



Oracle

Exam Questions 1z0-829

Java SE 17 Developer

About ExamBible

Your Partner of IT Exam

Found in 1998

ExamBible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, ExamBible has its unique advantages that other companies could not achieve.

Our Advances

* 99.9% Uptime

All examinations will be up to date.

* 24/7 Quality Support

We will provide service round the clock.

* 100% Pass Rate

Our guarantee that you will pass the exam.

* Unique Gurantee

If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

NEW QUESTION 1

Given:

```
public class Main {  
    void print(int i){  
        System.out.println("hello");  
    }  
    void print(long j){  
        System.out.println("there");  
    }  
  
    public static void main(String[] args) {  
        new Main().print(0b1101_1010);  
    }  
}
```

- A. Hello
- B. Compilation fails
- C. A NumberFormatException is thrown
- D. there

Answer: B

Explanation:

The code fragment will fail to compile because the `parseInt` method of the `Integer` class is a static method, which means that it can be invoked without creating an object of the class. However, the code is trying to invoke the `parseInt` method on an object of type `Integer`, which is not allowed. The correct way to invoke the `parseInt` method is by using the class name, such as `Integer.parseInt(s)`. Therefore, the code fragment will produce a compilation error. References: `Integer` (Java SE 17 & JDK 17) - Oracle

NEW QUESTION 2

Given the code fragment:

```
List lst = new ArrayList();  
lst.add("e1");  
lst.add("e3");  
lst.add("e2");  
  
int x1 = Collections.binarySearch(lst, "e3");  
System.out.println(x1);  
Collections.sort(lst);  
int x2 = Collections.binarySearch(lst, "e3");  
System.out.println(x2);  
  
Collections.reverse(lst);  
int x3 = Collections.binarySearch(lst, "e3");  
System.out.println(x3);
```

What is the result?

- A. 2
- B. -2
- C. 22E.111F.12-4

Answer: B

Explanation:

The code fragment uses the Collections.binarySearch method to search for the string ??e3?? in the list. The first search returns the index of the element, which is 2. The second search returns the index of the element, which is 0. The third search returns the index of the element, which is -4. The final result is 2. References: Collections (Java SE 17 & JDK 17) - Oracle

NEW QUESTION 3

Given:

```
interface IFace {
    public void m1();
    public default void m2() {
        System.out.println("m2");
    }
    public static void m3() {
        System.out.println("m3");
    }
    private void m4() {
        System.out.println("m4");
    }
}

class MyC implements IFace {
    public void m1() {
        System.out.println("Hello");
    }
}
```

Which two method invocation execute?

- A. IFace myclassobj = new Myc (); myclassObj.m3 ();
- B. Ifnce.m3 ();
- C. iFace mucloassObj = new Myc (); myClassObj.m4();
- D. new MyC() .m2 ();
- E. IFace .,4():
- F. IFace.m2();

Answer: DE

Explanation:

The code given is an interface and a class that implements the interface. The interface has three methods, m1(), m2(), and m3(). The class has one method, m1(). The only two method invocations that will execute are D and E. D is a call to the m2() method in the class, and E is a call to the m3() method in the interface.

References: https://education.oracle.com/products/trackp_OCPJSE17, 3, 4, 5

NEW QUESTION 4

Given the code fragment:

```
Stream<String> s1 = Stream.of("A", "B", "C", "B");
Stream<String> s2 = Stream.of("A", "D", "E");
Stream.concat(s1, s2).parallel().distinct().forEach(element -> System.out.print(element));
```

What is the result:

- A. ADEABCB // the order of element is unpredictable
- B. ABCE
- C. ABCDE // the order of elements is unpredictable
- D. ABBBCDE // the order of elements is unpredictable

Answer: D

Explanation:

The answer is D because the code fragment uses the Stream API to create two streams, s1 and s2, and then concatenates them using the concat() method. The resulting stream is then processed in parallel using the parallel() method, and the distinct() method is used to remove duplicate elements. Finally, the forEach() method is used to print the elements of the resulting stream to the console. Since the order of elements in a parallel stream is unpredictable, the output could be any of the options given, but option D is the most likely. References:

? Oracle Certified Professional: Java SE 17 Developer

? Java SE 17 Developer

? OCP Oracle Certified Professional Java SE 17 Developer Study Guide

? Parallelizing Streams

NEW QUESTION 5

Given the code fragment:

```
String s = "10_00";
Integer s2 = 10_00;
// Line n1
System.out.println(res);
```

Which two statements at Line n1 independently enable you to print 1250?

- A. Integer res = 250 + integer.parseInt (s)
- B. Integer res = 250 + s;
- C. Integer res = 250 + integer (s2):
- D. Integer res= 250 + s2;
- E. Integer res = 250 + integer . valueOf (s);
- F. Integer res = 250; Res = + s2;

Answer: AE

Explanation:

The code fragment is creating a string variable ??s?? with the value ??10_00?? and an integer variable ??s2?? with the value 10. The string ??s?? is using an underscore as a separator for readability, which is allowed in Java SE 17.1. The question is asking for two statements that can add 250 to the numeric value of ??s?? and assign it to an integer variable ??res??. The correct answers are A and E because they use the methods parseInt and valueOf of the Integer class to convert the string ??s?? to an integer. Both methods interpret the string as a signed decimal integer and return the equivalent int or Integer value. The other options are incorrect because they either use invalid syntax, such as B and C, or they do not convert the string ??s?? to an integer, such as D and F. References: Binary Literals (The Java™ Tutorials > Learning the Java Language > Numbers and Strings), Integer (Java SE 17 & JDK 17), Integer (Java SE 17 & JDK 17)

NEW QUESTION 6

Given the code fragment:

```
class Book {
    String author;
    String title;
    Book(String authorName, String title) {
        this.author = authorName;
        this.title = title;
    }
}

class SortBook {
    public static void main(String[] args) {
        List books = List.of(new Book("A1","T1"), new Book("A2", "T2"), new Book("A1","T2")); // Line n1
        books.sort((Book a, Book b) -> a.title.compareTo(b.title)); // Line n2
        System.out.println(books);
    }
}
```

Which action sorts the book list?

- A. At Line n2, replace books.sort() with books.stream().sort(0.
- B. At line n1, convert books type to mutable ArrayList type.
- C. At Line n1, convert type to mutable array type.
- D. At Line n2, replace compareTo () with compare ().

Answer: D

Explanation:

The code fragment is trying to sort a list of books using the `Collections.sort()` method. The correct answer is D, because the `compareTo()` method is not the correct way to compare two objects in a `Comparator`. The `compare()` method is the correct way to compare two objects in a `Comparator` and return an `int` value that indicates their order¹. The `compareTo()` method is used to implement the `Comparable` interface, which defines the natural order of objects of a class². The other options are incorrect because they either do not change the type of the list, which is already mutable, or they do not use the correct syntax for sorting a stream, which requires a terminal operation such as `collect()`³. References: `Comparator` (Java SE 17 & JDK 17), `Comparable` (Java SE 17 & JDK 17), `Stream` (Java SE 17 & JDK 17)

NEW QUESTION 7

Given:

```
class StockException extends Exception {  
    public StockException(String s) { super(s); }  
}  
class OutofStockException extends StockException {  
    public OutofStockException(String s) { super(s); }  
}
```

and the code fragment:

```
public class Test {  
    public static void main(String[] args) throws OutofStockException {  
        m();  
    }  
    public static void m() throws OutofStockException {  
        try {  
            throw new StockException("Raised.");  
        } catch (Exception e) {  
            throw new OutofStockException(e.getMessage());  
        }  
    }  
}
```

Which statement is true?

- A. The program throws `StockException`.
- B. The program fails to compile.
- C. The program throws `outofStockException`.
- D. The program throws `ClassCastException`

Answer: B

Explanation:

The answer is B because the code fragment contains a syntax error that prevents it from compiling. The code fragment tries to catch a `StockException` in line 10, but the catch block does not have a parameter of type `StockException`. The catch block should have a parameter of type `StockException`, such as:

```
catch (StockException e) { // handle the exception }
```

This is required by the Java syntax for the catch clause, which must have a parameter that is a subclass of `Throwable`. Without a parameter, the catch block is invalid and causes a compilation error.

Option A is incorrect because the program does not throw a `StockException`, as it does not compile.

Option C is incorrect because the program does not throw an `OutofStockException`, as it does not compile.

Option D is incorrect because the program does not throw a `ClassCastException`, as it does not compile. References:

? Oracle Certified Professional: Java SE 17 Developer

? Java SE 17 Developer

? OCP Oracle Certified Professional Java SE 17 Developer Study Guide

? The try-with-resources Statement (The Java™ Tutorials > Essential Classes > Exceptions)

? The catch Blocks (The Java™ Tutorials > Essential Classes > Exceptions)

NEW QUESTION 8

Given the code fragment:

```
record Product(int pNumber, String pName) {
    int regNo = 100;
    public int getRegNumber() {
        return regNo;
    }
}

public class App {
    public static void main(String[] args) {
        Product p1 = new Product (1111, "Ink Bottle");
    }
}
```

Which action enables the code to compile?

- A. Replace record with void.
- B. Remove the regNO initialization statement.
- C. Make the regNo variable static.
- D. Replace the regNo variable static
- E. Make the regNo variable public

Answer: E

Explanation:

The code will compile if the regNo variable is made public. This is because the regNo variable is being accessed in the main method of the App class, which is outside the scope of the Product class. Making the regNo variable public will allow it to be accessed from outside the class. References:
https://education.oracle.com/products/trackp_OCPJSE17, <https://mylearn.oracle.com/ou/learning-path/java-se-17-developer/99487>,
<https://docs.oracle.com/javase/tutorial/java/javaOO/accesscontrol.html>

NEW QUESTION 9

Daylight Saving Time (DST) is the practice of advancing clocks at the start of spring by one hour and adjusting them backward by one hour in autumn.

Considering that in 2021, DST in Chicago (Illinois) ended on November 7th at 2 AM, and given the fragment:

```
ZoneId zoneID = ZoneId.of("America/Chicago");
ZonedDateTime zdt = ZonedDateTime.of(
    LocalDate.of(2021, 11, 7),
    LocalTime.of(1, 30),
    zoneID
);
ZonedDateTime anHourLater = zdt.plusHours(1);
System.out.println(zdt.getHour() == anHourLater.getHour());
System.out.print(zdt.getOffset().equals(anHourLater.getOffset()));
```

What is the output?

- A. true false
- B. False false
- C. true true
- D. false true

Answer: A

Explanation:

The answer is A because the code fragment uses the ZoneId and ZonedDateTime classes to create two date-time objects with the same local date-time but different zone offsets. The ZoneId class represents a time-zone ID, such as America/Chicago, and the ZonedDateTime class represents a date-time with a time-zone in the ISO-8601 calendar system. The code fragment creates two ZonedDateTime objects with the same local date-time of 2021-11-07T01:30, but different zone IDs of America/Chicago and UTC. The code fragment then compares the two objects using the equals and isEqual methods. The equals method compares the state of two objects for equality. In this case, it compares the local date-time, zone offset, and zone ID of the two ZonedDateTime objects. Since the zone offsets and zone IDs are different, the equals method returns false. The isEqual method compares the instant of two temporal objects for equality. In this case, it compares the instant of the two ZonedDateTime objects, which is derived from the local date-time and zone offset. Since DST in Chicago ended on November 7th at 2 AM in 2021, the local date-time of 2021-11-07T01:30 in America/Chicago corresponds to the same instant as 2021-11-07T06:30 in UTC. Therefore, the isEqual method returns true. Hence, the output is true false. References:

? Oracle Certified Professional: Java SE 17 Developer
? Java SE 17 Developer
? OCP Oracle Certified Professional Java SE 17 Developer Study Guide
? Zoneld (Java Platform SE 8)
? ZonedDateTime (Java Platform SE 8)
? Time Zone & Clock Changes in Chicago, Illinois, USA
? Daylight Saving Time Changes 2023 in Chicago, USA

NEW QUESTION 10

Assume you have an automatic module from the module path display-ascii-0.2. jar. Which name is given to the automatic module based on the given JAR file?

- A. Display.ascii
- B. Display-ascii-0.2
- C. Display-ascii
- D. Display-ascii-0

Answer: C

Explanation:

An automatic module name is derived from the name of the JAR file when it does not contain a module-info.class file. If the JAR file has an `Automatic-Module-Name` attribute in its main manifest, then its value is the module name. Otherwise, the module name is derived from the JAR file's name by removing any version numbers and converting it to lower case. Therefore, for a JAR named display-ascii-0.2.jar, the automatic module name would be display-ascii, following these rules.

NEW QUESTION 10

Given the directory structure:

```
module1:
    p1\
        Doc.java
    p2\
        Util.java
```

Given the definition of the Doc class:

```
package p1;
    public sealed class Doc permits WordDoc {
}
```

Which two are valid definition of the wordDoc class?

- A. Package p1;Public non-sealed class wordDoc extends Doc ()
- B. Package p1;Public class wordDoc extends Doc ()
- C. Package p1, p2;Public non-sealed class WordDoc extends Doc ()
- D. Package p1, p2;Public sealed class WordDoc extends Doc ()
- E. Package p1,non-sealed abstract class WordDoc extends Doc ()
- F. Package p1;Public final class WordDoc extends Doc ()

Answer: AF

Explanation:

The correct answer is A and F because the wordDoc class must be a non-sealed class or a final class to extend the sealed Doc class. Option B is incorrect because the wordDoc class must be non-sealed or final. Option C is incorrect because the wordDoc class cannot be in a different package than the Doc class. Option D is incorrect because the wordDoc class cannot be a sealed class. Option E is incorrect because the wordDoc class cannot be an abstract class. References: Oracle Certified Professional: Java SE 17 Developer, 3 Sealed Classes - Oracle Help Center

NEW QUESTION 13

.....

Relate Links

100% Pass Your 1z0-829 Exam with ExamBible Prep Materials

<https://www.exambible.com/1z0-829-exam/>

Contact us

We are proud of our high-quality customer service, which serves you around the clock 24/7.

Viste - <https://www.exambible.com/>