Vendor: Oracle

Exam Code: 1Z0-804

Exam Name: Java SE 7 Programmer II

Version: DEMO

QUESTION 1
Given these facts about Java types in an application:

```
- Type x is a template for other types in the application.
- Type x implements dostuff ().
- Type x declares, but does NOT implement doit().
- Type y declares doOther() .
```

Which three are true?

A. Type y must be an interface.
B. Type x must be an abstract class.
C. Type y must be an abstract class.
D. Type x could implement or extend from Type y.
E. Type x could be an abstract class or an interface.
F. Type y could be an abstract class or an interface.

Answer: BDF
Explanation:
Unlike interfaces, abstract classes can contain fields that are not static and final, and they can contain implemented methods. Such abstract classes are similar to interfaces, except that they provide a partial implementation, leaving it to subclasses to complete the implementation. If an abstract class contains only abstract method declarations, it should be declared as an interface instead.
Note:
An interface in the Java programming language is an abstract type that is used to specify an interface (in the generic sense of the term) that classes must implement. Interfaces are declared using the interface keyword, and may only contain method signature and constant declarations (variable declarations that are declared to be both static and final). An interface may never contain method definitions.
Note 2:
an abstract class is a class that is declared abstract--it may or may not include abstract methods. Abstract classes cannot be instantiated, but they can be subclassed. An abstract method is a method that is declared without an implementation (without braces, and followed by a semicolon)

QUESTION 2
Which represents part of a DAO design pattern?

A.
```
interface EmployeeDAO {
  int getID();
  Employee findByID (intid);
  void update();
  void delete();
  }
```
B.
```
class EmployeeDAO {
  int getID() { return 0;}
  Employee findByID (int id) { return null;}
  void update () {}
  void delete () {}
  }
```
C.
```
class EmployeeDAO {
  void create (Employee e) {}
  void update (Employee e) {}
  void delete (int id) {}
```

```
      Employee findByID (int id) {return id}
      }
D.  interface EmployeeDAO {
      void create (Employee e);
      void update (Employee e);
      void delete (int id);
      Employee findByID (int id);
      }
E.  interface EmployeeDAO {
      void create (Connection c, Employee e);
      void update (Connection c, Employee e);
      void delete (Connection c, int id);
      Employee findByID (Connection c, int id);
      }
```

Answer: D

QUESTION 3
Assuming the port statements are correct, which two code fragments create a one-byte file?

A. OutputStream fos = new FileOutputStream(new File("/tmp/data.bin")); OutputStream bos = new BufferedOutputStream(fos);
    DataOutputStream dos = new DataOutputStream(bos);
    dos.writeByte(0);
    dos.close();
B. OutputStream fos = new FileOutputStream ("/tmp/data.bin"); dataOutputStream dos = new DataOutputStream(fos);
    dos.writeByte(0);
    dos.close();
C. OutputStream fos = new FileOutputStream (new File ("/tmp/data.bin")); dataOutputStream dos = new DataOutputStream(os);
    dos.writeByte(0);
    dos.close();
D. OutputStream fos = new FileOutputStream ("/tmp/data.bin"); fos.writeByte(0);
    fos.close();

Answer: BC
Explanation:
B:Create DataOutputStream from FileOutputStream public static void main(String[] args) throws
Exception { FileOutputStream fos = new FileOutputS tream("C:/demo.txt"); DataOutputStream
dos = new DataOutputStream(fos);
Note:
The FileOutputStream class is a subclass of OutputStream. You can construct a
FileOutputStream object by passing a string containing a path name or a File object.
You can also specify whether you want to append the output to an existing file.
public FileOutputStream (String path)
public FileOutputStream (String path, boolean append)
public FileOutputStream (File file)
public FileOutputStream (File file, boolean append)
With the first and third constructors, if a file by the specified name already exists, the file will be
overwritten. To append to an existing file, pass true to the second or fourth constructor.
Note 2:
public class DataOutputStream
extends FilterOutputStream

---

implements DataOutput
A data output stream lets an application write primitive Java data types to an output stream in a portable way. An application can then use a data input stream to read the data back in.
Reference:java.io Class DataOutputStream

QUESTION 4
Which two code blocks correctly initialize a Locale variable?

A. Locale loc1 = "UK";
B. Locale loc2 = Locale.getInstance("ru");
C. Locale loc3 = Locale.getLocaleFactory("RU");
D. Locale loc4 = Locale.UK;
E. Locale loc5 = new Locale("ru", "RU");

Answer: E
Explanation:
D:The Locale class provides a number of convenient constants that you can use to create Locale objects for commonly used locales. For example, the following creates a Locale object for the United States:
Locale.US
E:Create a Locale object using the constructors in this class:
Locale(String language)
Locale(String language, String country)
Locale(String language, String country, String variant)
Reference: java.utilClass Locale

QUESTION 5
Which statement declares a generic class?

A. public class Example < T >{ }
B. public class <Example>{ }
C. public class Example <> { }
D. public class Example (Generic){ }
E. public class Example (G) { }
F. public class Example { }

Answer: A

QUESTION 6
Given the code fragment:

```
1. path file = path.get (args.[0])
2. try {
3. } / / statements here
4. catch (IOException) i ) {
5. }
```

And a DOS-based file system:

Which option, containing statement(s), inserted at line 3, creates the file and sets its attributes to hidden and read-only?

A. DOSFileAttributes attrs = Files.setAttribute(file,"dos:hidden","dos: readonly") Files.createFile(file, attrs)
B. Files.craeteFile(file);
   Files.setAttribute(file,"dos:hidden","dos:readonly");
C. Files.createFile(file,"dos:hidden","dos:readonly");
D. Files.createFile(file);
   Files.setAttribute(file,"dos:hidden", true);
   Files.setAttribute(file,"dos:readonly", true);

Answer: D
Explanation:
You can set a DOS attribute using the setAttribute(Path, String, Object, LinkOption...) method, as follows:

Path file = ...;
Files.setAttribute(file, "dos:hidden", true);

Note:
setAttribute
public static Path setAttribute(Path path,
String attribute,
Object value,
LinkOption... options)
throws IOException
Sets the value of a file attribute.

Reference:Interface DosFileAttribute

QUESTION 7
When using the default file system provider with a JVM running on a DOS-based file system, which statement is true?

A. DOS file attributes can be read as a set in a single method call.
B. DOS file attributes can be changed as a set in a single method call.
C. DOS file attributes can be modified for symbolic links and regular files.
D. DOS file attributes can be modified in the same method that creates the file.

Answer: A
Explanation:
File attributes associated with a file in a file system that supports legacy "DOS" attributes.
Usage Example:

Path file = ...
DosFileAttributes attrs = Files.readAttributes(file, DosFileAttributes.class);

Note:
The methodreadAttributes() reads a file's attributes as a bulk operation.

QUESTION 8
The two methods of code reuse that aggregate the features located in multiple classes are
_____ .

A. Inheritance
B. Copy and Paste
C. Composition
D. Refactoring
E. Virtual Method Invocation

Answer: AC
Explanation:
A: Inheritance is a way of reusing code and building bigger more functional objects from a basic object.
The original little object, the parent, is called the super-class. The more functional object that inherits from it is called the sub-class .
C: When your goal is code reuse, composition provides an approach that yields easier-to-change code.

QUESTION 9
Which two actions can be used in registering a JDBC 3.0 driver?

A. Add the driver class to the META-INF/services folder of the JAR file.
B. Set the driver class name by using the jdbc.drivers system property.
C. Include the JDBC driver class in a jdbcproperties file.
D. Use the java.lang.class.forName method to load the driver class.
E. Use the DriverManager.getDriver method to load the driver class.

Answer: AD
Explanation:
A:f your JDBC Driver is NOT JDBC 4-compliant then we can update the driver using"jar"utility by adding the"META-INF/services/java.sql.Driver"inside it. as following:
D:Dynamic loading of Java classes at runtime provides tremendous flexibility in the development of enterprise systems. It provides for the basis of "application servers", and allows even simpler, lighter-weight systems to accomplish some of the same ends. Within Java, dynamic-loading is typically achieved by calling the forName method on the class java.lang.Class

QUESTION 10
Which concept allows generic collections to interoperate with java code that defines collections that use raw types?

A. bytocode manipulation
B. casting
C. autoboxing
D. auto-unboxing
E. type erasure

Answer: C
Explanation:
Autoboxing, introduced in Java 5, is the automatic conversion the Java compiler makes between the primitive (basic) types and their corresponding object wrapper classes (eg, int and Integer, double and Double, etc). The underlying code that is generated is the same, but autoboxing provides a sugar coating that avoids the tedious and hard-to-read casting typically required by Java Collections, which can not be used with primitive types.

Thank You for Trying Our Product

Braindump2go Certification Exam Features:

★ More than 99,900 Satisfied Customers Worldwide.

★ Average 99.9% Success Rate.

★ Free Update to match latest and real exam scenarios.

★ Instant Download Access! No Setup required.

★ Questions & Answers are downloadable in PDF format and VCE test engine format.

★ Multi-Platform capabilities - Windows, Laptop, Mac, Android, iPhone, iPod, iPad.

★ 100% Guaranteed Success or 100% Money Back Guarantee.

★ Fast, helpful support 24x7.

View list of all certification exams: http://www.braindump2go.com/all-products.html

10% Discount Coupon Code: **BDNT2014**