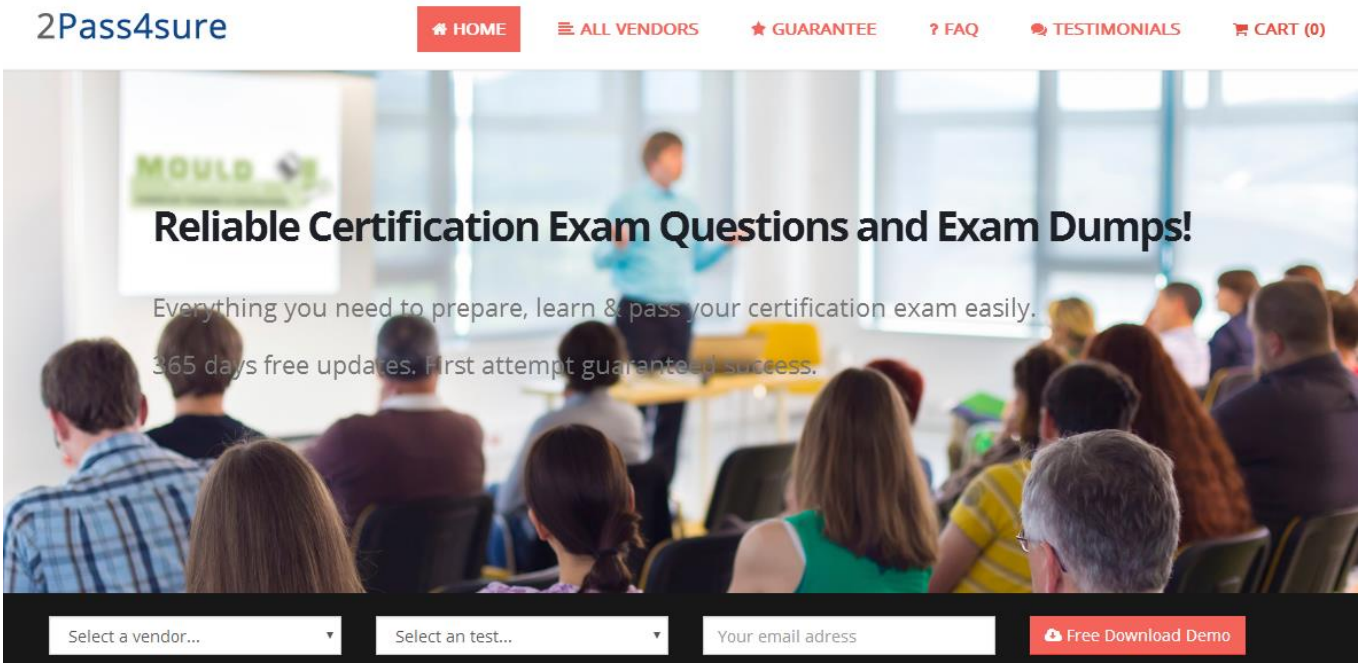
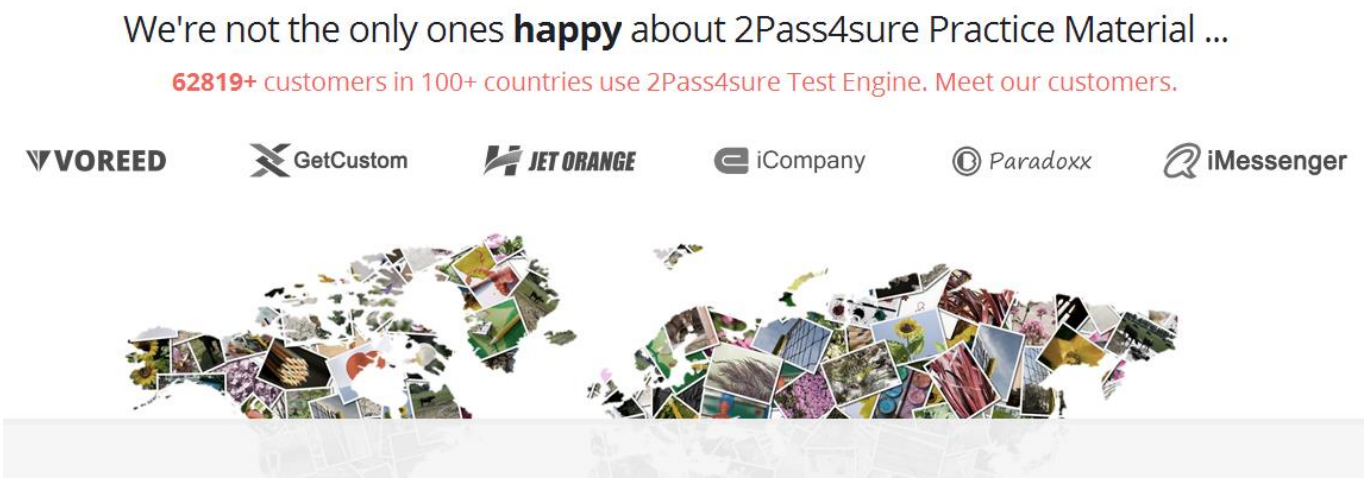
2Pass4sure





http://www.2pass4sure.com/

Reliable Certification Exam Questions and Exam Dumps - 2Pass4sure

Exam : **1z0-809-JPN**

Title : Java SE 8 Programmer II (1z0-809日本語版)

Vendor : Oracle

Version : DEMO

QUESTION NO: 1
抽象クラスの代わりにインターフェイスを使用する必要がある2つの理由はどれですか？
A. You expect that classes that implement your interfaces have many common methods or fields, or require access modifiers other than public.
B. You expect that unrelated classes would implement your interfaces.
C. You want to share code among several closely related classes.
D. You want to declare non-static on non-final fields.
E. You want to take advantage of multiple inheritance of type.
Answer: B,E

QUESTION NO: 2
Vehicle クラスの定義は次のとおりです。
Class Vehhicle {
int distance;//line n1
Vehicle (int x) {
this distance = x;
}
public void increSpeed(int time) {//line n2
int timeTravel = time;//line n3
class Car {
int value = 0;
public void speed () {
value = distance /timeTravel;
System.out.println ("Velocity with new speed"+value+"kmph");
}
}
new Car().speed();
}
}
and this code fragment:
Vehicle v = new Vehicle (100);
v.increSpeed(60);
結果はどうなりましたか？
A. 新しい速度 1 km/h の速度
B. 行 n1 でコンパイル エラーが発生します。
C. 行 n2 でコンパイル エラーが発生します。
D. 行 n3 でコンパイル エラーが発生します。
Answer: D

QUESTION NO: 3
与えられた：
class Vehicle implements Comparable<Vehicle>{
int vno;
String name;

public Vehicle (int vno, String name) {
this.vno = vno,;
this.name = name;
}
public String toString () {
return vno + ":" + name;
}
public int compareTo(Vehicle o) {
return this.name.compareTo(o.name);
}
and this code fragment:
Set<Vehicle> vehicles = new TreeSet <> ();
vehicles.add(new Vehicle (10123, "Ford"));
vehicles.add(new Vehicle (10124, "BMW"));
System.out.println(vehicles);
結果は何ですか？
A. [10123:Ford, 10124:BMW]
B. [10124:BMW, 10123:Ford]
C. A compilation error occurs.
D. A ClassCastException is thrown at run time.
Answer: A

QUESTION NO: 4
Given:

```
public class Foo<K, V> {
    private K key;
    private V value;

    public Foo(K key, V value) { this.key = key; this.value = value; }

    public static <T> Foo<T, T> twice(T value) { return new Foo<T, T>(value, value); }

    public K getKey() { return key; }
    public V getValue() { return value; }
}
```

どのオプションが失敗しますか？
A.Foo <String、Integer> mark = new Foo <String、Integer> （ "Steve"、100 ）;
B.Foo <String、String> pair = Foo。<String> twice （ "Hello World！" ）;
C.Foo <Object、Object>パーセンテージ= new Foo <String、Integer> （ "Steve"、100 ）;
D.Foo <String、String> grade = new Foo <> （ "John"、 "A" ）;
Answer: A

QUESTION NO: 5
コードの断片を考えると：
Stream<Path> files = Files.list(Paths.get(System.getProperty("user.home"))); files.forEach
(fName -> {//line n1 try { Path aPath = fName.toAbsolutePath();//line n2
System.out.println(fName + ":"
+ Files.readAttributes(aPath, Basic.File.Attributes.class).creationTime ());

```
} catch (IOException ex) {
ex.printStackTrace();
});
```

結果は何ですか？

A.ホームディレクトリの下のすべてのファイルとディレクトリが、その属性とともに一覧表示されます。

B.行n1でコンパイルエラーが発生します。

C.ホームディレクトリ内のファイルがその属性とともに一覧表示されます。

D.コンパイルエラーが行n2で発生します。

Answer: C

QUESTION NO: 6

与えられた：

```
class FuelNotAvailException extends Exception { }
class Vehicle {
void ride() throws FuelNotAvailException {//line n1
System.out.println("Happy Journey!");
}
}
class SolarVehicle extends Vehicle {
public void ride () throws FuelNotAvailException {//line n2
super ride ();
}
}
```

そしてコードの断片：

```
public static void main (String[] args) throws Exception {
Vehicle v = new SolarVehicle ();
v.ride();
}
```

どの変更により、コードフラグメントはHappy Journey !?を出力できますか？

A. Replace line n1 with public void ride() throws FuelNotAvailException {

B. Replace line n1 with protected void ride() throws Exception {

C. Replace line n2 with void ride() throws Exception {

D. Replace line n2 with private void ride() throws FuelNotAvailException {

Answer: C

QUESTION NO: 7

Given the content of /resourses/Message.properties:

```
welcome1="Good day!"
```

and given the code fragment:

```
Properties prop = new Properties ();
FileInputStream fis = new FileInputStream
("/resources/Message.properties");
prop.load(fis);
System.out.println(prop.getProperty("welcome1"));
```

System.out.println(prop.getProperty("welcome2", "Test"));//line n1
System.out.println(prop.getProperty("welcome3"));
結果は何ですか？
A. Good day!
Test
followed by an Exception stack trace
B. Good day!
followed by an Exception stack trace
C. Good day!
Test
null
D. A compilation error occurs at line n1.
Answer: C

QUESTION NO: 8
コードの断片を考えると：
List<String> listVal = Arrays.asList("Joe", "Paul", "Alice", "Tom");
System.out.println (
// line n1
）;
lineに挿入されると、コードが長さが3を超える文字列要素の数を出力できるコードフラグメントはどれですか？
A.listVal.stream（ ）。filter（x-> x.length（ ）> 3）.count（ ）
B.listVal.stream（ ）。map（x-> x.length（ ）> 3）.count（ ）
C.listVal.stream（ ）。peek（x-> x.length（ ）> 3）.count（ ）。get（ ）
D.listVal.stream（ ）。filter（x-> x.length（ ）> 3）.mapToInt（x-> x）.count（ ）
Answer: A

QUESTION NO: 9
次のコードフラグメントを考えてみましょう:
List<String> empDetails = Arrays.asList("100, Robin, HR", "200, Mary, AdminServices","101, Peter, HR"); empDetails.stream()
.filter(s-> s.contains("r"))
.sorted()
.forEach(System.out::println); //line n1
結果はどうなりましたか？
A. 100、ロビン、HR101、ピーター、HR
B. E. 行 n1 でコンパイル エラーが発生します。
C. 101、ピーター、HR200、メアリー、管理サービス
D. 100、ロビン、HR200、メアリー、AdminServices101、ピーター、HR
Answer: D

QUESTION NO: 10
どのクラス定義がコンパイルされますか?

A. public class ProductCode {
private String code;
public ProductCode(String code) {
this.code = code;
}
}
B. public class ProductCode {
private String code;
private ProductCode(String code) {
this.code = code;
}
}
C.
public class ProductCode {
private String code;
public ProductCode() {
this.code = "default";
}
}
C. public class ProductCode {
private String code;
public ProductCode(String code) {
if (code == null) {
throw new IllegalArgumentException("Code cannot be null");
}
this.code = code;
}
}
Answer: B

QUESTION NO: 11
コードの断片を考えると：
interface CourseFilter extends Predicate<String> {
public default boolean test (String str) {
return str.equals ("Java");
}
}
and
List<String> strs = Arrays.asList("Java", "Java EE", "Java ME");
Predicate<String> cf1 = s - > s.length() > 3;
Predicate cf2 = new CourseFilter() { //line n1
public boolean test (String s) {
return s.contains ("Java");
}
};

```
long c = strs.stream()
.filter(cf1)
.filter(cf2//line n2
.count();
System.out.println(c);
```

What is the result?

A. 2

B. 3

C. A compilation error occurs at line n1.

D. A compilation error occurs at line n2.

Answer: B

QUESTION NO: 12

アプリケーションのローカライズに関して当てはまる2つのステートメントはどれですか？

A. 新しい地域言語のサポートには、コードの再コンパイルは必要ありません。

B. テキスト要素（メッセージとGUIラベル）はコードにハードコーディングされています。

C.
言語および地域固有のプログラムは、ローカライズされたデータを使用して作成されます。

D. リソースバンドルファイルには、データと通貨情報が含まれます。

E. 言語コードは小文字を使用し、地域コードは大文字を使用します。

Answer: A,E

References:

QUESTION NO: 13

Given:

```
interface Interface1 {
    public default void sayHi() {
        System.out.println("Hi Interface-1");
    }
}

interface Interface2 {
    public default void sayHi() {
        System.out.println("Hi Interface-2");
    }
}
public class MyClass implements Interface1, Interface2 {
    public static void main(String[] args) {
        Interface1 obj = new MyClass();
        obj.sayHi();
    }
    public void sayHi() {
        System.out.println("Hi MyClass");
    }
}
```

結果は何ですか？

A. Hi Interface-2

B. A compilation error occurs.

C. Hi Interface-1

D. Hi MyClass

Answer: D

QUESTION NO: 14

コードの断片を考えると：

Path source = Paths.get ("/data/december/log.txt");
Path destination = Paths.get("/data");
Files.copy (source, destination);
and assuming that the file /data/december/log.txt is accessible and
contains:
10-Dec-2014 ?Executed successfully
What is the result?

A. A file with the name log.txt is created in the /data directory and the content of the /data/december/log.txt file is copied to it.

B. The program executes successfully and does NOT change the file system.

C. A FileNotFoundException is thrown at run time.

D. A FileAlreadyExistsException is thrown at run time.

Answer: D

QUESTION NO: 15

Given:
class Book {
int id;
String name;
public Book (int id, String name) {
this.id = id;
this.name = name;
}
public boolean equals (Object obj) { //line n1
boolean output = false;
Book b = (Book) obj;
if (this.name.equals(b name))}
output = true;
}
return output;
}
}
and the code fragment:
Book b1 = new Book (101, "Java Programing");
Book b2 = new Book (102, "Java Programing");
System.out.println (b1.equals(b2)); //line n2
どちらの説明が正しいですか？

A. The program prints true.

B. The program prints false.

C. A compilation error occurs. To ensure successful compilation, replace line n1 with:
boolean equals (Book obj) {

D. A compilation error occurs. To ensure successful compilation, replace line n2 with:
System.out.println (b1.equals((Object) b2));

Answer: A

QUESTION NO: 16
Given that version.txtis accessible and contains:
1234567890
and given the code fragment:

```
try (FileInputStream fis = new FileInputStream("version.txt");
        InputStreamReader isr = new InputStreamReader(fis);
        BufferedReader br = new BufferedReader(isr);) {
    if (br.markSupported()) {
        System.out.print((char) br.read());
        br.mark(2);
        System.out.print((char) br.read());
        br.reset();
        System.out.print((char) br.read());
    }
} catch (Exception e) {
    e.printStackTrace();
}
```

What is the result?

A. 121

B. 122

C. 135

D. The program prints nothing.

Answer: B

QUESTION NO: 17

与えられた：

```
public class Foo<K, V> {
    private K key;
    private V value;

    public Foo(K key, V value) { this.key = key; this.value = value; }

    public static <T> Foo<T, T> twice(T value) { return new Foo<T, T>(value, value); }

    public K getKey() { return key; }
    public V getValue() { return value; }
}
```

どのオプションが失敗しますか？

A. Foo<String, Integer> mark = new Foo<Object, Object> ("Steve", 100);

B. Foo<String, String> pair = Foo.<String>twice ("Hello World!");

C. Foo<Object, Object> percentage = new Foo<Object, Object>("Steve", 100);

D. Foo<String, String> grade = new Foo <> ("John", "A");

Answer: C

QUESTION NO: 18

与えられた：

```
class Student {
String course, name, city;
public Student (String name, String course, String city) {
this.course = course; this.name = name; this.city = city;
}
public String toString() {
return course + ":" + name + ":" + city;
```

```
}
public String getCourse() {return course;}
public String getName() {return name;}
public String getCity() {return city;}
and the code fragment:
List<Student> stds = Arrays.asList(
new Student ("Jessy", "Java ME", "Chicago"),
new Student ("Helen", "Java EE", "Houston"),
new Student ("Mark", "Java ME", "Chicago"));
stds.stream()
.collect(Collectors.groupingBy(Student::getCourse))
.forEach(src, res) -> System.out.println(res));
```

結果は何ですか？

A. A compilation error occurs.

B. Java EEJava ME

C. [Java EE: Helen:Houston][Java ME: Jessy:Chicago, Java ME: Mark:Chicago]

D. [Java ME: Jessy:Chicago, Java ME: Mark:Chicago][Java EE: Helen:Houston]

Answer: B

QUESTION NO: 19

Given:

Book.java:

```
public class Book {
private String read(String bname) { return "Read" + bname }
}
```

EBook.java:

```
public class EBook extends Book {
public class String read (String url) { return "View" + url }
}
```

Test.java:

```
public class Test {
public static void main (String[] args) {
Book b1 = new Book();
b1.read("Java Programing");
Book b2 = new EBook();
b2.read("http://ebook.com/ebook");
}
}
```

結果は何ですか？

A. Read Java Programming
View http:/ ebook.com/ebook

B. Read Java Programming
Read http:/ ebook.com/ebook

C. The EBook.java file fails to compile.

D. The Test.java file fails to compile.

Answer: D

QUESTION NO: 20
コードの断片を考えると：

```
try {
    Properties prop = new Properties();
    prop.put("user", userName);
    prop.put("password", passWord);
    Connection conn = DriverManager.getConnection(dbURL, prop);
    if(conn != null){
        System.out.print("Connection Established");
    }
} catch (Exception e) {
    System.out.print(e);
}
```

and the information:
- The required database driver is configured in the classpath.
- The appropriate database is accessible with the dbURL, username, and passWord exists.
結果は何ですか？

A. A ClassNotFoundException is thrown at runtime.

B. The program prints nothing.

C. The program prints Connection Established.

D. A SQLException is thrown at runtime.

Answer: C

QUESTION NO: 21
Given:
class Vehicle {
int vno;
String name;
public Vehicle (int vno, String name) {
this.vno = vno,;
this.name = name;
}
public String toString () {
return vno + ":" + name;
}
}
and this code fragment:
Set<Vehicle> vehicles = new TreeSet <> ();
vehicles.add(new Vehicle (10123, "Ford"));
vehicles.add(new Vehicle (10124, "BMW"));
System.out.println(vehicles);
結果は何ですか？

A. 10123 Ford

10124 BMW
B. 10124 BMW
10123 Ford
C. A compilation error occurs.
D. A ClassCastException is thrown at run time.
Answer: D