Experiment Zero [0]

Analyze the different java programming

Define, use, and differentiate such concepts as JSP JAVA PACKAGES.

Explain JDK, JRE and JVM

Sol:

Viva Questions

**1 Explain JDK, JRE and JVM?**

|  |  |  |
| --- | --- | --- |
| **JDK vs. JRE vs. JVM** | | |
| **JDK** | **JRE** | **JVM** |
| It stands for Java Development Kit. | It stands for Java Runtime Environment. | It stands for Java Virtual Machine. |
| It is the tool necessary to compile, document and package Java programs. | JRE refers to a runtime environment in which Java byte code can be executed. | It is an abstract machine. It is a specification that provides a run-time environment in which Java byte code can be executed. |
| It contains JRE + development tools. | It’s an implementation of the JVM which physically exists. | JVM follows three notations: Specification, **Implementation,**and **Runtime Instance**. |

**2 Explain public static void main (String args []) in Java.**

Main () in Java is the entry point for any Java program. It is always written as **public static void main (String [] args)**.

**Public**: Public is an access modifier, which is used to specify who can access this method. Public means that this Method will be accessible by any Class.

**Static**: It is a keyword in java which identifies it is class-based. Main () is made static in Java so that it can be accessed without creating the instance of a Class. In case, main is not made static then the compiler will throw an error as **main** () is called by the JVM before any objects are made and only static methods can be directly invoked via the class.

**Void**: It is the return type of the method. Void defines the method which will not return any value.

**Main**: It is the name of the method which is searched by JVM as a starting point for an application with a particular signature only. It is the method where the main execution occurs.

**String args []**: It is the parameter passed to the main method.

### **3 What are wrapper classes in Java?**

Wrapper classes convert the Java primitives into the reference types (objects). Every primitive data type has a class dedicated to it. These are known as wrapper classes because they “wrap” the primitive data type into an object of that class. Refer to the below image which displays different primitive type, wrapper class and constructor argument.

**4 What are the differences between Heap and Stack Memory in Java?**

The major difference between Heap and Stack memory are:

|  |  |  |
| --- | --- | --- |
| **Features** | **Stack** | **Heap** |
| **Memory** | Stack memory is used only by one thread of execution. | Heap memory is used by all the parts of the application. |
| **Access** | Stack memory can’t be accessed by other threads. | Objects stored in the heap are globally accessible. |
| **Memory Management** | Follows LIFO manner to free memory. | Memory management is based on the generation associated with each object. |
| **Lifetime** | Exists until the end of execution of the thread. | Heap memory lives from the start till the end of application execution. |
| **Usage** | Stack memory only contains local primitive and reference variables to objects in heap space. | Whenever an object is created, it’s always stored in the Heap space. |

**5 What is a package in Java? List down various advantages of packages.**

Packages in Java are the collection of related classes and interfaces which are bundled together. By using packages, developers can easily modularize the code and optimize its reuse. Also, the code within the packages can be imported by other classes and reused. Below I have listed down a few of its advantages:

* Packages help in avoiding name clashes
* They provide easier access control on the code
* Packages can also contain hidden classes which are not visible to the outer classes and only used within the package
* Creates a proper hierarchical structure which makes it easier to locate the related classes