Java & JEE Training

Day 18+19 – Review Sessions

MindsMapped Consulting

Java Interview Tips
+
Typical Interview Questions

Interview Tips

- How to prepare for a typical Java interview?
- Typical interview questions…

Give few reasons for using Java / Advantages of Java over other programming languages

- Built-in support for multi-threading, socket communication, and memory management (automatic garbage collection).
- Object Oriented (OO).
- Better portability than other languages across operating systems
- Supports Web based applications (Applet, Servlet, and JSP), distributed applications (sockets, RMI, EJB etc.) and network protocols (HTTP, JRMP etc) with the help of extensive standardized APIs (Application Programming Interfaces).

Which package is imported implicitly by all Java programs?

The java.lang package is imported implicitly.

What are static blocks or static initializers?

- When a class is loaded, all blocks that are declared static and don’t have function name (i.e. static initializers) are executed even before the constructors are executed. As the name suggests they are typically used to initialize static fields.

Difference between constructors and other regular methods?

- Constructors must have the same name as the class name and cannot return a value.
- The constructors are called only once per creation of an object while regular methods can be called many times

Can you call one constructor from another?

Yes, by using this()

```
public Pet(int id) {
  this.id = id; // “this” means this object
}
public Pet (int id, String type) {
    this(id); // calls constructor public Pet(int id)
    this.type = type; // ”this” means this object
}
```

How to call superclass constructor?

By using super keyword

```
public SpecialPet(int id) {
    super(id); //must be the very first statement in the constructor.
}
```

You can also use super.myMethod() to call the superclass's myMethod. This can be called at any line.

What happens if you do not provide a constructor?

Java runtime provides a default constructor

Advantages of OOP?

- Encapsulation
- Polymorphism
- Inheritance
- Abstraction

Difference between inheritance and composition/aggregation?

- The *'**is a**'* relationship is expressed with **inheritance** and *'**has a**'* relationship is expressed with **composition**. Both inheritance and composition allow you to place sub-objects inside your new class. Two of the main techniques for **code reuse** are **class inheritance** and **object composition**

- **Example: ScienceStudent is a Student; Student has an Address.**

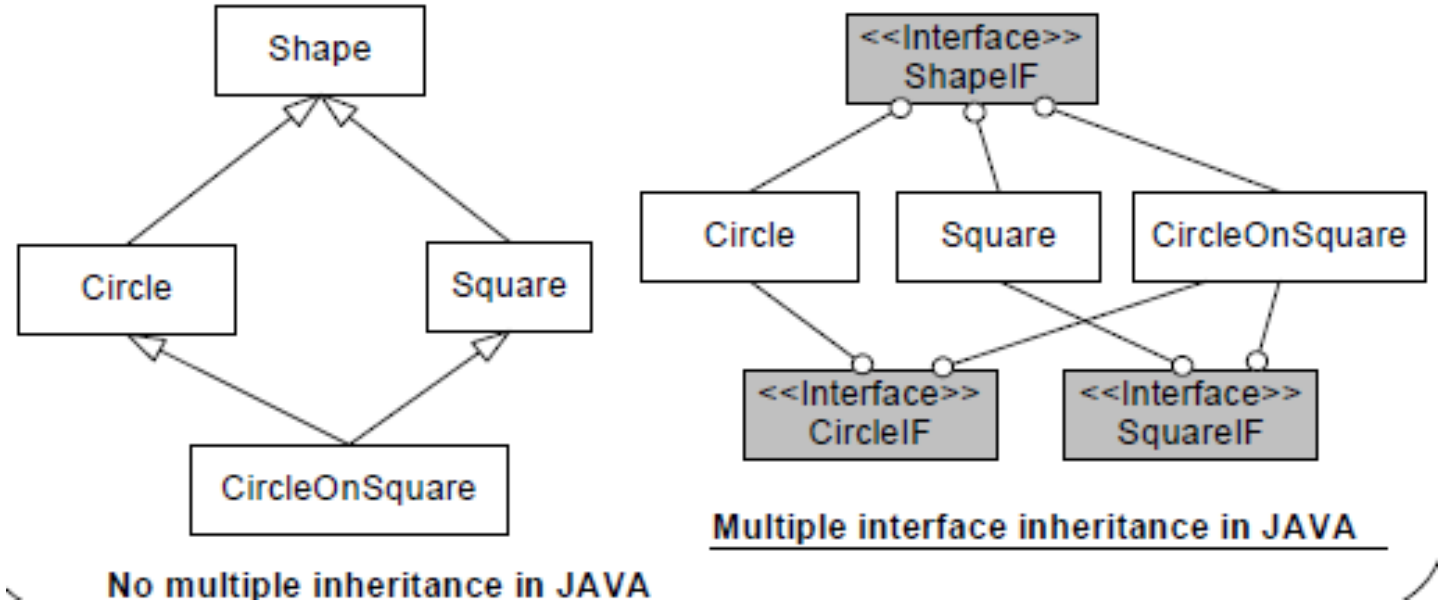
Difference between abstract class and interface?

Refer previous session slides

- Abstract classes have executable methods and abstract methods. Interfaces have no implementation code. All methods are abstract.
- One class can only subclass one abstract class. A class can implement any number of interfaces

Why multiple inheritance is not allowed?

- Diamond problem



Difference between Method Overloading vs Overriding?

- Please refer lecture slides

Difference between Comparable and Comparator interface?

- Please refer lecture slides

Difference between == and equals()?

??

Explain the non-final methods toString(), hashCode() and equals() of Object class

- Refer lecture

Difference between String and StringBuffer/StringBuilder

- Please refer lecture

Explain the different access modifiers..

- Public
- Private
- Default
- Protected

Code output?

1. What is true about the following code?

```
1.  short s1 = 100;
2.  Short s2 = s1;
3.  Integer i = s1;
```

   a. There is a compile error at line 1.

   b. There is a compile error at line 2.

   c. There is a compile error at line 3.

   d. It will compile fine.

Code output?

1. What is true about the following code?

```
1.  short s1 = 100;
2.  Short s2 = s1;
3.  Integer i = s1;
```

a. There is a compile error at line 1.

b. There is a compile error at line 2.

c. There is a compile error at line 3.

d. It will compile fine.

c. Autoboxing will convert a short to a Short, but not to an Integer.

Code output?

What is true about the following code?

```
1.  Integer i = 10;
2.  int j = 5;
3.  if (i.compareTo(j) > 0) {
4.    System.out.println("OK");
5.  }
```

a. It will print "OK".

b. It will print nothing.

c. There is a compile error at line 2.

d. There is a compile error at line 3.

Code output?

What is true about the following code?

```
1.  Integer i = 10;
2.  int j = 5;
3.  if (i.compareTo(j) > 0) {
4.    System.out.println("OK");
5.  }
```

a. It will print "OK".

b. It will print nothing.

c. There is a compile error at line 2.

d. There is a compile error at line 3.

a. "j" will be converted to an Integer automatically when it is passed to compareTo().

Code output?

What is true about the following code?

```
1.  int i = 10;
2.  Integer j = 5;
3.  if (i.compareTo(j) > 0) {
4.    System.out.println("OK");
5.  }
```

a. It will print "OK".

b. It will print nothing.

c. There is a compile error at line 2.

d. There is a compile error at line 3.

Code output?

What is true about the following code?

```
1.  int i = 10;
2.  Integer j = 5;
3.  if (i.compareTo(j) > 0) {
4.    System.out.println("OK");
5.  }
```

a. It will print "OK".

b. It will print nothing.

c. There is a compile error at line 2.

d. There is a compile error at line 3.

d. "i" will not be converted to an Integer automatically because there is no assignment nor casting there.

What is the output?

```
class Foo {
  void g(int i) {
    System.out.println("a");
  }
}
class Bar extends Foo {
  void g(Integer i) {
    System.out.println("b");
  }
}
...
Bar b = new Bar();
b.g(10);
```

What is the output?

```
class Foo {
  void g(int i) {
    System.out.println("a");
  }
}
class Bar extends Foo {
  void g(Integer i) {
    System.out.println("b");
  }
}
...
Bar b = new Bar();
b.g(10);
```

It will print “a”. The g() in Bar is not overriding the g() in Foo. It is just overloading the name "g". When determining which method to call, autoboxing is not considered first so only the g() in Foo is applicable and thus is used.

True or False

"X extends Y" is correct if and only if X is a class and Y is an interface.

True or False

"X extends Y" is correct if and only if X is a class and Y is an interface.

False

Question

Which method names follow the JavaBeans standard? (Choose all that apply.)

A. `addSize`

B. `getCust`

C. `deleteRep`

D. `isColorado`

E. `putDimensions`

Question

Which method names follow the JavaBeans standard? (Choose all that apply.)

A. `addSize`
B. `getCust`
C. `deleteRep`
D. `isColorado`
E. `putDimensions`

B and D

What is the output?

```
package pkgA;
public class Foo {
  int a = 5;
  protected int b = 6;
  public int c = 7;
}

package pkgB;
import pkgA.*;
public class Baz {
  public static void main(String[] args) {
    Foo f = new Foo();
    System.out.print(" " + f.a);
    System.out.print(" " + f.b);
    System.out.print(" " + f.c);
  }
}
```

What is the output?

```
package pkgA;
public class Foo {
  int a = 5;
  protected int b = 6;
  public int c = 7;
}

package pkgB;
import pkgA.*;
public class Baz {
  public static void main(String[] args) {
    Foo f = new Foo();
    System.out.print(" " + f.a);
    System.out.print(" " + f.b);
    System.out.print(" " + f.c);
  }
}
```

Compilation error on lines that print f.a and f.b

Which of these is not correct?

```
public class Electronic implements Device
      { public void doIt() { } }

abstract class Phone1 extends Electronic { }

abstract class Phone2 extends Electronic
  { public void doIt(int x) { } }

class Phone3 extends Electronic implements Device
  { public void doStuff() { } }

interface Device { public void doIt(); }
```

Which of these is not correct?

```
public class Electronic implements Device
      { public void doIt() { } }

abstract class Phone1 extends Electronic { }

abstract class Phone2 extends Electronic
  { public void doIt(int x) { } }

class Phone3 extends Electronic implements Device
  { public void doStuff() { } }

interface Device { public void doIt(); }
```

All are correct

What is the output?

```
public class Frodo extends Hobbit {
  public static void main(String[] args) {
    Short myGold = 7;
    System.out.println(countGold(myGold, 6));
  }
}
class Hobbit {
  int countGold(int x, int y) { return x + y; }
}
```

What is the output?

```
public class Frodo extends Hobbit {
  public static void main(String[] args) {
    Short myGold = 7;
    System.out.println(countGold(myGold, 6));
  }
}
class Hobbit {
  int countGold(int x, int y) { return x + y; }
}
```

Compilation error… cannot invoke countGold from static context

What is the difference between a List and a Set?

Set:

HashSet: Default Set …must implement equals and hashCode

LinkedHashSet: Maintains the elements in the order in which they were entered

TreeSet: Maintains in ascending order/sorted

TreeSet<Student> ts = new TreeSet<Student>();

Comparable: Default comparison logic…. Student implements Comparable… override compareTo()

Comparator: Multiple Comparator classes by implementing Comparator and overriding compare()

What is the difference between a List and a Set?

- List, by default characteristic maintains order; a set does not.
- A set does not entertain duplicate members; whereas a List does