www.Javagenious.com

Free Downloads From www.Javagenious.com

10/10/2011

info@javagenious.com

Q:Are the imports checked for validity at compile import such as java.lang.ABCD compile?

A:Yes the imports are checked for the semantic validity at compile time. The code containing above line of import will not compile. It will throw an error saying,can not resolve symbol
symbol : class ABCD
location: package io
import java.io.ABCD;

Q:Does importing a package imports the subpackages as well? e.g. Does importing com.MyTest.* also import com.MyTest.UnitTests.*?

A:No you will have to import the subpackages explicitly. Importing com.MyTest.* will import classes in the package MyTest only. It will not import any class in any of it's subpackage.

Q:What is the difference between declaring a variable and defining a variable?

A:In declaration we just mention the type of the variable and it's name. We do not initialize it. But defining means declaration + initialization. e.g String s; is just a declaration while String s = new String ("abcd"); Or String s = "abcd"; are both definitions.

Q:What is the default value of an object reference declared as an instance variable?

A:null unless we define it explicitly.

Q:Can a top level class be private or protected?

A:No. A top level class can not be private or protected. It can have either "public" or no modifier. If it does not have a modifier it is supposed to have a default access.If a top level class is declared as private the compiler will complain that the "modifier private is not allowed here". This means that a top level class can not be private. Same is the case with protected.

Q:What type of parameter passing does Java support?

A:In Java the arguments are always passed by value .

Q:Primitive data types are passed by reference or pass by value?

A:Primitive data types are passed by value.

Q:Objects are passed by value or by reference?

A:Java only supports pass by value. With objects, the object reference itself is passed by value and so both the original reference and parameter copy both refer to the same object .

Q:What is serialization?

A:Serialization is a mechanism by which you can save the state of an object by converting it to a byte stream.

Q:How do I serialize an object to a file?

A:The class whose instances are to be serialized should implement an interface Serializable. Then you pass the instance to the ObjectOutputStream which is connected to a fileoutputstream. This will save the object to a file.

Q:Which methods of Serializable interface should I implement?

A:The serializable interface is an empty interface, it does not contain any methods. So we do not implement any methods.

Q:How can I customize the seralization process? i.e. how can one have a control over the serialization process?

A:Yes it is possible to have control over serialization process. The class should implement Externalizable interface. This interface contains two methods namely readExternal and writeExternal. You should implement these methods and write the logic for customizing the serialization process.

Q:What is the common usage of serialization?

A:Whenever an object is to be sent over the network, objects need to be serialized. Moreover if the state of an object is to be saved, objects need to be serilazed.

Q:What is Externalizable interface?

A:Externalizable is an interface which contains two methods readExternal and writeExternal. These methods give you a control over the serialization mechanism. Thus if your class implements this interface, you can customize the serialization process by implementing these methods.

Q:When you serialize an object, what happens to the object references included in the object?

A:The serialization mechanism generates an object graph for serialization. Thus it determines whether the included object references are serializable or not. This is a recursive process. Thus when an object is serialized, all the included objects are also serialized alongwith the original obect.

Q:What one should take care of while serializing the object?

A:One should make sure that all the included objects are also serializable. If any of the objects is not serializable then it throws a NotSerializableException.

Q:What happens to the static fields of a class during serialization?

A:There are three exceptions in which serialization doesnot necessarily read and write to the stream. These are
1. Serialization ignores static fields, because they are not part of ay particular state state.
2. Base class fields are only hendled if the base class itself is serializable.
3. Transient fields.