www.Javagenious.com

Free Downloads From www.Javagenious.com

10/10/2011

info@javagenious.com

Q:What if the main method is declared as private?
A:The program compiles properly but at runtime it will give "Main method not public." message.

Q:What if the static modifier is removed from the signature of the main method?
A:Program compiles. But at runtime throws an error "NoSuchMethodError".

Q:What if I write static public void instead of public static void?
A:Program compiles and runs properly.

Q:What if I do not provide the String array as the argument to the method?
A:Program compiles but throws a runtime error "NoSuchMethodError".

Q:What is the first argument of the String array in main method?
A:The String array is empty. It does not have any element. This is unlike C/C++ where the first element by default is the program name.

Q:If I do not provide any arguments on the command line, then the String array of Main method will be empty or null?
A:It is empty. But not null.

Q:How can one prove that the array is not null but empty using one line of code?
A:Print args.length. It will print 0. That means it is empty. But if it would have been null then it would have thrown a NullPointerException on attempting to print args.length.

Q:What environment variables do I need to set on my machine in order to be able to run Java programs?
A:CLASSPATH and PATH are the two variables.

Q:Can an application have multiple classes having main method?
A:Yes it is possible. While starting the application we mention the class name to be run. The JVM will look for the Main method only in the class whose name you have mentioned. Hence there is not conflict amongst the multiple classes having main method.

Q:Can I have multiple main methods in the same class?
A:No the program fails to compile. The compiler says that the main method is already defined in the class.

Q:Do I need to import java.lang package any time? Why ?

A: No. It is by default loaded internally by the JVM.

Q:Can I import same package/class twice? Will the JVM load the package twice at runtime?

A: One can import the same package or same class multiple times. Neither compiler nor JVM complains abt it. And the JVM will internally load the class only once no matter how many times you import the same class.

Q:What are Checked and UnChecked Exception?

A: A checked exception is some subclass of Exception (or Exception itself), excluding class RuntimeException and its subclasses.

Making an exception checked forces client programmers to deal with the possibility that the exception will be thrown. eg, IOException thrown by java.io.FileInputStream's read() method·

Unchecked exceptions are RuntimeException and any of its subclasses. Class Error and its subclasses also are unchecked. With an unchecked exception, however, the compiler doesn't force client programmers either to catch the

exception or declare it in a throws clause. In fact, client programmers may not even know that the exception could be thrown. eg, StringIndexOutOfBoundsException thrown by String's charAt() method· Checked exceptions must be caught at compile time. Runtime exceptions do not need to be. Errors often cannot be.

Q:What is Overriding?

A: When a class defines a method using the same name, return type, and arguments as a method in its superclass, the method in the class overrides the method in the superclass.

When the method is invoked for an object of the class, it is the new definition of the method that is called, and not the method definition from superclass. Methods may be overridden to be more public, not more private.

Q:What are different types of inner classes?

A: *Nested top-level classes*, *Member classes, Local classes, Anonymous classes*

Nested top-level classes- If you declare a class within a class and specify the static modifier, the compiler treats the class just like any other top-level class.

Any class outside the declaring class accesses the nested class with the declaring class name acting similarly to a package. eg, outer.inner. Top-level inner classes implicitly have access only to static variables.There can also be inner interfaces. All of these are of the nested top-level variety.

Member classes - Member inner classes are just like other member methods and member variables and access to the member class is restricted, just like methods and variables. This means a public member class acts similarly to a nested top-level class. The primary difference between member classes and nested top-level classes is that member classes have access to the specific instance of the enclosing class.

Local classes - Local classes are like local variables, specific to a block of code. Their visibility is only within the block of their declaration. In order for the class to be useful beyond the declaration block, it would need to implement a
more publicly available interface.Because local classes are not members, the modifiers public, protected, private, and static are not usable.

Anonymous classes - Anonymous inner classes extend local inner classes one level further. As anonymous classes have no name, you cannot provide a constructor.