, create simple console application
 - JDK tools and utilities
 - **Overview of IDE**
 - Create simple console application
 - Structure of simple console application
 - The main() method
 - String output and input to console
 - Java Code Conventions

IDE

- **Eclipse** : <https://www.eclipse.org/downloads/>
- **NetBeans** : <https://netbeans.apache.org/>
- **IntelliJ IDEA** : <https://www.jetbrains.com/idea/>



NetBeans



IntelliJIDEA

Module contents

- Start programming with Java, create simple console application
 - JDK tools and utilities
 - Overview of IDE
 - **Create simple console application**
 - Structure of simple console application
 - The main() method
 - String output and input to console
 - Java Code Conventions

First java program



```
package com.mycompany.lesson1;
```

```
/**
```

```
 * Created by Student.
```

```
 */
```

```
public class MyFirstClass {
```

```
    public static void main(String[] arg){
```

```
        System.out.println("Hello World!");
```

```
    }
```

```
}
```

Module contents

- Start programming with Java, create simple console application
 - JDK tools and utilities
 - Overview of IDE
 - Create simple console application
 - **Structure of simple console application**
 - The main() method
 - String output and input to console
 - Java Code Conventions

Structure of simple console application

package com.mycompany.lesson1; → Package declaration

/**
 * *Created by Student.* → Comments
 */

```
public class MyFirstClass { → Class declaration
    public static void main(String[] arg){
        System.out.println("Hello World!"); → The main method body
    }
}
```

One source file can several classes, but just one can be public!

Name of source file and public class name must be the same !

Module contents

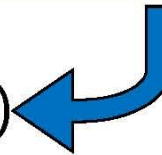
- Start programming with Java, create simple console application
 - JDK tools and utilities
 - Overview of IDE
 - Create simple console application
 - Structure of simple console application
 - **The main() method**
 - String output and input to console
 - Java Code Conventions

The main() method

// This is a simple Java program.

```
public class MyFirstClass
{
    public static void main(String [] args)
    {
    }
}
```

**This is the method header
for the main method**



This area is the body of the main method.

1. Create simple application in IDE



2. String output and input to console

<https://youtu.be/YCnd0IAJHd4>

Module contents

- Start programming with Java, create simple console application
 - JDK tools and utilities
 - Overview of IDE
 - Create simple console application
 - Structure of simple console application
 - The main() method
 - String output and input to console
 - Java Code Conventions

Code Conventions

- Follow Code Conventions
- Improves readability
- For other programmers.
- For yourself.
- Readability means...
- ... less bugs.
- ... easier to debug.

<https://www.oracle.com/docs/tech/java/codeconventions.pdf>

Java Programming Language Coding Conventions 1/2

- Packages:
 - `com.example.domain;`
- Classes, interfaces, and enum types:
 - `SavingsAccount`
- Methods:
 - `getAccount()`
- Variables:
 - `currentCustomer`
- Constants:
 - `HEAD_COUNT`

<https://www.oracle.com/docs/tech/java/codeconventions.pdf>

Java Programming Language Coding Conventions 2/2

Control structures:

```
if (condition ) {  
    statement1;  
} else {  
    statement2;  
}
```

Spacing:

Use one statement per line.

Use two or four spaces for indentation.

Comments:

Use// to comment inline code.

Use/** documentation */ for class members.

<https://www.oracle.com/docs/tech/java/codeconventions.pdf>

Summary

- Downloading and installing JDK
- Set Environment variables
- Create Java source file and compile and run it
- Downloading IntelliJ IDEA
- Create sample Java project
- Executing a class that has main() method
- Apple Java Code Conventions