

オラクル認定資格教科書 Java プログラマ Gold SE 11
(試験番号 1Z0-816)

模擬試験

Oracle Certified Java Programmer, Gold SE 11

模擬試験

問題 1

次のコード（抜粋）があります。

```
3.    Boolean a = true;
4.    var b = a.booleanValue();
5.    long c = 3_000L;
6.    Long d = c;
7.    d = null;
8.    c = d;
```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. 4行目でコンパイルエラー
- ☐ B. 7行目でコンパイルエラー
- ☐ C. 実行時に `NumberFormatException` 例外が発生する
- ☐ D. 実行時に `NullPointerException` 例外が発生する
- ☐ E. コンパイル、実行ともに成功する

問題 2

`equals()` メソッドと `hashCode()` メソッドの実装ルールとして、正しいものは次のどれですか。2つ選択してください。

- ☐ A. `s1.equals(s2)` が `true` の場合、`s1.hashCode() == s2.hashCode()` は、常に `true` となる
- ☐ B. `s1.equals(s2)` が `true` の場合、`s1.hashCode() == s2.hashCode()` は、`true`、`false` のいずれかとなる
- ☐ C. `s1.equals(s2)` が `false` の場合、`s1.hashCode() == s2.hashCode()` は、常に `true` となる
- ☐ D. `s1.equals(s2)` が `false` の場合、`s1.hashCode() == s2.hashCode()` は、`true`、`false` のいずれかとなる

問題 3

次のコードがあります。

```
1. public class Test {
2.     private int num = 100;
3.     public void show() {
4.         String str = "num : ";
5.         class Foo {
6.             private int num = Test.this.num;
7.             public void show() {
8.                 num += 100;
9.                 System.out.println(str + num);
10.            }
11.        }
12.        new Foo().show();
13.    }
14.    public static void main(String[] args) {
15.        new Test().show();
16.    }
17. }
```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- | | |
|---|--|
| <input type="radio"/> A. num : 100 | <input type="radio"/> B. num : 200 |
| <input type="radio"/> C. 6 行目でコンパイルエラー | <input type="radio"/> D. 7 行目でコンパイルエラー |
| <input type="radio"/> E. 12 行目でコンパイルエラー | <input type="radio"/> F. 実行時エラー |

問題 4

次のコードがあります。

```
1. public enum Vals {
2.     VAL1(false), VAL2(false),
3.     VAL3(Boolean.FALSE), VAL4(Boolean.FALSE);
4.     private boolean data;
5.     public Vals(boolean data) { this.data = data; }
6.     public boolean getData() { return data; }
7.     public void setData() { this.data = data; }
8.
9.     public static void main(String[] args) {
```

```

10.    System.out.print(Vals.VAL3 + " ");
11.    System.out.println(Vals.VAL3.getData());
12.    }
13.    }

```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. 3行目でコンパイルエラー
- ☐ B. 5行目でコンパイルエラー
- ☐ C. 6、7行目でコンパイルエラー
- ☐ D. 実行時エラー
- ☐ E. VAL3 false
- ☐ F. VAL3 FALSE

問題 5

次のコードがあります。

```

1. public class Foo {
2.     static class X {
3.         public void method() { System.out.println("X"); } }
4.     static class Y extends X {
5.         public void method() { System.out.println("Y"); } }
6.     static class Z extends X {
7.         public void method() { System.out.println("Z"); } }
8.     public static void main(String[] args) {
9.         X obj1 = new Y();
10.        Z obj2 = (Z)obj1;
11.        obj2.method();
12.    }
13. }

```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. X
- ☐ B. Y
- ☐ C. Z
- ☐ D. 9行目でコンパイルエラー
- ☐ E. 10行目でコンパイルエラー
- ☐ F. 実行時エラー

問題 6

インタフェースの説明として正しいものは次のどれですか。2つ選択してください。

- ☐ A. static な抽象メソッドを定義できる

- ☐ B. private である static な具象メソッドを定義できる
- ☐ C. private である抽象メソッドを定義できる
- ☐ D. private であるデフォルトメソッドを定義できる
- ☐ E. private である非 static な具象メソッドを定義できる
- ☐ F. public である非 static な具象メソッドを定義できる

問題 7

次のコードがあります。

```

1. interface X {
2.     default void methodA() { System.out.println("X"); }
3. }
4. interface Y {
5.     default void methodA() { System.out.println("Y"); }
6.     public abstract void methodB();
7. }
8. interface Z extends X, Y {
9.     void methodC(int a, int b);
10. }
```

説明として正しいものは次のどれですか。1つ選択してください。

- ☐ A. 5 行目でコンパイルエラー
- ☐ B. 6 行目でコンパイルエラー
- ☐ C. 8 行目でコンパイルエラー
- ☐ D. 9 行目でコンパイルエラー
- ☐ E. コンパイルは成功する

問題 8

次のコードがあります。

```

1. public abstract class Test {
2.     String s1;
3.     public abstract void foo();
4.     public static void main(String[] args) {
5.         Test obj = new ExTest();
6.         obj.s1 = "hello"; obj.foo();
7.     }
8.     public static class ExTest extends Test {
```

```

9.     void foo() { System.out.println(s1); }
10.  }
11.  }

```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. 1行目でコンパイルエラー
- ☐ B. 3行目でコンパイルエラー
- ☐ C. 5行目でコンパイルエラー
- ☐ D. 8行目でコンパイルエラー
- ☐ E. 9行目でコンパイルエラー
- ☐ F. hello

問題 9

次のコードがあります。

```

1. public interface Foo {
2.     default void method() { }
3. }
4. class MyFoo implements Foo {
5.     【 ① 】
6. }

```

①に挿入するコードとして正しいものは次のどれですか。3つ選択してください。

- ☐ A. @Override public boolean equals(Object o) { return false; }
- ☐ B. @Override public boolean equals(MyFoo o) { return false; }
- ☐ C. @Override public int hashCode() { return 55; }
- ☐ D. @Override public long hashCode() { return 55; }
- ☐ E. @Override void method() { }
- ☐ F. @Override public void method() { }

問題 10

インタフェース宣言として有効なコードは次のどれですか。2つ選択してください。

- ☐ A. @FunctionalInterface
interface Test<T> {
 default void equals() {
 //more code

- ```

 }
 void task(T t);
}

```
- ☐ B. @FunctionalInterface
- ```

interface Test<T> {
    public int bar() throws RuntimeException;
    public String toString() {
        return "";
    }
}

```
- ☐ C. @FunctionalInterface
- ```

interface Test<T> {
 public static void foo() {
 //more code
 }
 public boolean equals(Object obj);
}

```
- ☐ D. @FunctionalInterface
- ```

interface Test<T> {
    public void foo();
    public void foo(int a);
}

```
- ☐ E. @FunctionalInterface
- ```

interface Test<T> {
 public void foo();
 private void bar() {
 //more code
 }
}

```
- ☐ F. @FunctionalInterface
- ```

interface Test<T> {
    public static void foo();
    private void bar(char[] c) {}
}

```

問題 11

次のコード（抜粋）があります。

```
3. class Item {
4.     //more code
5. }
6. public class Test {
7.     @SafeVarargs
8.     void order(List<Item>... items) {
9.         if(items != null){
10.             System.out.println(items.length);
11.         }
12.     }
13. }
```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. 7行目でコンパイルエラー
- ☐ B. 8行目でコンパイルエラー
- ☐ C. 10行目でコンパイルエラー
- ☐ D. 8行目で実行時エラー
- ☐ E. 10行目で実行時エラー

問題 12

次のコードがあります。

```
1. import java.lang.annotation.*;
2. import static java.lang.annotation.ElementType.*;
3.
4. @Target({TYPE, METHOD, PARAMETER})
5. public @interface MyAnnot {
6. }
```

MyAnnot アノテーションが使用できる箇所は次のどれですか。2つ選択してください。

- ☐ A. クラス
- ☐ B. インタフェース
- ☐ C. インスタンス変数
- ☐ D. コンストラクタ
- ☐ E. ローカル変数

問題 13

次のコードがあります。

```
1. import java.io.*;
2. public class Test {
3.     static class Foo implements AutoCloseable {
4.         public void close() {
5.             System.out.print("A");
6.             throw new RuntimeException(); } }
7.     static class Bar implements Closeable {
8.         public void close() {
9.             System.out.print("B");
10.            throw new RuntimeException(); } }
11.    public static void main(String[] args) {
12.        try {
13.            Foo foo = new Foo();
14.            Bar bar = new Bar();
15.        } catch (Exception e) { System.out.print("C"); }
16.        } finally { System.out.print("D"); }
17.    }
18. }
```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- | | |
|-------------------------------|-------------------------------|
| <input type="radio"/> A. ABCD | <input type="radio"/> B. BACD |
| <input type="radio"/> C. ABC | <input type="radio"/> D. ABD |
| <input type="radio"/> E. C | <input type="radio"/> F. D |

問題 14

次のコード（抜粋）があります。

```
3. public static void main(String[] args) throws BException {
4.     try{
5.         throw new BException();
6.     } catch(AException | RuntimeException e) {
7.         【 ① 】
8.         throw e;
9.     }
10. }
```

11. static class AException extends Exception { }
12. static class BException extends AException { }

コードを正常にコンパイルするために、①に挿入するコードとして正しいものは次のどれですか。1つ選択してください。

- ☐ A. e = new AException();
- ☐ B. e = new BException();
- ☐ C. e = new Exception();
- ☐ D. e = new RuntimeException();
- ☐ E. 何も記述しない

問題 15

次のコード（抜粋）があります。

```

3. public static void main(String[] args) {
4.     try(Foo obj1 = new Foo(); Foo obj2 = new Foo()) {
5.         throw new RuntimeException("main");
6.     } catch(Exception e) {
7.         System.out.print(e.getMessage() + " " +
8.                         e.getSuppressed().length);
9.     }}
10. static class Foo implements AutoCloseable {
11.     public void close() {
12.         throw new RuntimeException("close");
13.     }}

```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. close 0
- ☐ B. close 1
- ☐ C. close 2
- ☐ D. main 0
- ☐ E. main 1
- ☐ F. main 2
- ☐ G. コンパイルエラー

問題 16

説明として正しいものは次のどれですか。2つ選択してください。

- ☐ A. AutoCloseable インタフェースは java.io パッケージで提供されている
- ☐ B. Closeable インタフェースは java.lang パッケージで提供されている
- ☐ C. AutoCloseable および Closeable は java.lang パッケージで提供されている
- ☐ D. AutoCloseable および Closeable は java.io パッケージで提供されている
- ☐ E. AutoCloseable の close() メソッドは throws Exception と宣言されている
- ☐ F. Closeable の close() メソッドは throws IOException と宣言されている
- ☐ G. AutoCloseable および Closeable の close() メソッドは throws Throwable と宣言されている

問題 17

次のコード（抜粋）があります。

```
6. private int method(int x, int y) {  
7.     boolean assert = false;  
8.     assert ++x > 0;  
9.     assert y > 0;  
10.    return x + y;  
11. }
```

説明として正しいものは次のどれですか。2つ選択してください。

- ☐ A. 7行目でコンパイルエラーとなる
- ☐ B. 8行目はアサーションの適切な使い方である
- ☐ C. 9行目はアサーションの適切な使い方である
- ☐ D. 8、9行目で boolean 式の後にメッセージが指定されていないためコンパイルエラーとなる
- ☐ E. 8、9行目で boolean 式が () で囲まれていないためコンパイルエラーとなる

問題 18

次のコードがあります。

```

1. public class Test {
2.     public static void main(String[] args) {
3.         Integer num = 100;
4.         num++;
5.         assert num != null && num <= 100;
6.         System.out.println(num);
7.     }
8. }

```

コンパイル、実行は以下とします。

```

>javac Test.java
(コンパイル結果は省略)
>java Test
(実行結果は省略)

```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. 4行目でコンパイルエラー
- ☐ B. 5行目でコンパイルエラー
- ☐ C. 5行目で実行時に `AssertionError` が発生する
- ☐ D. 5行目で実行時に他の例外クラスが発生する
- ☐ E. 100
- ☐ F. 101

問題 19

次のコードがあります。

```

1. import java.util.List;
2. class Foo {
3.     String str;
4.     private final List<String> list;
5.     public Foo(String str, List<String> list) {
6.         this.str = str;
7.         this.list = list;
8.     }
9.     public String getStr() { return str; }
10.    public List<String> getList() { return list; }
11. }

```

`Foo` クラスをイミュータブルオブジェクトとして定義するには、説明として正しいもの

は次のどれですか。4つ選択してください。

- ☐ A. Foo クラスを final クラスとする
- ☐ B. コンストラクタの定義内で、list 変数には List<String> のコピーを作成し、代入する
- ☐ C. 4 行目について、List<String> 型を List<Object> 型に変更する
- ☐ D. 10 行目について、参照経由で List<String> に直接アクセスされない実装方法に変更する
- ☐ E. str 変数は private、final の修飾子を付与する
- ☐ F. Foo クラスを Immutable インタフェースを実装したクラスとする

問題 20

次のコード（抜粋）があります。

```

4. Deque<String> deque = new ArrayDeque<>();
5. deque.push("A"); deque.push("B"); deque.push("C");
6. deque.pop();
7. deque.peek();
8. while(deque.remove() != null) {
9.     System.out.print(deque.pop());
10. }
```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. A
- ☐ B. BC
- ☐ C. ABC
- ☐ D. C
- ☐ E. BA
- ☐ F. CBA
- ☐ G. コンパイルエラー
- ☐ H. 実行時エラー

問題 21

コンパイルが成功するコードとして正しいものは次のどれですか。3つ選択してください。

- ☐ A. List<Integer> obj = new Vector<Integer>();
- ☐ B. List<Object> obj = new HashSet<Object>();
- ☐ C. List<Object> obj = new LinkedList<? extends Object>();

- ☐ D. `HashSet<Number> obj = new HashSet<Double>();`
- ☐ E. `HashSet<? super ArithmeticException> obj =
new HashSet<Exception>();`
- ☐ F. `Map<String, ? extends Number> obj =
new HashMap<String, Double>();`

問題 22

次のコード（抜粋）があります。

- ```
4. Map<String, Double> map = new HashMap<>();
5. 【 ① 】
```

コンパイル、実行がともに成功する①に挿入可能なコードとして正しいものは次のどれですか。1つ選択してください。

- ☐ A. `map.put(10.0);`
- ☐ B. `map.add("val", 10L);`
- ☐ C. `map.put(new Double(0.0));`
- ☐ D. `map.add("val", 0.0);`
- ☐ E. A～Dのいずれも正しくない

## 問題 23

次のコード（抜粋）があります。

- ```
3. public static void main(String[] args) {
4.   【 ① 】
5. }
6. static <T extends Exception> void foo(T t) {
7.   System.out.println(t.getMessage());
8. }
```

コンパイル、実行がともに成功する①に挿入可能なコードとして正しいものは次のどれですか。3つ選択してください。

- ☐ A. `Test.<NullPointerException>foo(new NullPointerException
("ex"));`
- ☐ B. `Test.<Throwable>foo(new Exception("ex"));`

- ☐ C. `Test.foo(new Throwable("ex"));`
- ☐ D. `Test.foo(new IOException("ex"));`
- ☐ E. `Test.foo(new Exception("ex"));`

問題 24

Comparatorの実装として正しいものは次のどれですか。2つ選択してください。

- ☐ A.

```
Comparator comp = new Comparator<?>() {
    public int compare(Integer x, Integer y) {
        return x.compareTo(y);
    }
};
```
- ☐ B.

```
Comparator<Integer> comp = new Comparator<>() {
    public int compare(Integer x, Integer y) {
        return x.compareTo(y);
    }
};
```
- ☐ C.

```
Comparator<> comp = new Comparator<Integer>() {
    public int compare(Integer x, Integer y) {
        return x.compareTo(y);
    }
};
```
- ☐ D.

```
Comparator<Integer> comp = (x, y) -> Integer.compare(x, y);
```
- ☐ E.

```
var comp = (x, y) -> Integer.compare(x, y);
```

問題 25

次のコードがあります。

```
1. import java.util.*;
2. class Foo implements Comparable<Foo>, Comparator<Foo> {
3.     private String val1;
4.     private String val2;
5.     Foo(String val1, String val2) {
6.         this.val1 = val1;
```

```

7.     this.val2 = val2;
8. }
9. public String toString() {
10.     return val1; }
11. public int compareTo(Foo obj) {
12.     return val2.compareTo(obj.val2); }
13. public int compare(Foo obj1, Foo obj2) {
14.     int num = Integer.parseInt(obj1.val1) -
15.             Integer.parseInt(obj2.val1) ;
16.     return num; }
17. }
18. public class Test {
19.     public static void main(String[] args) {
20.         Foo obj1 = new Foo("60", "x");
21.         Foo obj2 = new Foo("10", "y");
22.         TreeSet<Foo> set1 = new TreeSet<>();
23.         set1.add(obj1); set1.add(obj2);
24.         TreeSet<Foo> set2 = new TreeSet<>(obj1);
25.         set2.add(obj1); set2.add(obj2);
26.         System.out.println(set1 + " " + set2);
27.     }
28. }

```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- | | |
|--|--|
| <input type="radio"/> A. [60, 10] [10, 60] | <input type="radio"/> B. [10, 60] [60, 10] |
| <input type="radio"/> C. [60, 10] [60, 10] | <input type="radio"/> D. [10, 60] [10, 60] |
| <input type="radio"/> E. コンパイルエラー | <input type="radio"/> F. 実行時エラー |

問題 26

次のコード（抜粋）があります。

```

4. Comparator<Integer> obj = (v1, v2) -> v2 - v1;
5. List<Integer> list = Arrays.asList(10, 2, 30, 2);
6. Collections.sort(list, obj);
7. System.out.println(Collections.binarySearch(list, 2));

```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- | | |
|----------------------------|----------------------------|
| <input type="radio"/> A. 0 | <input type="radio"/> B. 1 |
|----------------------------|----------------------------|

- ☐ C. 3
- ☐ D. 検索結果の正しさは保証されない
- ☐ E. コンパイルエラー
- ☐ F. 実行時エラー

問題 27

次のコードがあります。

```

1. class Foo 【 ① 】 { }
2. public class Test {
3.     public static void main(String[] args) {
4.         Foo<Integer> obj1 = new Foo 【 ② 】 ();
5.         Foo<Object> obj2 = new Foo();
6.     }
7. }
```

コンパイル、実行がともに成功する①、②に挿入可能なコードとして正しいものは次のどれですか。2つ選択してください。

- | | |
|-----------------------------------|-----------------------------------|
| <input type="checkbox"/> A. ①に<?> | <input type="checkbox"/> B. ①に<T> |
| <input type="checkbox"/> C. ①に<> | <input type="checkbox"/> D. ②に<?> |
| <input type="checkbox"/> E. ②に<T> | <input type="checkbox"/> F. ②に<> |

問題 28

次のコード（抜粋）があります。

```
5. public static T method(T t) { return t; }
```

コードを正常にコンパイルするために、説明として正しいものは次のどれですか。1つ選択してください。

- ☐ A. public の後に<T>を追加する
- ☐ B. static の後に<T>を追加する
- ☐ C. T の後に<T>を追加する
- ☐ D. public もしくは static の後に<T>を追加する
- ☐ E. static もしくは T の後に<T>を追加する
- ☐ F. 修正の必要はなく、現状のコードでコンパイルは成功する

問題 29

次のコード（抜粋）があります。

```
5.    var myList = Arrays.asList(2.0, 3.0, 4.0);
6.    UnaryOperator<Double> u = 【 ① 】 ;
7.    myList.replaceAll(u);
8.    System.out.println(myList);
```

myList の要素を 2 倍とする UnaryOperator の実装として①に挿入する正しいコードは次のどれですか。1 つ選択してください。

- ☐ A. `x -> y * 2`
- ☐ B. `x -> { return x * 2 }`
- ☐ C. `(var v) -> (v * 2)`
- ☐ D. `(double y) -> y * 2`
- ☐ E. `var y -> { return y * 2; }`

問題 30

次のコード（抜粋）があります。

```
4. interface Foo<T, R> {
5.     public R exec(T t);
6. }
7. public class Test {
8.     public static void main(String[] args) {
9.         double num = 1000.0;
10.        Foo<Integer, Double> obj = x -> x + num;
11.        Double ans = obj.exec(2000);
12.        System.out.println(ans);
13.    }
14. }
```

4～6 行目の Foo インタフェースを、`java.util.function` パッケージで提供されている関数型インタフェースを使用したコードに置き換えたいと考えています。適切なインタフェースは次のどれですか。1 つ選択してください。

- ☐ A. `Consumer`
- ☐ B. `Predicate`

- ☐ C. Supplier
- ☐ D. Function
- ☐ E. BinaryOperator

問題 31

次のコード（抜粋）があります。

```
6.    DoubleFunction f = i -> i * 0.1;
7.    Double ans = f.apply(100);
```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. 10
- ☐ B. 10.0
- ☐ C. コンパイル、実行ともに成功するが何も出力されない
- ☐ D. コンパイルエラー
- ☐ E. 実行時エラー

問題 32

次のコード（抜粋）があります。

```
6.    int val = 0;
7.    for(int i = 1; i <= 10; i++){
8.        val += i;
9.    }
10.   System.out.println(val);
```

7～9行目を置き換えた際に同じ実行結果を得られるコードは次のどれですか。1つ選択してください。

- ☐ A. `val = IntStream.range(0, 10).sum();`
- ☐ B. `val = IntStream.rangeClosed(0, 11)`
`.skip(1)`
`.limit(10)`
`.reduce(0, (a, b) -> a + b)`
- ☐ C. `val = IntStream.range(0, 11)`
`.peek(a -> val += a)`
`.sum();`
- ☐ D. `IntStream.rangeClosed(1, 10)`

```
.forEach(n -> n += val)
```

問題 33

次のコード（抜粋）があります。

```
4. Stream<String> stream = Stream.iterate("", s -> s + "a");  
5. System.out.println(stream.limit(2).map(x -> x + "x"));
```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. xax
- ☐ B. axax
- ☐ C. xaxaax
- ☐ D. java.util.stream.ReferencePipeline\$3@1218025c
- ☐ E. コンパイルエラー
- ☐ F. 実行時エラー

問題 34

次のコード（抜粋）があります。

```
5. Predicate<? super String> f = s -> s.length() > 3;  
6. Stream<String> stream = Stream.iterate("x", s -> s + s);  
7. boolean a = stream.noneMatch(f);  
8. boolean b = stream.anyMatch(f);  
9. System.out.println(a + " " + b);
```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. false true
- ☐ B. false false
- ☐ C. true true
- ☐ D. コンパイルエラー
- ☐ E. 実行時エラー
- ☐ F. コンパイルは成功するが、実行するとハングする

問題 35

リダクション操作をサポートする終端操作メソッドとして正しいものは次のどれですか。2つ選択してください。

- | | |
|--|---------------------------------------|
| <input type="checkbox"/> A. findAny() | <input type="checkbox"/> B. sorted() |
| <input type="checkbox"/> C. distinct() | <input type="checkbox"/> D. collect() |
| <input type="checkbox"/> E. count() | <input type="checkbox"/> F. toArray() |

問題 36

次のコード（抜粋）があります。

```
4. Stream<String> stream = Stream.of(" ");
5. boolean x = stream.【 ① 】(String::isEmpty);
6. System.out.println(x);
```

コンパイル、実行がともに成功し、出力結果が true となるために①に入るメソッド名として正しいものは次のどれですか。1 つ選択してください。

- | | |
|------------------------------------|-----------------------------------|
| <input type="radio"/> A. noneMatch | <input type="radio"/> B. allMatch |
| <input type="radio"/> C. anyMatch | <input type="radio"/> D. findAny |
| <input type="radio"/> E. findFirst | |

問題 37

次のコード（抜粋）があります。

```
10. static List<String> mySort(List<String> list) {
11.     List<String> copyList = new ArrayList<>(list);
12.     Collections.sort(copyList, (x, y) -> y.compareTo(x));
13.     return copyList;
14. }
```

11～13 行目と同様の結果を得るコードとして正しいものは次のどれですか。1 つ選択してください。

- | | |
|---|--------------------------------------|
| <input type="radio"/> A. return list.stream() | |
| | .compareTo((x, y) -> y.compareTo(x)) |
| | .sort(); |
| <input type="radio"/> B. return list.stream() | |
| | .sorted((x, y) -> y.compareTo(x)) |
| | .collect(); |

- C. `return list.stream()
 .sorted((x, y) -> y.compareTo(x))
 .collect(Collectors.toList());`
- D. `return list.stream()
 .compare((x, y) -> y.compareTo(x))
 .collect(Collectors.toList());`
- E. `return list.stream()
 .compare((x, y) -> y.compareTo(x))
 .sort();`
- F. `return list.stream()
 .compareTo((x, y) -> y.compareTo(x))
 .collect(Collectors.toList());`

問題 38

次のコード（抜粋）があります。

```
4. Stream.generate(() -> "hello")
5.     .limit(2)
6.     .filter(s -> s.length() > 3)
7.     .peek(x -> System.out.println(x + " "))
8.     .forEach(x -> System.out.println(x + " "));
```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- A. `hel` が 2 回出力される
- B. `hello` が 2 回出力される
- C. `hello` が 4 回出力される
- D. コンパイルエラー
- E. 実行時エラー
- F. コンパイルは成功するが、実行するとハングする

問題 39

次のコード（抜粋）があります。

```
4. Stream<Integer> stream1 = Stream.of(10);
5. IntStream stream2 = stream1.mapToInt(a -> a);
6. Stream<Integer> stream3 = Stream.of(10);
```

```

7. DoubleStream stream4 = stream3.mapToDouble(a -> a);
8. Stream<Integer> stream5 = stream4.mapToInt(a -> a);

```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. 4行目でコンパイルエラー
- ☐ B. 5行目でコンパイルエラー
- ☐ C. 7行目でコンパイルエラー
- ☐ D. 8行目でコンパイルエラー
- ☐ E. コンパイルは成功するが、実行時エラー
- ☐ F. コンパイル、実行ともに成功する

問題 40

次のコード（抜粋）があります。

```

5. Stream<String> stream1 = Stream.of("a", "b", "ax");
6. Stream<String> stream2 = Stream.empty();
7. Map<Boolean, List<String>> map1 =
8.   stream1.collect(Collectors.partitioningBy(s -> ➡
      s.startsWith("x")));
9. Map<Boolean, List<String>> map2 =
10.  stream2.collect(Collectors.groupingBy(s -> s.startsWith("x")));
11. System.out.println(map1 + " " + map2);

```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. {false=[a, b, ax], true=[]} {false=[], true=[]}
- ☐ B. {false=[a, b, ax]} {false=[], true=[]}
- ☐ C. {false=[a, b, ax], true=[]} {}
- ☐ D. {false=[a, b, ax]} {}
- ☐ E. {} {}
- ☐ F. コンパイルエラー

問題 41

次のコード（抜粋）があります。

```

4. DoubleStream stream = DoubleStream.of(0.3, 0.8);
5. stream.peek(System.out::println).filter(a -> a > 0.5).count();

```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. 1 が出力される
- ☐ B. 2 が出力される
- ☐ C. 0.3 が出力される
- ☐ D. 0.8 が出力される
- ☐ E. 0.3 と 0.8 が出力される
- ☐ F. 0.3 と 0.8 と 1 が出力される
- ☐ G. 0.3 と 0.8 と 2 が出力される

問題 42

次のコード（抜粋）があります。

```

4. public static void main(String[] args) {
5.     【 ① 】
6. }
7. public static void foo(Stream<Integer> stream) {
8.     Optional option = stream.filter(x -> x < 5)
9.         .limit(3)
10.        .max((x, y) -> x-y);
11.    System.out.println(option.get());
12. }
```

説明として正しいものは次のどれですか。2つ選択してください。

- ☐ A. ①に `foo(Stream.empty());` を挿入するとコンパイルエラーとなる
- ☐ B. ①に `foo(Stream.empty());` を挿入すると実行時エラーとなる
- ☐ C. ①に `foo(Stream.iterate(1, x -> ++x));` を挿入するとコンパイルエラーとなる
- ☐ D. ①に `foo(Stream.iterate(1, x -> ++x));` を挿入すると実行時エラーとなる
- ☐ E. ①に `foo(Stream.of(5, 10));` を挿入するとコンパイルエラーとなる
- ☐ F. ①に `foo(Stream.of(5, 10));` を挿入すると実行時エラーとなる

問題 43

次のコード（抜粋）があります。

```

6.    Stream<LocalDate> stream = Stream.of(LocalDate. ➡
```



```

        of(2020, 12, 20));
7.    UnaryOperator<LocalDate> uo = l -> l;
8.    stream.filter(l -> l != null)
9.        .map(uo)
10.       .peek(System.out::println);

```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. 2020-12-20
- ☐ B. 12-20
- ☐ C. 7行目でコンパイルエラー
- ☐ D. 8行目でコンパイルエラー
- ☐ E. 実行時エラー
- ☐ F. コンパイル、実行は成功するが何も出力されない

問題 44

次のコード（抜粋）があります。

```

4. public class Test {
5.     public static void main(String[] args) {
6.         List<Item> items = Arrays.asList(
7.             new Item("apple", 100),
8.             new Item("banana", 200),
9.             new Item("orange", 130));
10.        Predicate<Item> p = b -> b.getPrice() > 150;
11.        items.stream().filter(p)
12.            .map(Item::getName)
13.            .peek(x -> System.out.print(x + " "))
14.            .collect(Collectors.toList());
15.    }
16. }
17. class Item {
18.     String name;
19.     int price;
20.     Item(String name, int price) {
21.         this.name = name;
22.         this.price = price;
23.     }
24.     String getName() {
25.         return name;
26.     }

```

```

27. int getPrice() {
28.     return price;
29. }
30. public String toString() {

```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. banana
- ☐ B. banana:200
- ☐ C. コンパイルエラー
- ☐ D. 実行時エラー
- ☐ E. コンパイル、実行は成功するが何も出力されない

問題 45

次のコード（抜粋）があります。

```

4. public class Test {
5.     public static void main(String[] args) {
6.         List<Item> items = Arrays.asList(
7.             new Item("apple", 100),
8.             new Item("banana", 200),
9.             new Item("orange", 130));
10.        Stream<Item> s = items.stream();
11.        items = s.filter(x -> x.getPrice() > 150)
12.            .collect(Collectors.toList());
13.        Stream<String> names = s.map(n -> n.getName());
14.        names.forEach(x -> System.out.println(x + " "));
15.    }
16. }

```

Item クラスは問題 44 と同じコードとします。コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. banana
- ☐ B. banana:200
- ☐ C. コンパイルエラー
- ☐ D. 実行時エラー
- ☐ E. コンパイル、実行は成功するが何も出力されない

問題 46

次のコード（抜粋）があります。

```

6.    List<Integer> x = Arrays.asList(40, 20);
7.    List<Integer> y = Arrays.asList(10, 30);
8.    【 ① 】

```

コンパイル、実行がともに成功し、実行結果として 10 20 30 40 を得るために、①に挿入可能なコードとして正しいものは次のどれですか。1 つ選択してください。

- A. `Stream.of(x, y).flatMap(e -> e.stream())`
`.sorted((a, b) -> b.compareTo(a))`
`.forEach(i -> System.out.print(i + " "));`
- B. `Stream.of(x).map(y -> y.stream())`
`.sorted()`
`.forEach(i -> System.out.print(i + " "));`
- C. `Stream.of(x, y).flatMap(e -> e.stream())`
`.sorted()`
`.forEach(i -> System.out.print(i + " "));`
- D. `List<List<Integer>> z = Arrays.asList(x, y);`
`z.stream().map(e -> e.stream())`
`.forEach(i -> {`
`i.forEach(s -> System.out.print(s + " "));});`

問題 47

次のコード（抜粋）があります。

```

6.    List<Item> items = Arrays.asList(
7.        new Item("apple", 100),
8.        new Item("banana", 200),
9.        new Item("orange", 130));
10.   double avg = 0.0;
11.   【 ① 】
12.   System.out.println(avg);

```

Item クラスは問題 44 と同じコードとします。コンパイル、実行がともに成功し、実行結果として price の平均を得るために、①に挿入可能なコードとして正しいものは次のどれですか。1 つ選択してください。

- A. `avg = items.stream()
 .average()
 .orElse(0.0);`
- B. `avg = items.stream()
 .mapToDouble(Item::getPrice)
 .average();`
- C. `avg = items.stream()
 .map(x -> x.getPrice())
 .average();`
- D. `avg = items.stream()
 .mapToInt(Item::getPrice)
 .average()
 .orElse(0.0);`

問題 48

次のコード（抜粋）があります。

```

6.    List<Item> items = Arrays.asList(
7.        new Item("apple", 100),
8.        new Item("banana", 200),
9.        new Item("orange", 130));
10.   BinaryOperator<Integer> bo = (a, b) -> a + b;
11.   int total = items.stream()
12.       .map(e -> e.getPrice())
13.       【 ① 】
14.   System.out.println(total);

```

Item クラスは問題 44 と同じコードとします。コンパイル、実行がともに成功し、実行結果として price の合計を得るために、①に挿入可能なコードとして正しいものは次のどれですか。1 つ選択してください。

- A. `.sum(e -> e);`
- B. `.sum();`
- C. `.reduce(bo).orElse(0);`
- D. `.reduce(bo).getAsInt();`

問題 49

seapp アプリケーションでは、モジュール化されていない以下の JAR ファイルを使用したいと考えています。

- util-2.0.1.jar (Automatic-Module-Name : なし)
- format-5.3.3.jar (Automatic-Module-Name : com.se.format5)
- convert-4.2.jar (Automatic-Module-Name : com.se.convert4)

seapp をモジュール化する場合、適切な module-info.java は次のどれですか。1 つ選択してください。

- ☐ A.

```
module seapp {  
    requires util;  
    requires format5;  
    requires convert4;  
}
```
- ☐ B.

```
module seapp {  
    requires util;  
    requires com.se.format5;  
    requires com.se.convert4;  
}
```
- ☐ C.

```
module seapp {  
    requires util2;  
    requires format5;  
    requires convert4;  
}
```
- ☐ D.

```
module seapp {  
    requires util-2.0.1;  
    requires com.se.format5;  
    requires com.se.convert4;  
}
```

問題 50

次のコード（抜粋）があります。

```
3. public class Test {
4.     public static void main(String[] args) {
5.         Connection con = null;
6.         try {
7.             String url = "jdbc:mysql://localhost:3306/golddb" +
8.                 "?verifyServerCertificate=false&" +
9.                 "useSSL=false&serverTimezone=JST" ;
10.            con = DriverManager.getConnection(url, "root", ➡
11.                "training");
12.        } catch (SQLException e) {
13.            e.printStackTrace();
14.        } finally {
15.            try {
16.                if(con != null) con.close();
17.            } catch (SQLException e) { e.printStackTrace(); }
18.        }
19.    }
```

また以下の実行結果があります。

実行結果

```
Test.class -> java.base
Test.class -> java.sql
```

実行したコマンドラインは次のどれですか。1つ選択してください。

- ☐ A. `javac Test.java`
`jdeps -summary Test.class`
- ☐ B. `javac Test.java`
`jdeps -jdkinternals Test.class`
- ☐ C. `javac Test.java`
`jdeps -verbose Test.class`
- ☐ D. `javac Test.java`
`jdeps -deps Test.class`

問題 51

次の 2 つの module-info.java があります。

```
module client {
    requires lib;
    uses com.seshop.BookInfo;
}

module lib {
    exports com.seshop;
    exports com.seshop.impl;
    provides com.seshop.BookInfo with
        com.seshop.impl.TeckBookInfo,
        com.seshop.impl.CookBookInfo;
}
```

説明として正しいものは次のどれですか。1 つ選択してください。

- ☐ A. client モジュールは、TeckBookInfo クラスに依存する
- ☐ B. client モジュールは、lib パッケージを公開している
- ☐ C. lib モジュールは、「requires java.util」ディレクティブを追加する必要がある
- ☐ D. lib モジュールは、com.seshop.BookInfo の実装クラスとして com.seshop.impl.TeckBookInfo クラスを提供する

問題 52

問題 51 で扱った client モジュールには、以下のコードがあります。

```
8.     ServiceLoader<BookInfo> loader = 【 ① 】;
9.     Iterator<BookInfo> iter = loader.iterator();
10.    while (iter.hasNext()) {
11.        BookInfo obj = iter.next();
12.        System.out.println(obj.sayTitle());
13.    }
```

BookInfo インタフェースの実装オブジェクトを取得するために①に挿入するコードとして正しいものは次のどれですか。1 つ選択してください。

- ☐ A. `new ServiceLoader()`
- ☐ B. `ServiceLoader.load(BookInfo.class)`
- ☐ C. `ServiceLoader.getInstance(BookInfo.class)`
- ☐ D. `ServiceLoader.builder().load(BookInfo.class)`

問題 53

jlink コマンドの説明として正しいものは次のどれですか。2 つ選択してください。

- ☐ A. カスタム JDK を作成するコマンドである
- ☐ B. jlink はビルドツールである
- ☐ C. jlink は必要に応じて Oracle 社からダウンロードする
- ☐ D. jlink はアプリケーションに必要なモジュールのみで構成されるランタイムを作成できる
- ☐ E. jlink 実行時にランチャーファイルの作成を指定することができる

問題 54

無名モジュールである、次の jar ファイルがあります。

- libX.jar は libY.jar を利用
- libY.jar は libZ.jar を利用

ボトムアップ移行では、まず、【 ① 】をモジュール化します。【 ① 】では、公開すべきパッケージを `exports` し、【 ② 】上へ配置します。【 ③ 】にある残りの無名モジュールは段階的にモジュール化します。

説明文の①、②、③に入るものはどれですか。1 つ選択してください。

- ☐ A. ① libZ.jar ② モジュールパス ③ クラスパス
- ☐ B. ① libZ.jar ② クラスパス ③ モジュールパス
- ☐ C. ① libX.jar ② モジュールパス ③ クラスパス
- ☐ D. ① libX.jar ② クラスパス ③ モジュールパス

問題 55

次の説明文があります。

- ①複数のスレッドが同じオブジェクトにアクセスし、すべてのスレッドが待ち状態となる状況
- ②複数のスレッドが進捗のない処理を繰り返し続ける状況

①と②を指す適切な用語として正しいものは次のどれですか。1つ選択してください。

- ☐ A. ①スレッドスタベーション ②デッドロック
- ☐ B. ①デッドロック ②スレッドスタベーション
- ☐ C. ①デッドロック ②ライブロック
- ☐ D. ①ライブロック ②デッドロック
- ☐ E. ①スレッドスタベーション ②ライブロック

問題 56

次のコード（抜粋）があります。

```
6. Integer a = Arrays.asList(3, 4, 5).stream().findAny().get();
7. synchronized(a) {
8.     Integer b = Arrays.asList(6, 7, 8)
9.         .parallelStream()
10.        .sorted()
11.        .findAny().get();
12.    System.out.println(a + " " + b);
13. }
```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. 7行目でコンパイルエラー
- ☐ B. 10行目でコンパイルエラー
- ☐ C. 3 6
- ☐ D. 3 8
- ☐ E. 実行結果は実行ごとに異なる可能性がある
- ☐ F. 実行時エラー

問題 57

次のコードがあります。

```
1. import java.util.concurrent.*;
2. public class Test {
3.     public static void main(String[] args) throws Exception {
```

```

4.    Object obj1 = new Object();
5.    Object obj2 = new Object();
6.    ExecutorService service = Executors.newFixedThreadPool(2);
7.    Future<?> result1 =
8.        service.submit(() -> {
9.            synchronized(obj1) {
10.                synchronized(obj2) { System.out.println("hello"); }
11.            }
12.        });
13.    Future<?> result2 =
14.        service.submit(() -> {
15.            synchronized(obj2) {
16.                synchronized(obj1) { System.out.println("bye"); }
17.            }
18.        });
19.    System.out.println(result1.get() + " " + result2.get());
20. }
21. }

```

コンパイル、実行した結果として正しいものは次のどれですか。2つ選択してください。

- ☐ A. 10行目でコンパイルエラー
- ☐ B. 16行目でコンパイルエラー
- ☐ C. hello の後に bye が出力され、その後に null が2回出力される
- ☐ D. bye の後に hello が出力され、その後に null が2回出力される
- ☐ E. 実行結果は実行ごとに異なる可能性がある
- ☐ F. 実行時にデッドロックが発生する可能性がある
- ☐ G. 実行時にライブロックが発生する可能性がある

問題 58

次のコード（抜粋）があります。

```

4. static void exec(CyclicBarrier c) {
5.     try {
6.         c.await();
7.         System.out.println("hey");
8.     } catch (BrokenBarrierException | InterruptedException e) { }
9. }
10. public static void main(String[] args) {
11.     CyclicBarrier c = new CyclicBarrier(5,

```

```

12.                                     () -> System.out.print("task ");
13.  IntStream.iterate(0, i -> i+1).limit(3)
14.                                     .parallel().forEach(i -> exec(c));
15. }

```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. 6行目でコンパイルエラー
- ☐ B. 12行目でコンパイルエラー
- ☐ C. 14行目でコンパイルエラー
- ☐ D. 実行時エラー
- ☐ E. task と hey が1回ずつ出力される
- ☐ F. コンパイルは成功するが、実行するとハングする

問題 59

並列処理の説明として正しいものは次のどれですか。2つ選択してください。

- ☐ A. 並列処理は、どのタスクが最初に完了するかは保証しない
- ☐ B. 並列処理は、常にアプリケーションのパフォーマンスを向上させる
- ☐ C. 各スレッドは、個別のメモリ領域を利用する
- ☐ D. 1つのCPUしか搭載していないPCでは、並列処理の便益は得られない
- ☐ E. 多くのリソースを必要とする重い処理を行うアプリケーションでは、並列処理の方がCPUを集中的に使用する処理より便益を得られる傾向にある

問題 60

次のコード（抜粋）があります。

```

4. public class Test {
5.     private static Integer num = 0;
6.     public static void main(String[] args) throws Exception {
7.         ExecutorService service = Executors. ➡
           newSingleThreadExecutor();
8.         List<Future<?>> list = new ArrayList<>();
9.         IntStream.iterate(0, i -> i + 1).limit(5)
10.                .forEach(i -> list.add(service.execute(() -> ➡
           ++num)));
11.         for(Future<?> r : list){
12.             System.out.print(r.get() + " ");

```

```

13.     }
14.     service.shutdown();
15. }
16. }

```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. 0 1 2 3 4
- ☐ B. 1 2 3 4 5
- ☐ C. 10 行目でコンパイルエラー
- ☐ D. 12 行目でコンパイルエラー
- ☐ E. コンパイルは成功するが、実行するとハングする
- ☐ F. 実行時エラー

問題 61

次のコード（抜粋）があります。

```

4. static void exec(CyclicBarrier c) {
5.     try {
6.         c.await();
7.         System.out.print("hey ");
8.     } catch(BrokenBarrierException | InterruptedException e) { }
9. }
10. public static void main(String[] args) {
11.     ExecutorService service = null;
12.     try {
13.         service = Executors.newFixedThreadPool(2);
14.         CyclicBarrier c = new CyclicBarrier(1);
15.         IntStream.iterate(0, i -> i + 1).limit(3)
16.             .parallel().forEach(i -> exec(c));
17.     } finally {
18.         if(service != null) service.shutdown();
19.     }
20. }

```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. 14 行目でコンパイルエラー
- ☐ B. 14 行目で実行時エラー
- ☐ C. hey が 1 回出力される
- ☐ D. hey が 2 回出力される
- ☐ E. hey が 3 回出力される

問題 62

次のコード（抜粋）があります。

```
4. Console console = System.console();
5. String str = console.readLine("please input : ");
6. console.【 ① 】(str);
```

コンパイル、実行がともに成功し、コンソールから入力した文字列を表示するために、①に挿入するコードとして正しいものは次のどれですか。3 つ選択してください。

- | | |
|------------------------------------|--|
| <input type="checkbox"/> A. out | <input type="checkbox"/> B. writer().println |
| <input type="checkbox"/> C. format | <input type="checkbox"/> D. println |
| <input type="checkbox"/> E. printf | <input type="checkbox"/> F. print |

問題 63

次のコードがあります。

```
1. class Foo implements java.io.Serializable {
2.     transient String data = "taro";
3.     public void setData(String data) { this.data = data; }
4.     public String getdData() { return data; }
5.     public Foo() { this.data = "hana"; }
6. }
7. class Bar extends Foo implements java.io.Serializable {
8.     { this.data = "nao"; }
9.     public Bar() { this.data = "kei"; }
10. }
```

Bar オブジェクトをシリアライズ後、デシリアライズした際に、data 変数に格納されている値として正しいものは次のどれですか。1 つ選択してください。

- | | |
|-------------------------------|-------------------------------|
| <input type="radio"/> A. taro | <input type="radio"/> B. hana |
| <input type="radio"/> C. nao | <input type="radio"/> D. kei |
| <input type="radio"/> E. null | |

問題 64

次のコードがあります。

```

1. import java.io.*;
2. public class Test {
3.     public static void main(String[] args) throws IOException {
4.         BufferedInputStream bi =
5.             new BufferedInputStream(new FileInputStream("t.txt"));
6.         FileInputStream in = new FileInputStream("t.txt");
7.         foo(bi);
8.         foo(in);
9.     }
10.    public static void foo(InputStream stream) throws IOException {
11.        for(int i = 0; i < 3; i++) stream.read();
12.        stream.mark(10);
13.        StringBuilder sb = new StringBuilder();
14.        for(int i = 0; i < 5; i++) sb.append((char)stream.read());
15.        stream.reset();
16.        stream.skip(1);
17.        sb.append((char)stream.read());
18.        System.out.println(sb.toString());
19.    }
20. }

```

t.txt ファイルの中身

JavaWorld

このプログラムを保存しているディレクトリ（カレントディレクトリ）には t.txt ファイルが存在します。コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. 3行目でコンパイルエラー
- ☐ B. 10行目でコンパイルエラー
- ☐ C. 7行目の呼び出しで実行時エラー
- ☐ D. 8行目の呼び出しで実行時エラー
- ☐ E. コンパイル、実行ともに成功し、18行目は同じ出力結果となる
- ☐ F. コンパイル、実行ともに成功し、18行目は異なる出力結果となる

問題 65

次のコード（抜粋）があります。

```
4. Path path = Paths.get("/memo/doc/sample/t.txt");
```

5. `System.out.println(path.subpath(1, 3).getName(1).toAbsolutePath → Path());`

このプログラムを保存しているディレクトリ（カレントディレクトリ）は、`/tmp/miko` です。コンパイル、実行した結果として正しいものは次のどれですか。1 つ選択してください。

- ☐ A. `/doc/sample/t.txt`
- ☐ B. `/memo/doc/sample`
- ☐ C. `/memo/doc/t.txt`
- ☐ D. `/tmp/miko/memo/doc/sample`
- ☐ E. `/tmp/miko/sample/t.txt`
- ☐ F. `/tmp/miko/sample`

問題 66

次のコード（抜粋）があります。

```
4. public static void main(String[] args) throws IOException {
5.     Path path = Paths.get("/foo");
6.     if (Files.isDirectory(path) && Files.isSymbolicLink(path))
7.         Files.createDirectory(path.resolve("test"));
8. }
```

`/foo` は `/doc/manual` ディレクトリのシンボリックリンクファイルです。コンパイル、実行した結果として正しいものは次のどれですか。3 つ選択してください。

- ☐ A. 常に `/doc/manual` 以下に新規に `test` ディレクトリが作成される
- ☐ B. `/doc/manual` 以下に `test` ディレクトリがない場合、新規に作成される
- ☐ C. `test` ディレクトリが作成された場合、`/doc/manual/test` でアクセスできる
- ☐ D. `test` ディレクトリが作成された場合、`/foo/test` でアクセスできる
- ☐ E. コンパイルエラーとなる
- ☐ F. コンパイルは成功するが常に実行時エラーが発生する

問題 67

次のコード（抜粋）があります。

```

6. Path path = Paths.get("/bar");
7. Files.find(path, 0, (pt, attr) -> attr.isSymbolicLink())
8.     .map(p -> p.toString())
9.     .collect(Collectors.toList())
10.    .stream()
11.    .filter(x -> x.toString().endsWith(".txt"))
12.    .forEach(System.out::print);

```

/bar ディレクトリ以下には、拡張子が txt である通常のテキストファイルおよびシンボリックファイルが保存されています。コンパイル、実行した結果として正しいものは次のどれですか。1 つ選択してください。

- ☐ A. 7 行目でコンパイルエラー
- ☐ B. 9 行目でコンパイルエラー
- ☐ C. 10 行目でコンパイルエラー
- ☐ D. 11 行目でコンパイルエラー
- ☐ E. 実行時エラー
- ☐ F. 拡張子が txt であるシンボリックファイル名がすべて表示される
- ☐ G. 何も表示されない

問題 68

次のコード（抜粋）があります。

```

5. public static void main(String[] args) throws IOException {
6.     Path path = Paths.get("foo/hello.txt");
7.     Files.find(
8.         path.getParent(), 10,0,
9.         (p) -> p.toString().endsWith(".txt") ||
10.            Files.isDirectory(p))
11.        .collect(Collectors.toList())
12.        .forEach(System.out::println);
13. }

```

このプログラムを保存しているディレクトリ（カレントディレクトリ）には、foo/hello.txt が存在します。また、foo ディレクトリ以下にはディレクトリや txt 拡張子をもつファイルが保存されています。コンパイル、実行した結果として正しいものは次のどれですか。2 つ選択してください。

- ☐ A. 8 行目でコンパイルエラー
- ☐ B. 9 行目でコンパイルエラー
- ☐ C. 10 行目でコンパイルエラー

- ☐ D. foo 以下にあるディレクトリ名が表示される
- ☐ E. foo 以下にある txt 拡張子をもつファイル名が表示される

問題 69

説明として正しいものは次のどれですか。2 つ選択してください。

- ☐ A. JDBC4.0 以降のドライバを使用している場合、Class.forName() によるドライバのロード処理は必須である
- ☐ B. Class.forName() でドライバクラスが見つからない場合は、ClassNotFoundException 例外がスローされる
- ☐ C. Class.forName() でドライバクラスが見つからない場合は、SQLException 例外がスローされる
- ☐ D. DriverManager.getConnection() で Connection が取得できない場合、ClassNotFoundException 例外がスローされる
- ☐ E. DriverManager.getConnection() で Connection が取得できない場合、SQLException 例外がスローされる

問題 70

次のコード（抜粋）があります。

```

4. public static void main(String[] args) throws SQLException {
5.     String sql = "update department set dept_address = 'tmp'";
6.     try(Connection con = DbConnector.getConnection();
7.         Statement stmt = con.createStatement()) {
8.         int num = stmt.executeUpdate(sql);
9.         System.out.println(num);
10.    }
11. }
```

前提

- DbConnector クラスの getConnection() メソッドから接続が確立された Connection オブジェクトは取得できているとします
- department テーブルには 5 レコードが格納されているとします

コンパイル、実行した結果として正しいものは次のどれですか。1 つ選択してください。

- ☐ A. 0
- ☐ B. 1
- ☐ C. 5
- ☐ D. コンパイルエラー
- ☐ E. 実行時エラー

問題 71

次のコード（抜粋）があります。

```
4. public static void main(String[] args) {
5.     String sql = "SELECT dept_code FROM department " +
6.                 "order by dept_code";
7.     try(Connection con = DbConnector.getConnection();
8.         Statement stmt = con.createStatement();
9.         ResultSet rs = stmt.executeQuery(sql)) {
10.        System.out.println(rs.getString(1));
11.    } catch (SQLException e) { e.printStackTrace(); }
12. }
```

前提

- DbConnector クラスの getConnection() メソッドから接続が確立された Connection オブジェクトは取得できているとします
- department テーブルには dept_code が 1 ～ 5 までの 5 レコードが格納されているとします

コンパイル、実行した結果として正しいものは次のどれですか。1 つ選択してください。

- ☐ A. 1 のみ出力する
- ☐ B. 1 ～ 5 を出力する
- ☐ C. 1 ～ 5 のいずれかの値を出力する
- ☐ D. コンパイルエラー
- ☐ E. 実行時エラー

問題 72

次のコード（抜粋）があります。

```
4. public static void main(String[] args) {
5.     String sql = "SELECT dept_code FROM department " +
6.                 "order by dept_code";
```

```

7. try(Connection con = DbConnector.getConnection();
8.     Statement stmt = con.createStatement(
9.         ResultSet.TYPE_SCROLL_INSENSITIVE,
10.        ResultSet.CONCUR_UPDATABLE);
11.    ResultSet rs = stmt.executeQuery(sql)) {
12.        rs.beforeFirst(); rs.absolute(3);
13.        rs.previous(); rs.relative(2);
14.        System.out.println(rs.afterLast());
15.    } catch (SQLException e) { e.printStackTrace(); }
16. }

```

前提

- DbConnector クラスの getConnection() メソッドから接続が確立された Connection オブジェクトは取得できているとします
- department テーブルには dept_code が 1 ～ 5 までの 5 レコードが格納されているとします

コンパイル、実行した結果として正しいものは次のどれですか。1 つ選択してください。

- ☐ A. 4
 ☐ B. 5
☐ C. -1
 ☐ D. コンパイルエラー
☐ E. 実行時エラー

問題 73

有効なコードは次のどれですか。2 つ選択してください。

- ☐ A. Locale.create("US");
☐ B. Locale.create("en", "US");
☐ C. new Locale("US");
☐ D. new Locale("en", "US");
☐ E. Locale."US";
☐ F. Locale.getLocale("US");

問題 74

リソースバンドルの説明として正しいものは次のどれですか。1 つ選択してください。

- A. プロパティファイルは複数配置することはできない
- B. ListResourceBundle もしくはプロパティファイルを使用したリソースバンドルでは、String 型以外のデータ型でリソースの取得が可能である
- C. プロパティファイルを使用したリソースバンドルは、ListResourceBundle を使用した Java クラスに置き換えることができる
- D. リソースバンドルの実装では、アプリケーション内で ListResourceBundle を使用した Java クラスファイルとプロパティファイルを同梱することはできない
- E. プロパティファイルのファイル名、拡張子は任意である

問題 75

次のファイルが用意されています。

(MyResources_en_US.properties)

```
val1 = hana
val2 = 30
```

また、次のコード（抜粋）があります。

```
5. ResourceBundle bundle
6.           = ResourceBundle.getBundle("MyResources", Locale.US);
```

プロパティファイル内のすべての値のみ（キーは含まない）出力するコードとして正しいものは次のどれですか。1 つ選択してください。

- A. bundle.keys()
 - .stream()
 - .map(k -> bundle.getString(k))
 - .forEach(System.out::println);
- B. bundle.keys()
 - .stream()
 - .map(k -> k)
 - .forEach(System.out::println);
- C. bundle.stream()
 - .map(k -> bundle.getString(k))

- ```

 .forEach(System.out::println);
○ D. bundle.stream()
 .map(k -> k)
 .forEach(System.out::println);
○ E. bundle.keySet()
 .stream()
 .map(k -> bundle.getString(k))
 .forEach(System.out::println);
○ F. bundle.keySet()
 .stream()
 .map(k -> k)
 .forEach(System.out::println);

```

## 問題 76

次のコード（抜粋）があります。

```

7. String pattern = "#,###,000.0#";
8. DoubleStream.of(5.21, 8.49, 1234)
9. .mapToObj(v -> new DecimalFormat(pattern).format(v))
10. .forEach(x -> System.out.print(x + " "));

```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- A. 5.21 8.49 1,234                      ○ B. 005.21 008.49 1,234.0  
 ○ C. 005.2 008.4 1,234.0                ○ D. 005.21 008.49 1,234.00  
 ○ E. 実行時エラー

## 問題 77

次のコード（抜粋）があります。

```

5. public static void main(String[] args) throws IOException {
6. Path p1 = Paths.get("foo/.").resolve(Paths.get("x.txt"));
7. Path p2 = new File("foo/./bar/./x.txt").toPath();
8. boolean b1 = Files.isSameFile(p1, p2);
9. boolean b2 = p1.equals(p2);
10. boolean b3 = p1.normalize().equals(p2.normalize());

```

```
11. System.out.println(b1 + " " + b2 + " " + b3);
12. }
```

コンパイル、実行した結果として正しいものは次のどれですか。1つ選択してください。

- ☐ A. true false true
- ☐ B. true false false
- ☐ C. false true false
- ☐ D. false true true
- ☐ E. true true true
- ☐ F. false false false

#### 問題 78

入力値の妥当性チェックについて、説明として正しいものは次のどれですか。1つ選択してください。

- ☐ A. Normalizer クラスの normalize() メソッドで文字列を正規化してから検査する
- ☐ B. Normalizer クラスの matcher() メソッドで文字列を正規化してから検査する
- ☐ C. 文字列は検査してから Normalizer クラスの normalize() メソッドで正規化する
- ☐ D. 文字列は検査してから Normalizer クラスの matcher() メソッドで正規化する

#### 問題 79

PreparedStatementの説明として正しいものは次のどれですか。1つ選択してください。

- ☐ A. 検索処理には使用できるが、更新処理には使用できない
- ☐ B. Statement はどのデータベースでも利用可能であるが、PreparedStatement の利用可否はデータベースに依存する
- ☐ C. IN パラメータに対し、setter メソッドが呼び出していない場合は、デフォルト値として NULL が渡される
- ☐ D. 文字列を格納する IN パラメータに列名が含まれるとコンパイルエラーとなる
- ☐ E. IN パラメータを利用しない PreparedStatement も実行可能である

## 問題 80

次のコード（抜粋）があります。

```
2. public static long add(long num1, long num2) {
3. return num1 + num2;
4. }
```

add() メソッドの引数では、オーバーフロー（扱える値の上限を超える）の可能性があります。このメソッドの修正として誤った説明は次のどれですか。1つ選択してください。

- A. add() メソッドの呼び出し前に、事前に num1 と num2 がオーバーフローにならない範囲の値であるか検査する
- B. num1 と num2 を BigInteger に変換して、BigInteger クラスのメソッドを使用して演算する
- C. + 演算子ではなく、Math クラスの addExact() メソッドで演算する
- D. num1、num2 をアップキャストしてから演算する

## 解答・解説

## 問題 1 正解：D

3行目：基本データ型からラッパークラスへの自動変換（Boxing）が行われません。

4行目：Boolean クラスの booleanValue() メソッドにより基本データ型へ変換しています。

5行目：数字の途中に「\_」（アンダースコア）の使用は可能です。

6行目：基本データ型からラッパークラスへの自動変換（Boxing）が行われません。

7行目：Long 型はクラス型であるため null の代入が可能です。

8行目：c変数はlong型、d変数はLong型であり、文法上は問題ないためコンパイルは成功します。Long→longの変換では、実際はlongValue()メソッドが使用されます。しかし、d変数にはnullが格納されており、8行目でnullに対してlongValue()メソッドが実行されるため、NullPointerException例外が発生します。

**問題 2 正解：A、D**

---

選択肢 A と D が正しいです。なお、`s1.hashCode() == s2.hashCode()` が `false` の場合は、`equals()` メソッドによる比較でも `false` を返すように実装します。

**問題 3 正解：B**

---

コードはコンパイル、実行ともに成功します。2行目と6行目で外側のクラスとローカルクラスで同じ名前の変数を使用していますが、問題ありません。なお、ローカルクラスから外側クラスの同名のインスタンス変数にアクセスする場合は、6行目のようにクラス名.`this`.変数名とします。また、3行目、7行目も同じメソッド名を使用していますが問題ありません。したがって、15行目により3行目が呼び出され、12行目により7行目が実行されるため、出力結果は `num : 200` です。

**問題 4 正解：B**

---

列挙型では明示的なコンストラクタの定義は可能ですが、`new` によるインスタンス化は許可されていないため、暗黙で `private` 修飾子が付与されます。したがって、5行目のようにコンストラクタに `public` を付与するとコンパイルエラーとなります。なお、5行目の `public` を削除すると、コンパイル、実行ともに成功し、`VAL3 false` と出力します。Vals 列挙型では、`boolean` 型の `data` 変数を保持しています。3行目では `Boolean.FALSE` を使用していますが、`Boolean` クラスの `FALSE` 定数は `false` が格納されているため問題ありません。

**問題 5 正解：F**

---

X、Y、Z は `static` なネストクラスであり、Y と Z は X のサブクラスです。9行目のコードは問題ありません。10行目では、`obj1` 変数を Z 型でキャストしています。`obj1` (つまり X 型) と `obj2` (つまり Z 型) には継承関係があるため、コンパイルは成功します。しかし、もともと `obj1` 変数が参照しているオブジェクトは Y 型であるため、実行時に型の違いにより、`ClassCastException` 例外が発生します。

**問題 6 正解：B、E**

---

A. インタフェースでの抽象メソッドの使用可能な修飾子は、`public`、



abstractのみです。staticは使用できません。

- B. privateなstaticメソッドを具象メソッドとして定義可能です。
- C. インタフェースでの抽象メソッドの使用可能な修飾子は、public、abstractのみです。privateは使用できません。
- D. デフォルトメソッドは、インタフェース内でデフォルトの実装を定義しておき、他クラスから利用できるように用意するものであるため、privateは使用できません。
- E. privateな具象メソッドは定義可能です。
- F. 具象メソッドは、staticもしくはデフォルトメソッドである必要があります。

インタフェースでの抽象メソッドと具象メソッドの特徴をまとめます。

#### インタフェースでの抽象メソッド

- 使用可能な修飾子は、public、abstractのみ

#### インタフェースでの具象メソッド

- staticメソッド (public、private)
- デフォルト (default) メソッド (publicのみ)
- 非staticのメソッド (privateのみ)

### 問題7 正解：C

SE 8以降では、インタフェースに処理を記述したメソッド(デフォルトメソッド)が定義可能です。したがって、2行目、5行目は問題ありません。インタフェースで抽象メソッドを宣言すると、暗黙的に「public abstract」修飾子が付与されるため、6行目のように明示しても、9行目のように省略しても問題ありません。8行目のように、インタフェースが複数のインタフェースを継承することは許可されていますが、問題文のコードは、XおよびYのいずれもmethodA()をデフォルトメソッドとして定義しており、Zはそれらを引き継ぐこととなるため、コンパイルエラーとなります。

### 問題8 正解：E

Testクラスは抽象クラスであり、抽象メソッドとしてpublicなfoo()を宣言し

ています。Testクラスを継承したExTestはstaticなネストクラスです。9行目でfoo()メソッドのオーバーライドを試みっていますが、アクセス修飾子を付与していません。これはpublicより狭くなるため、9行目でコンパイルエラーです。もし、publicを付与した場合はコンパイル、実行ともに成功しhelloが出力されます。

### 問題 9 正解：A、C、F

@Overrideアノテーションを使用すると、オーバーライド時にスーパークラスと同じメソッド名、引数の型、数、並び、戻り値の型をすべてチェックします。したがって、Objectクラスで定義されているメソッドと同じである選択肢A、Cは正しいです。なお、選択肢Bは@Overrideを削除すると、継承でのオーバーロードとみなされ定義が可能となります。また、インタフェースで定義したデフォルトメソッドのオーバーライドも可能です。ただし、インタフェースで宣言したメソッドは暗黙でpublicが付与されるため、オーバーライド時にはpublicの付与が必須です。したがって、選択肢Fは正しく、選択肢Eは誤りです。

### 問題 10 正解：A、E

@FunctionalInterfaceは、関数型インタフェースとしての要件を満たしているかをチェックする役割をします。

- 単一の抽象メソッドをもつインタフェースとする
- ただし、staticメソッドやデフォルトメソッドは定義可能
- java.lang.Objectクラスのpublicメソッドは抽象メソッドとしての宣言は可能
- 関数型インタフェースとして明示する場合は、@FunctionalInterfaceを付与する

- A. java.lang.Objectクラスで提供されているequals()、hashCode()、toString()の3つはデフォルトメソッドとして定義することはできませんが、問題文のequals()は戻り値がvoidであるため定義可能です。
- B. staticもしくはデフォルト、private以外で具象メソッドは定義できないためコンパイルエラーです。

- C. equals() を抽象メソッドとして宣言していますが、スーパークラスとなる `java.lang.Object` クラスで equals() を実装しているため、このインタフェースは抽象メソッドが宣言されていないことによりコンパイルエラーです。
- D. 抽象メソッドが2つあるためコンパイルエラーです。
- E. `private` な具象メソッドと、`public` な抽象メソッドが1つあるため正しいです。
- F. `static` な抽象メソッドは定義できないためコンパイルエラーです。

### 問題 11 正解：B

問題文の `order()` メソッドは、引数が可変長引数であるため警告が表示されます。そのため、`@SafeVarargs` により警告を抑制しています。しかし、`@SafeVararg` を付与するメソッドは、`final`、`static`、`private` のいずれか（もしくはその組み合わせ）である必要があるため、8行目でコンパイルエラーです。問題文の `order()` メソッドに `static` もしくは `final` が付与されれば、コンパイルは成功します。なお、可変長引数の実体は配列であるため、10行目の `length`（配列の大きさを取得）の利用も問題ありません。

### 問題 12 正解：A、B

`TYPE` はクラス、インタフェース（アノテーションを含む）、`enum` です。  
`METHOD` はメソッド、`PARAMETER` はメソッドパラメータです。選択肢にあるローカル変数は `LOCAL_VARIABLE` 定数で定義されています。

### 問題 13 正解：F

`Foo` クラスは `AutoCloseable` を実装し、`Bar` クラスは `Closeable` を実装しています。しかし、13、14行目では、通常の `try` ブロックでインスタンス化しているため、`close` メソッドが自動的に呼ばれることはありません。したがって、出力結果は選択肢 F の「D」のみです。以下のコード例のように `try-with-resources` 文で実装すれば、各 `close` メソッドが呼び出され、出力結果は「BACD」となります。

```
12. try(
13. Foo foo = new Foo();
14. Bar bar = new Bar()) {
```

**問題 14 正解：E**

---

6行目ではマルチキャッチで例外をキャッチしているため、`e`変数は暗黙的に`final`となります。したがって、選択肢A～Dのように再代入を行うことはできません。

**問題 15 正解：F**

---

`main()` メソッドでは、5行目により `RuntimeException` 例外が発生します。また、4行目でインスタンス化している2つの `Foo` オブジェクトの `close()` メソッドでも `RuntimeException` 例外が発生します。6行目では `main()` 側の例外がキャッチされるため、`close()` メソッドで発生した `RuntimeException` 例外オブジェクトは抑制されます。したがって、7行目の表示は「main」となり、8行目の `getSuppressed()` メソッドの戻り値となる `Throwable[]` には、2つの `RuntimeException` 例外オブジェクトが格納されています。その結果、出力は「2」となります。

**問題 16 正解：E、F**

---

`AutoCloseable` インタフェースは `java.lang` パッケージ、`Closeable` インタフェースは `java.io` パッケージで提供されているため、選択肢A、B、C、Dは誤りです。選択肢E、Fのシグニチャは正しいため、選択肢Gは誤りです。

`AutoCloseable` : `void close() throws Exception`

`Closeable` : `void close() throws IOException`

**問題 17 正解：A、C**

---

`assert` はJDK1.4から予約語となっており変数名には使用できないため、選択肢Aは正しいです。また、アサーションはプログラムの動作をチェックする目的で使用するため、8行目のように処理（インクリメント）を含むことは不適切です。また、アサーション実行時のメッセージは省略可能であり、`boolean` 式を囲む()の有無は任意なので、選択肢D、Eは誤りです。

**問題 18 正解：F**

---

3行目で`num`変数は100で初期化され、4行目で101となります。5行目では、`num`変数が`null`ではなく100以下であれば`AssertionError`は発生しませ

ん。現在 num 変数は 101 であるため、アサーションを有効にして実行していれば `AssertionError` が発生します。しかし、問題文の実行例を見ると「java Test」としているため、実行結果は「101」となり、選択肢 F が正しいです。もし「java -ea Test」とすれば `AssertionError` が発生します。

#### 問題 19 正解：A、B、D、E

イミュータブルオブジェクトは、保持している値が不変であることを保証する必要があります。したがって、クラスは `final` とし、メンバ変数は `private` かつ、`final` の修飾子を付与するため、選択肢 A、E は正しいです。なお、メンバ変数が参照型の場合、メソッド（およびコンストラクタ）の引数や戻り値では参照値のコピーがやりとりされるため、引数で外部から受け取った場合は、元の参照場所から更新されないように配慮すべきです。したがって、選択肢 B、D は正しいです。List で保持する型を Object 型にすると、7、10 行目でコンパイルエラーとなります。したがって、選択肢 C は誤りです。なお、選択肢 F の `Immutable` インタフェースは API として提供されていません。

#### 問題 20 正解：H

問題文のコードのコンパイルは成功します。5 行目により `deque` オブジェクトには CBA が格納されています。6 行目の `pop()` によりキューの最初（つまり C）が削除されます。7 行目の `peek()` によりキューの先頭を取得しますが、削除しません。8 行目の `remove()` により B が削除され、9 行目の `pop()` により A が削除されます。そして 8 行目の `remove()` メソッドが再度呼ばれた際、キューが空のため実行時に `NoSuchElementException` 例外が発生します。

#### 問題 21 正解：A、E、F

`Vector` クラスは `List` インタフェースを実装していますが、`HashSet` クラスは実装していないため、選択肢 A は正しく、選択肢 B は誤りです。「`<? extends Object>`」はインスタンス化の際の型として指定することができません。したがって、選択肢 C は誤りです。選択肢 D は `Number` と `Double` に継承関係がありますが、`HashSet<Number>` と `HashSet<Double>` 自体には継承関係がないため、誤りです。`Exception` クラスは `ArithmeticException` のスーパークラスであるため、選択肢 E は正しいです。選択肢 F は、`Double` 型が「`? extends`

Number」に対応するため正しいです。

#### 問題 22 正解：E

---

4行目ではHashMapをインスタンス化しているため、put(キー, 値)で格納する必要があります。選択肢A、Cは、引数が1つであるため誤りです。また、選択肢B、Dは、メソッド名がadd()であるため誤りです。もし選択肢Dのメソッド名がputであれば、コンパイル、実行ともに成功します。

#### 問題 23 正解：A、D、E

---

6行目により、foo()メソッドの引数はExceptionもしくはそのサブクラスであれば受取可能です。メソッド呼び出し時に使用する型名の明示的な宣言は必要ありませんが、選択肢Aのようにメソッド名の前に型宣言をすることも可能です。選択肢B、CのThrowableは、Exceptionのスーパークラスになるため誤りです。選択肢DのIOExceptionは、Exceptionのサブクラスであるため正しいです。

#### 問題 24 正解：B、D

---

- A. 1行目の右辺がComparator<?>となっており、型が不明のためコンパイルエラーです。
- B. 1行目の左辺でComparator<Integer>としており、また、右辺は<>のみですが、左辺により類推可能であるため正しいコードです。
- C. 1行目の左辺でComparator<>としており、ダイヤモンド演算子は左辺では使用できないためコンパイルエラーです。
- D. ラムダ式で実装しており、正しいコードです。
- E. ラムダ式で実装していますが、左辺でvarが宣言されています。varが使用可能なのは、右辺により型推論できる場合であり、ラムダ式は左辺で型推論して実装するため、このコードは型推論ができずコンパイルエラーです。

#### 問題 25 正解：A

---

高度な出題内容ですが、1行ずつ確認します。TreeSetの内部ではソートが行われます。したがって、22行目でTreeSetオブジェクトを生成し、23行目

で要素を格納したタイミングで、`compareTo()` メソッドにより比較が行われます。11、12行目では、`String` 型の `val2` に対して `compareTo()` メソッドを呼び出しています。その結果、`TreeSet` 内は `("60","x")`、`("10","y")` の順番で `Foo` オブジェクトが管理されます。また、24行目では `Comparator` を実装した `Foo` クラスの `obj1` をコンストラクタの引数に指定して、`TreeSet` オブジェクトを生成しています。すると、25行目で要素が格納されたタイミングで、`Comparator` の `compare()` メソッド (13行目) によって比較が行われます。14、15行目では、`val1` を `int` 型に変換し比較を行っています。その結果、`TreeSet` 内は、`("10","y")`、`("60","x")` の順番で `Foo` オブジェクトが管理されます。なお、`TreeSet` では重複した要素は保持しないため、`obj1` は上書きされます。26行目により9行目が実行されるため、出力結果は選択肢 A のとおり `[60, 10] [10, 60]` となります。

### 問題 26 正解：D

`binarySearch()` メソッドは、コレクションや配列から特定の要素を検索し、見つかった場合は目的の要素のインデックスを返します。しかし、検索を実行する前に自然順序づけに従って昇順にソートする必要があります。これを実行していなかった場合、検索結果の正しさは保証していません。4行目では、`Comparator` を降順の実装とし、6行目の `sort()` メソッドの第2引数で指定しているため、選択肢 D が正しいです。

### 問題 27 正解：B、F

4、5行目より、`Foo` クラスでは実行時に `Integer` 型と `Object` 型のオブジェクトを扱う必要があるため、①の型パラメータは `<T>` とします。また、②は左辺とあわせて `<Integer>` とするか、ダイヤモンド演算子である `<>` とします。また、5行目の右辺では `<>` 自体を省略しているため、コンパイル時に警告メッセージが出力されますが、コンパイルは成功します。

### 問題 28 正解：B

メソッドで型パラメータリストを指定する位置は、メソッド宣言の修飾子と戻り値の間です。`public` の後や戻り値の後に指定するとコンパイルエラーとなります。また、コード内で型パラメータを使用する場合、`<>` による型パラメータリストが必要であるため、選択肢 F は誤りです。

**問題 29 正解：C**

List インタフェースの `replaceAll()` メソッドは、引数に `UnaryOperator` をとります。

```
default void replaceAll(UnaryOperator<E> operator)
```

また、`UnaryOperator` の実装は、以下の抽象メソッドを実装します。

```
T apply(T t)
```

- A. 宣言のない `y` 変数が出ていますので、コンパイルエラーです。
- B. `return` を使用した際は文にする必要があります。最後に `;` (セミコロン) がないため、コンパイルエラーです。
- C. 正しいコードです。
- D. ラムダ式の左辺が `double` となっているので、コンパイルエラーです。  
`Double` 型であれば正しいコードです。
- E. ラムダ式の左辺で型 (この場合は `var`) を記述している場合は、`()` が必要のため、コンパイルエラーです。

**問題 30 正解：D**

Foo インタフェースの `exec()` メソッドは、引数に `T` をとり、戻り値に `R` を返します。引数の数、それぞれの型と同じメソッドをもつインタフェースは `Function` です。

`Function` の実装は、以下の抽象メソッドを実装します。

```
R apply(T t)
```

**問題 31 正解：D**

`DoubleFunction` インタフェースは、`XXXFunction` の命名により引数の型は `double` で決まっており、戻り値の型は `<>` により宣言する必要があります。  
`DoubleFunction` の実装は、以下の抽象メソッドを実装します。

```
R apply(double value)
```



問題文の6行目の左辺では、「DoubleFunction f」と宣言しているため、7行目のapply()メソッドはObject型を返してしまいます。Double型で返すためには、6行目を「DoubleFunction<Double> f」と宣言する必要があります。

### 問題 32 正解：B

- A. range()メソッドは終了値を含まないため、このコードは実行できますが、val値は45です。
- B. rangeClosed()メソッドは終了値を含みます。まず、skip(1)により1番目の要素（つまり0）を破棄します。次にlimit(10)により10まで切り詰めた要素のストリームを返します。次にreduce()メソッドにより初期値0とし、集約処理として「a + b」としています。その結果、val値は55です。
- C. ラムダ式から参照されるローカル変数は、finalまたは事実上のfinalでなければいけません。peek()メソッド内で「val += a」としているため、コンパイルエラーです。
- D. コンパイルは成功しますが、forEach()メソッド内で「n += val」としているため、val値は0です。

### 問題 33 正解：D

4行目ではiterate()メソッドで初期値として""を指定し、a文字列を結合した要素を無限に用意します。5行目ではlimit()メソッドで要素数を2個に制限し、map()メソッドで各要素にxを結合しています。ただし、mapは中間操作であり、そのストリームをSystem.out.println()で出力しているため、実行結果はDとなります。5行目を以下のように実装すると、実行結果は選択肢Aの「xax」となります。

```
(修正後) stream.limit(2).map(x -> x + "x")
 .forEach(x -> System.out.print(x));
```

### 問題 34 正解：E

5行目で各要素の文字数が3文字を超えるかテストするPredicate式を作成し、6行目ではx文字列を結合した要素を無限に用意します。7行目のnoneMatch()はどの要素も指定された条件に一致しなければtrueを返し、8行目のanyMatch()はいずれかの要素が指定された条件に一致すればtrueを返し

ます。7、8行目を単独で使用していた場合、7行目はfalseを返し、8行目はtrueを返します。しかし、各メソッドは終端操作であるため、7行目の実行後にstreamは終了します。したがって、実行時に8行目でIllegalStateException例外がスローされます。

#### 問題 35 正解：D、E

リダクション操作とは入力要素をもとに結合操作を繰り返し実行して、単一の結果を得る操作です。選択肢B、Cは中間操作のため誤りです。その他の選択肢は終端操作のメソッドですが、選択肢AのfindAny()はいずれかの要素を返し、選択肢FのtoArray()はストリーム内の要素を配列にして返します。したがって、いずれも誤りです。選択肢Dのcollect()はストリームから要素をまとめて1つのオブジェクトを取得し、選択肢Eのcount()は要素の個数を返すため正しいです。

#### 問題 36 正解：A

4行目では空白文字をもとに要素を作成しています。5行目により①に入るメソッドは引数をもつことがわかります。選択肢D、Eは引数をもたないため誤りです。選択肢A、B、CはPredicate型の引数を取り、実装内容はString::isEmptyであるため、要素の文字列が空文字（length()が0となる文字列）の条件にマッチするか否かを確認します。したがって、実行結果としてtrueとなるには、選択肢Aのいずれもマッチしない場合trueを返すnoneMatchが正しいです。

#### 問題 37 正解：C

問題文のコードでは、mySort()メソッドの引数で受け取ったリストを、Collections.sort()メソッドの第1引数に指定しています。また、第2引数では降順となるComparatorの実装を指定しています。StreamにはcompareTo()、compare()メソッドは提供されていないため、選択肢A、D、E、Fは誤りです。また、collect()メソッドは引数をとるため、選択肢Bは誤りです。選択肢CはStreamのsorted()を使用し、collect()でリストへ変換しているため正しいです。

#### 問題 38 正解：C

4、5行目によりストリームにはhello文字列をもつ要素が2つ格納されます。

6行目では各要素が3文字を超える要素のみにフィルタされますが、変化はありません。7行目の`peek()`は指定された処理を行う中間操作です。問題文のように出力処理の指定も可能です。また、8行目で`forEach()`により出力処理が行われるため、`hello`が4回出力されます。

### 問題 39 正解 : D

4行目で`Stream<Integer>`を生成後、5行目は`mapToInt()`による`Stream<Integer>→IntStream`の変換であるため正しいです。7行目は`mapToDouble()`による`Stream<Integer>→DoubleStream`の変換であるため正しいです。8行目では`mapToInt(a -> a)`を使用していますが、`stream4`は`DoubleStream`型であり、`double`値を戻り値として返しているためコンパイルエラーです。`DoubleStream→Stream<Integer>`の変換例は以下のとおりです。

```
Stream<Double> stream5 = stream4.mapToObj(a -> a);
Stream<Integer> stream6 = stream5.map(a -> a.intValue());
```

### 問題 40 正解 : C

`stream1`の要素は`"a"`, `"b"`, `"ax"`であり、`stream2`の要素は空です。8行目では`stream1`に対して`partitioningBy()`によるグルーピングを行っていますが、`x`から始まる要素はないため、すべて`false`グループに属します。また、10行目では`stream2`に対して`groupingBy()`によるグルーピングを行っていますが、`stream2`は空であるため実行結果は`{}`となります。なお、次のコード例のように`partitioningBy()`メソッドを空のストリームで行った場合の実行結果は、`「{false=[], true=[]}」`となります。

```
5. Stream<String> stream1 = Stream.empty();
6. Stream<String> stream2 = Stream.empty();
 (途中省略) // 問題文の7~10行目と同じコード
11. System.out.println(map1 + " " + map2); // {false=[],true=[]} {}
```

### 問題 41 正解 : E

5行目の`peek()`により`DoubleStream`内の要素が出力されます。その後、`filter()`により0.5を超える要素である0.8のみとなり、`count()`よりその個数であ

る1が返ります。しかし、1を出力するコードは記述していないため出力はされません。

#### 問題 42 正解：B、F

いずれのコードを挿入してもコンパイルは成功します。①に `foo(Stream.empty());` を挿入すると、11行目では `empty()` に対して `get()` メソッドを実行しているため、実行時に `NoSuchElementException` 例外が発生します。同様に、①に `foo(Stream.of(5, 10));` を挿入すると、8行目によりストリームは要素をもたないため、11行目で `NoSuchElementException` 例外が発生します。なお、`foo(Stream.iterate(1, x -> ++x));` を挿入した場合は、実行結果として3を出力します。

#### 問題 43 正解：F

6行目で2020-12-20の要素をもつストリームを取得します。8行目で `null` ではないものをフィルタし、9行目では特に変換する処理はなく、10行目の `peek()` が実行されます。しかし、`peek` は中間操作であるため、終端操作がなければ表示はされません。したがって、実行結果は選択肢Fです。もし、`peek()` の代わりに `forEach()` で出力を実装していれば、2020-12-20と表示されます。

#### 問題 44 正解：A

7～9行目では、Itemオブジェクトを3つ格納したリストを作成します。10行目で、Itemオブジェクトがもつ `price` が150を超えるかどうかの判断をする `Predicate` を作成します。11行目では `filter()` の引数に `Predicate` を指定しているので、この段階で `banana` のItemオブジェクトに絞り込まれます。12行目ではメソッド参照により `name` のみで構成されるストリームを作成し、13行目で要素(つまり `name`)を出力しています。したがって、実行結果は `banana` のみです。なお、14行目ではリストを返していますが、受け取っていないだけでありエラーにはなりません。

#### 問題 45 正解：D

11行目では `filter()` を使用して `price` が150を超える要素を絞り込み、12行目により `banana` のItemオブジェクトをもつリストが作成され、`items` 変数に格納

します。13行目では、10行目のストリーム(s変数)を使用してmap()を実行していますが、このストリームは12行目の終端処理(collect()メソッド)によりクローズしています。そのため、実行時に13行目でIllegalStateException例外が発生します。

#### 問題 46 正解 : C

---

問題文では2つのリストから実行結果として10 20 30 40を得る必要があるため、flatMap()を使用します。選択肢CはStream.of()により複数のリストを扱うストリームを取得し、flatMap()により連結します。sorted()により並び替えを行い、出力しています。なお、選択肢Aはsorted()の際に降順の実装を指定しているため、実行結果が40 30 20 10となります。選択肢Bはmap()メソッド内でy変数を指定しているため、ラムダ式の外側の変数でラムダ式内と同じ名前の変数を使用していることにより、コンパイルエラーです。選択肢Dはコンパイル、実行は成功しますが、並び替えを行っていないため、実行結果は40 20 10 30です。

#### 問題 47 正解 : D

---

average()はIntStreamやDoubleStreamなど数値型ストリームで提供されており、通常のStreamでは提供されていません。したがって、選択肢A、Cはコンパイルエラーです。また、average()の戻り値はOptional型(問題文の例ではOptionalDouble型)です。したがって、選択肢DのようにOptionalで提供するメソッドを使用して値(この例ではDouble値)を取得する必要があります。選択肢Bはコンパイルエラーとなりますが、average()の後にgetAsDouble()を追加すれば、Double値を取得できます。

#### 問題 48 正解 : C

---

IntStream、LongStream、DoubleStreamでsum()メソッドが提供されていますが、引数はとらないため選択肢Aは誤りです。Streamにはsum()メソッドが提供されていないため、選択肢Bは誤りです。もし、12行目がmap()ではなくmapToInt()メソッドを使用していれば、sum()メソッドによる合計値を取得可能です。選択肢Cはreduce()の引数に集約関数として10行目のBinaryOperatorを指定し、戻り値はOptional型となるため、orElse()で合計値

を取得しており正しいコードです。選択肢Dは`getAsInt()`メソッドを使用していますが、`getAsInt()`はOptionalではなく、OptionalIntで提供されているメソッドであるため、コンパイルエラーです。

#### 問題 49 正解 : B

モジュール化されていないJARファイルは、モジュール名がありません。もし、Automatic-Module-Name属性が設定されていればその名前を使用し、設定されていない場合はJARファイル名で解決します。問題文より`format-5.3.3.jar`と`convert-4.2.jar`はAutomatic-Module-Name属性が設定されているため、その名前を使用します。`util-2.0.1.jar`は設定されていないため、バージョン番号と拡張子が削除され、`util`という名前を使用します。したがって、選択肢Bが正しいです。

#### 問題 50 正解 : A

問題文の実行結果から依存関係のサマリー情報のみであるため、`-summary`オプションを使用しています。選択肢Bの「`-jdkinternals`」は、JDKの内部APIでクラス・レベルの依存関係を検索します。選択肢Cの「`-verbose`」は、依存関係を詳細に表示します。選択肢Dの「`-deps`」は、オプションとして提供されていません。

#### 問題 51 正解 : D

問題文の `module-info.java` から読み取れる内容をまとめます。

##### lib モジュール

- `com.seshop` パッケージを公開する
- `com.seshop.impl` パッケージを公開する
- 公開するサービス名は `com.seshop.BookInfo` とし、サービスプロバイダ（実装クラス）として `com.seshop.impl.TeckBookInfo` と `com.seshop.impl.CookBookInfo` を登録する

##### client モジュール

- lib モジュールを必要とする
- `com.seshop.BookInfo` サービスを使用する

上記により選択肢Dが正しいです。

#### 問題 52 正解 : B

---

ServiceLoaderクラスのload()メソッドを呼び、ServiceLoaderオブジェクトを取得します。その際、load()メソッドの引数にはサービス名となるクラスオブジェクトを指定します。

#### 問題 53 正解 : D、E

---

jlinkは、必要なモジュールのみで構成されたラインタイム(実行環境)イメージを作成するコマンドです。したがって、選択肢A、Bは誤りで、選択肢Dは正しいです。JDKにjlinkコマンドが提供されており個別のダウンロードは不要のため、選択肢Cは誤りです。また、jlink実行時に、任意で--launcherオプションを使用してランチャーファイル(起動ファイル)を作成することも可能であるため、選択肢Eは正しいです。

#### 問題 54 正解 : C

---

ボトムアップ移行では、最下位のモジュールから段階的にボトムアップで移行します。無名モジュールはクラスパス上に配置されていますが、モジュール化したものからモジュールパス上に配置します。

#### 問題 55 正解 : C

---

①のすべてのスレッドが待機状態になってしまい、notify()メソッドを呼ぶスレッドがないという状況はデッドロックです。②の複数のスレッドが進まない処理を繰り返し続ける状況はライブロックです。

#### 問題 56 正解 : E

---

6行目はシーケンシャルストリームからfindAny()により、いずれかの要素を取得します。つまり、3、4、5いずれかの値がa変数に格納されます。7行目ではsynchronizedブロックを使用していますが、8～12行目の処理に影響はありません。8行目で新規にリストを作成し、9行目でパラレルストリームを取得しています。10行目でsorted()を使用していますが、パラレルストリームに対して行った場合は、順序は不定です。11行目でfindAny()によりいずれかの要素

を取得していますが、どの要素が取得されるか不定であるため、選択肢Eが正しいです。

#### 問題 57 正解：E、F

synchronized ブロック内に synchronized ブロックを記述することは可能であるため、選択肢A、Bは誤りです。2つのタスクは同じスレッドプールを使用して実行していますが、実行順序は不定であるため、選択肢C、Dは誤りで選択肢Eは正しいです。6行目では newFixedThreadPool(2) としているため、もし1つ目のスレッドが obj1 のロックを取得し、2つ目のスレッドが obj2 のロックを取得すると、そのままデッドロックとなります。したがって、選択肢Fは正しいです。

#### 問題 58 正解：F

11 行目では CyclicBarrier コンストラクタの第1引数に5と指定しています。これにより、スレッドの数は5本になるまで待機します。13、14行目では IntStream を parallel() メソッドによりパラレル処理としていますが、要素が3つしかないためスレッドが5本になることはありません。したがって、6行目の await() により待ち状態となります。

#### 問題 59 正解：A、E

並列処理での実行順序はOSのスケジューラに委ねられているため、選択肢Aは正しいです。パフォーマンスは、CPUの数、搭載メモリ量、タスクの量などによって大きく左右され、並列処理によって向上すると一概には言えないため、選択肢Bは誤りです。スレッドは共有のメモリ領域を利用するため、synchronized などによるロックの制御が必要になります。したがって、選択肢Cは誤りです。多くのリソースを必要とするアプリケーションでは、並列処理を使用することでパフォーマンスが向上する傾向にありますが、CPUが1つしか搭載されていないPCにおいても、処理内容によっては効率よい処理が見込めるため、選択肢Dは誤りで選択肢Eは正しいです。

#### 問題 60 正解：C

10 行目の execute() メソッドは戻り値が void であるため、Future は返しません



ん。したがって、10行目でコンパイルエラーです。もし、`execute()`の代わりに`submit()`メソッドを使用した場合は、コンパイル、実行ともに成功します。9行目で0から始まる要素を5つ用意し、10行目で`list`に追加する際に`++num`しています。インクリメントが前置であるため、1加算してからリストに追加されるので、実行結果は選択肢Bの「1 2 3 4 5」の出力となります。

#### 問題 61 正解：E

---

13行目によりスレッドの数は2本に固定されます。また、14行目では`CyclicBarrier`コンストラクタの第1引数に1と指定しているため、1本のスレッドが通過すれば`await()`による待機は解除されます。15、16行目により3つの要素をもつパラレルストリームが`exec()`メソッドを呼ぶため、`exec()`は3回実行されます。したがって、選択肢Eが正しいです。

#### 問題 62 正解：B、C、E

---

`Console`クラスには、`println()`と`print()`メソッド、および`out`変数は提供されていません。したがって、選択肢A、D、Fは誤りです。また、選択肢Bのように`writer()`メソッドで`PrintWriter`オブジェクトを取得することで、`println()`メソッドを使用することも可能です。

#### 問題 63 正解：E

---

`data`変数は`Foo`クラス内で初期化しており、かつ、コンストラクタで代入処理を行っていますが、`transient`であるためデシリアライズ後は`null`です。また、8行目では`Bar`クラスにより初期化ブロックで初期化していますが、コンストラクタ同様、初期化ブロックはデシリアライズ時は呼び出されません。したがって、デシリアライズ後の`data`変数は`null`です。

#### 問題 64 正解：D

---

問題文のコードのコンパイルは成功します。`BufferedInputStream`クラスは`mark`操作をサポートしているため実行可能であり、18行目の出力結果は「aWorlW」です。しかし、`FileInputStream`クラスは`mark`操作をサポートしていないため、実行時にエラーとなります。8行目の`foo()`の呼び出しに対し、15行目の`reset()`メソッドで`IOException`例外が発生します。

**問題 65 正解：F**

---

5行目の`path.subpath(1, 3)`により「doc/sample」が得られ、`getName(1)`により「sample」が得られます。`toAbsolutePath()`は絶対パスを返すため、現在のカレントディレクトリである「/tmp/miko」に「sample」を結合した「/tmp/miko/sample」を返します。

**問題 66 正解：B、C、D**

---

問題文のコードのコンパイルは成功します。`/doc/manual`以下にすでにtestディレクトリがある場合、`FileAlreadyExistsException`例外が発生するため、選択肢A、E、Fは誤りです。`/foo`は`/doc/manual`ディレクトリへのシンボリックリンクファイルであるため、testディレクトリが作成されると、`/doc/manual/test`、`/foo/test`のいずれのパスでもアクセスが可能です。

**問題 67 正解：G**

---

問題文のコードのコンパイルは成功します。7～12行目ではシンボリックファイルのみ検索し、パスを文字列に変換後リスト格納し、リストからストリームを取得し、拡張子がtxtのものをフィルタして出力しています。しかし、7行目の`find()`メソッドの第2引数で探索対象となる階層が0となっているため、`/bar`にアクセスしますが、それ以上の探索は行われなため、実行結果は何も表示されません。したがって、選択肢Gは正しいです。もし、0ではなく1を指定していた場合、`/bar`内(サブディレクトリは含まず)を探索するため、該当するファイルがあれば表示します。

**問題 68 正解：A、B**

---

7行目の`find()`メソッドは、第1引数は探索対象のルートとなるPathオブジェクト、第2引数は探索対象となる階層の深さです。これはint型で指定する必要があります。また、第3引数はBiPredicateインタフェース(メソッドは「`boolean test(T t, U u)`」)です。`test()`メソッドの第1引数にはPathオブジェクト、第2引数にはBasicFileAttributesオブジェクトが必要となります。

**問題 69 正解：B、E**

---

JDBC4.0以降では、`Class.forName()`によるドライバのロード処理は不要

であるため、選択肢Aは誤りです。Class.forName()でドライバクラスが見つからない場合は、ClassNotFoundException例外がスローされます。また、DriverManager.getConnection()でConnectionが取得できない場合は、SQLException例外がスローされるため、選択肢B、Eが正しく、選択肢C、Dは誤りです。

---

**問題 70 正解：C**

5行目のsql文では、departmentテーブルのdept\_address列の値をtmpに更新します。ただし、whereによるレコードの絞り込みはしていないため、更新対象は全レコードとなります。問題文の前提では「departmentテーブルには5レコードが格納されている」とあるため、8行目のexecuteUpdate()の戻り値は5となります。

---

**問題 71 正解：E**

5、6行目のsql文では、departmentテーブルからdept\_code列のみ全レコードから取得します。9行目でResultSetを取得していますが、一度もnext()の呼び出しを行わずに、10行目でgetString()による取り出しを行っているため、実行時にSQLException例外が発生します。

---

**問題 72 正解：D**

beforeFirst()メソッドは先頭行の直前に移動し、afterLast()メソッドは最終行の直後に移動します。ともに戻り値はなくvoidです。したがって、14行目でコンパイルエラーとなります。

---

**問題 73 正解：C、D**

Localeクラスにcreate()、getLocale()の各メソッドは提供されていないため、選択肢A、B、Fは誤りです。なお、選択肢Eは誤りですが、「Locale obj = Locale.US;」のように定数を使用したLocaleオブジェクトの取得は可能です。

また、次のコード例でLocale.Builderクラスを使用したLocaleオブジェクトの取得も確認してください。

```
Locale lc = new Locale.Builder().setLanguage("sr")
```

```
.setScript("Latn").setRegion("RS").build());
```

#### 問題 74 正解：C

プロパティファイルで用意していたリソースを、ListResourceBundleでは各リソース（キーと値のペア）をObject型の配列で用意することで同等の処理が可能です。したがって、選択肢Cは正しいです。プロパティファイルは複数配置することが可能です。ファイル名は基底名、言語コード、国コードの組み合わせとし、拡張子は.propertiesとします。したがって、選択肢A、Eは誤りです。なお、Javaクラスファイルとプロパティファイルは同梱しても問題ありません。したがって、選択肢Dは誤りです。ListResourceBundleではリソースとしてString以外のデータ型も扱えますが、プロパティファイルではすべてString型として読み込まれます。したがって、選択肢Bは誤りです。

#### 問題 75 正解：E

ResourceBundleクラスにはkeys()メソッドは提供されていないため、選択肢A、Bは誤りです。stream()も提供されていないため、選択肢C、Dは誤りです。選択肢Fはmap(k -> k)としており、キーのみを取得して出力するため誤りです。

#### 問題 76 正解：B

0および#は数字を表しますが、#はゼロの場合は表示されません。patternのうち、フォーマット対象となる部分を抜粋すると以下のとおりです。

```
#,000.0#

005.21
008.49
1,234.0
```

よって、実行結果は選択肢Bです。なお、問題文のコードで、9行目でmapToObj()メソッドを使用している点に注意してください。処理としてはDouble値をformat()によりフォーマットすることで、戻り値はStream<String>型となります。引数と戻り値の型が異なるため、mapToObj()メソッドを使用しています。もし、map()を使用しているとコンパイルエラーとなります。

**問題 77 正解：A**

6行目のp1は「foo/./x.txt」、7行目のp2は「foo/./bar/./x.txt」です。どちらのパスも指し示すのは「foo/x.txt」であるため、8行目のisSameFile()はtrueを返します。しかし、パス文字列は異なるため、9行目のequals()はfalseを返しますが、10行目のnormalize()後のequals()はtrueを返します。

**問題 78 正解：A**

ユーザが入力した値は、同じ意味の文字などを、ルールに基づいて統一的に扱えるように変形してから検査すべきです。Javaでは、Normalizerクラスのnormalize()メソッドで文字列を正規化することができます。

**問題 79 正解：E**

- A. PreparedStatementには、検索系処理用のexecuteQuery()、更新系処理用のexecuteUpdate()の各メソッドが用意されているため誤りです。
- B. JDBCに対応しているデータベースでは、PreparedStatementは利用可能であるため誤りです。
- C. INパラメータに対し、setterメソッドを記述していない場合は、実行時にSQLExceptionが発生します。
- D. setString()メソッドでINパラメータに値を設定する際、値が文字列であるためコンパイルでは検知できません。列名が含まれると、実行時エラーとなります。
- E. 「SELECT dept\_name FROM department」といったINパラメータを使用しないSQL文も実行可能です。その場合、setterメソッドは不要になります。

**問題 80 正解：D**

- A. add()メソッドを呼び出す前に、num1とnum2の桁チェックは行うべきです。
- B. BigIntegerクラスは、基本データ型では扱えない大きい値を扱うことができます。
- C. MathクラスのaddExact(long x, long y)メソッドは、引数の合計を返します。その結果がlongをオーバーフローする場合は例外をスローするため、

例外処理を適切に行うことでオーバーフローへの対応が可能です。

- D. 問題文の num1 と num2 は、基本データ型の long 型であるため、上位の整数型はありません。よってアップキャストができないため、誤りです。