1Z0-804^{Q&As}

Java SE 7 Programmer II

Pass Oracle 1Z0-804 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.passapply.com/1Z0-804.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Oracle
Official Exam Center

⚙ **Instant Download** After Purchase

⚙ **100% Money Back** Guarantee

⚙ **365 Days** Free Update

⚙ **800,000+** Satisfied Customers

QUESTION 1

Given that myFile.txt contains:

First

Second

Third

And given:

```
import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

public class ReadFile04 {

public static void main(String[] args) {

try (BufferedReader buffIn =

new BufferedReader(new FileReader("D:\\faculty\\myfile.txt"))) {

String line = "";

int count = 1;

buffIn.mark(1);

line = buffIn.readLine();

System.out.println(count + ": " + line);

line = buffIn.readLine();

count++;

System.out.println(count + ": " + line);

buffIn.reset();

line = buffIn.readLine();

count++;

System.out.println(count + ": " + line);

} catch (IOException e) {

System.out.println("IOException");

}
```

}

}

What is the result?

A. 1: First

2: Second

3: Third

B. 1 : First

2: Second

3: First

C. 1: First

2: First

3: First

D. IOExcepton

E. Compilation fails

Correct Answer: B

QUESTION 2

Given the interface:

Public interface Idgenerator {

int getNextId();

}

Which class implements IdGenerator in a thread-safe manner, so that no threads can get a duplicate id value current access?

A. Public class generator Implements IdGenerator { Private AtomicInteger id = new AtomicInteger (0); return id.incrementAndget(); } }

B. Public class Generator Implements idGenerator { private int id = 0; return ++id; } }

C. Public class Generator Implements IdGenerator { private volatile int Id = 0; return + + Id; }

D. Public class Generator Implements IdGenerator { private int id = 0; public int getNextId() { synchronized (new Generator()) {

return + + id;

}

}

}

E. Public class Generator Implements IdGenerator { private int id = 0; public int getnextId() { synchronized (id) { return + + id; } } }

Correct Answer: D

Code that is safe to call by multiple threads simultanously is called thread safe. If a piece of code is thread safe, then it contains no race conditions. Race condition only occur when multiple threads update shared resources. Therefore it is important to know what resources Java threads share when executing.

In Java you can mark a method or a block of code as synchronized. Synchronized blocks can be used to avoid race conditions.

QUESTION 3

Given the code fragment:

public class Base {

BufferedReader br;

String record;

public void process() throws FileNotFoundException {

br = new BufferedReader(new FileReader("manual.txt"));

}

}

public class Derived extends Base {

// insert code here. Line ***

public static void main(String[] args) {

try {

new Derived().process();

} catch (Exception e) { } }

}

Which code fragment inserted at line ***, enables the code to compile?

A. public void process () throws FileNotFoundException, IOException { super.process (); while ((record = br.readLine())

!= null) { System.out.println(record); } }

B. public void process () throws IOException { super.process (); while ((record = br.readLine()) != null) {
System.out.println(record); } }

C. public void process () throws Exception { super.process (); while ((record = br.readLine()) != null) {
System.out.println(record); } }

D. public void process (){ try { super.process (); while ((record = br.readLine()) != null) { System.out.println(record); } }
catch (IOException | FileNotFoundException e) { } }

E. public void process (){ try { super.process (); while ((record = br.readLine()) != null) { System.out.println(record); } }
catch (IOException e) {} }

Correct Answer: E

Incorrect answer:

D: exception java.io.FileNotFoundException has already been caught

Alternatives in a multi-catch statement cannot be related to subclassing Alternative java.io. FileNotFoundException is a
subclass of alternative java.io.IOException

QUESTION 4

An application is waiting for notification of changes to a tmp directory using the following code statements:

Path dir = Paths.get("tmp")

WatchKey key = dir.register (watcher, ENTRY_CREATE, ENTRY_DELETE, ENTRY_MODIFY) ; In the tmp directory,
the user renames the file testA to testB, Which statement is true?

A. The events received and the order of events are consistent across all platforms.

B. The events received and the order of events are consistent across all Microsoft Windows versions.

C. The events received and the order of events are consistent across all UNIX platforms.

D. The events received and the order of events are platform dependent.

Correct Answer: A

Most file system implementations have native support for file change notification. The Watch Service API takes
advantage of this support where available. However, when a file system does not support this mechanism, the Watch
Service will poll the file system, waiting for events.

Note: WatchKey : When a Watchable entity is registered with a WatchService a key which is a WatchKey is generated.
Initially the key is in ready state waiting to be notified of any events on the Watchable entity. Once an event occurs the
key goes into signaled state and allows to access the events using its pollEvents method. After processing the poll
events the key has to be reset by invoking its reset method.

Reference: The Java Tutorials, Watching a Directory for Changes

---

QUESTION 5

Which two properly implement a Singleton pattern?

A. class Singleton { private static Singleton instance; private Singleton () {} public static synchronized Singleton getInstance() { if (instance = = null) { instance = new Singleton (); } return instance; } }

B. class Singleton { private static Singleton instance = new Singleton(); protected Singleton () {} public static Singleton getInstance () { return Instance; } }

C. class Singleton { Singleton () {} private static class SingletonHolder { private static final Singleton INSTANCE = new Singleton (); } public static Singleton getInstance () { return SingletonHolder.INSTANCE; } }

D. enum Singleton { INSTANCE; }

Correct Answer: AB

A: Here the method for getting the reference to the SingleTon object is correct.

B: The constructor should be private such as:

private static Singleton instance = new Singleton();

Note: Java has several design patterns Singleton Pattern being the most commonly used.

Java Singleton pattern belongs to the family of design patterns, that govern the instantiation process. This design pattern proposes that at any time there can only be one instance of a singleton (object) created by the JVM.

The class\'s default constructor is made private, which prevents the direct instantiation of the object by others (Other Classes). A static modifier is applied to the instance method that returns the object as it then makes this method a class level method that can be accessed without creating an object.

1Z0-804 Practice Test   1Z0-804 Study Guide   1Z0-804 Braindumps

To Read the Whole Q&As, please purchase the Complete Version from Our website.

Try our product !

100% Guaranteed Success
100% Money Back Guarantee
365 Days Free Update
Instant Download After Purchase
24x7 Customer Support
Average 99.9% Success Rate
More than 800,000 Satisfied Customers Worldwide
Multi-Platform capabilities - Windows, Mac, Android, iPhone, iPod, iPad, Kindle

We provide exam PDF and VCE of Cisco, Microsoft, IBM, CompTIA, Oracle and other IT Certifications.
You can view Vendor list of All Certification Exams offered:

https://www.passapply.com/allproducts

Need Help

Please provide as much detail as possible so we can best assist you.
To update a previously submitted ticket: