

- 1. 次のプログラムの空欄に入るコードとして、正しいものを選びなさい。(1つ選択)

```
1. import java.io.*;
2.
3. public class Main {
4.     public static void main(String[] args) {
5.         try {
6.             test(0);
7.         } 
8.     }
9.     private static void test(int num) throws IOException,
10.         IndexOutOfBoundsException {
11.         if (num < 10) {
12.             throw new FileNotFoundException();
13.         } else {
14.             throw new IndexOutOfBoundsException();
15.         }
16.     }
```

- A. `catch (FileNotFoundException e) { }`  
`catch (IndexOutOfBoundsException e) { }`  
B. `catch (IndexOutOfBoundsException e) { }`  
`catch (FileNotFoundException e) { }`  
C. `catch (FileNotFoundException | IOException e) { }`  
D. `catch (IOException e) { }`  
E. `catch (FileNotFoundException | IndexOutOfBoundsException e) { }`

⇒ P554

- 2. immutableなオブジェクトに関する説明として、正しいものを選びなさい。(3つ選択)

- A. どのメソッドもオーバーライドできない  
B. どのメソッドでもフィールドの値を変更しない  
C. すべてのフィールドをpublicにする  
D. すべてのフィールドをfinalにする

⇒ P554

- 3. 次のプログラムのコンパイルエラーを修正する方法として、正しいものを選びなさい。(1つ選択)

```
1. public class Main {
2.     public static void main(String[] args) {
3.         int a = 10;
4.         float b = 10.0f;
5.         double c = 20;
6.         a = b;
7.         b = a;
8.         c = b;
9.         c = a;
10.     }
11. }
```

- A. 6行目を「`a = (int) b;`」に置き換える  
B. 7行目を「`b = (float) a;`」に置き換える  
C. 8行目を「`c = (double) b;`」に置き換える  
D. 9行目を「`c = (double) a;`」に置き換える

⇒ P555

□ 4. 次のプログラムの説明として、正しいものを選びなさい。(3つ選択)

```
1. import java.util.ArrayList;
2. import java.util.List;
3.
4. public class Trash {
5.     List<String> list = new ArrayList<>();
6.     protected Trash(String val) {
7.         add(val);
8.     }
9.     public boolean test(String val) {
10.         return true;
11.     }
12.     public void delete(String val) {
13.         list.remove(val);
14.     }
15.     public final void add(String val) {
16.         list.add(val);
17.     }
18. }
```

```
1. public class CacheTrash extends Trash {
2.     private String tmp = null;
3.     public CacheTrash() {
4.         super(null);
5.     }
6.     public boolean test(String val) {
7.         return list.contains(val);
8.     }
9.     public void delete(String val) {
10.         this.tmp = val;
11.     }
12.     public void flush() {
13.         super.delete(tmp);
14.     }
15. }
```

- A. CacheTrashクラスのtestメソッドは常にtrueを返す
- B. CacheTrashクラスのdeleteメソッドは、Trashクラスのdeleteメソッドを呼び出す

- C. CacheTrashクラスのaddメソッドは、引数をlistに追加する
- D. Trashクラスのlistは、Trashクラスのすべてのサブクラスで参照できる
- E. CacheTrashクラスのflushメソッドは、Trashクラスのdeleteメソッドを呼び出す
- F. CacheTrashクラスのコンストラクタは、nullをlistに追加する

⇒ P555

□ 5. 次のプログラムをコンパイルし、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample {
2.     private String value;
3.     public void setValue(String value) {
4.         value = "Hello, " + value;
5.     }
6.     public String toString() {
7.         return value;
8.     }
9. }
```

```
1. public class Main {
2.     public static void main(String[] args) {
3.         Sample s = new Sample();
4.         s.setValue("Java");
5.         System.out.println(s.toString());
6.     }
7. }
```

- A. 「Hello, Java」と表示される
- B. 「Hello, null」と表示される
- C. 「null」と表示される
- D. 実行時に例外がスローされる

⇒ P556

- 6. 次のプログラムを確認し、Bクラスの空欄への記述に関する説明として、正しいものを選びなさい。(1つ選択)

```
1. package ex06.a;
2.
3. public class A {
4.     String a = "A";
5.     protected String b = "B";
6.     public String c = "C";
7. }
```

```
1. package ex06.b;
2.
3. import ex06.a.A;
4.
5. public class B extends A {
6.     public void test() {
7.         
8.     }
9. }
```

- A. 「new B().b;」と「new B().c;」の両方でコンパイルエラーが発生する
- B. 「new B().b;」のみでコンパイルエラーが発生する
- C. 「new B().a;」のみでコンパイルエラーが発生する
- D. 「new B().a;」と「new B().b;」の両方でコンパイルエラーが発生する

⇒ P556

- 7. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample {
2.     public static void main(String[] args) {
3.         short a = 97;
4.         int b = 98;
5.         System.out.print((char) a + " ");
6.         System.out.print((char) b);
7.     }
8. }
```

- A. 「a b」と表示される
- B. 「a」と表示されたあとに例外が発生する
- C. コンパイルエラーが発生する
- D. ClassCastExceptionがスローされる

⇒ P557

- 8. BクラスからAクラスのtestメソッドを呼び出すために、Bクラスの空欄に入るコードとして正しいものを選びなさい。(1つ選択)

```
1. public class A {
2.     public void test() {
3.         // do something
4.     }
5. }
```

```
1. public class B extends A {
2.     public void test() {
3.         
4.     }
5. }
```

- A. this.test();
- B. public.test();
- C. new.test();
- D. super.test();

⇒ P558

- 9. 次のプログラムを確認し、コンパイルエラーとなる理由を選びなさい。  
(1つ選択)

```
1. public class A {
2.     private String value;
3.     public String getValue() {
4.         return value;
5.     }
6.     public void setValue(String value) {
7.         this.value = value;
8.     }
9.     public String toString() {
10.        return getValue();
11.    }
12. }
```

```
1. public class B extends A {
2.     public void setValue(String value) {
3.         super.setValue "[" + value + "]";
4.     }
5.     public void setValue(String... values) {
6.         setValue(test(values));
7.     }
8.     private String test(String... values) {
9.         var sb = new StringBuilder();
10.        for (String value : values) {
11.            sb.append(value + ", ");
12.        }
13.        return sb.toString();
14.    }
15. }
```

```
1. public class Main {
2.     public static void main(String[] args) {
3.         A a = new B();
4.         a.setValue(args);
5.     }
6. }
```

- A. BのインスタンスをA型で扱ったため
- B. AクラスのtoStringメソッドをBクラスでオーバーライドしていないため
- C. Bクラスのtestメソッドがprivateであるため
- D. BクラスからAクラスのvalueにアクセスできないため
- E. AクラスにはsetValue(String... values)というメソッドがないため

⇒ P558

- 10. 次のプログラムを確認し、Testクラスの何行目でコンパイルエラーが発生するかを選びなさい。(2つ選択)

```
public class Sample {
    String name;
}
```

```
1. import java.util.ArrayList;
2.
3. public class Test {
4.     var sample = new ArrayList<Sample>();
5.     public var test() {
6.         var sample = new Sample();
7.         var samples = new ArrayList<>();
8.         samples.add("Java");
9.         samples.add("Lambda");
10.        for (var s : samples) {
11.            System.out.println(s);
12.        }
13.    }
14. }
```

- A. 5行目
- B. 10行目
- C. 7行目
- D. 6行目
- E. 4行目

⇒ P559



- 11. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Main {
2.     private static int num;
3.     static {
4.         num = 10;
5.     }
6.     static {
7.         num = 20;
8.     }
9.     static void test(int num) {
10.         num = num * num;
11.     }
12.     public static void main(String[] args) {
13.         test(num);
14.         System.out.println(num);
15.     }
16. }
```

- A. 10が表示される
- B. 20が表示される
- C. 100が表示される
- D. 400が表示される
- E. コンパイルエラーが発生する

⇒ P559

- 12. 次のコードのうち、正常にコンパイルされるものを選びなさい。(2つ選択)

- A. var a = {1.0, 2.0, 3.0, 4.0};
- B. var b = new Double{1.0, 2.0, 3.0, 4.0};
- C. var c = new Float[] {1.0F, 2.0F, 3.0F, 4.0F};
- D. var d = new double[] {1.0, 2.0, 3.0, 4.0};

⇒ P560

- 13. 次のプログラムの説明として、正しいものを選びなさい。(1つ選択)

```
1. package test;
2.
3. public class Sample {
4.     int a;
5.     private int b;
6.     protected int c;
7.     public int d;
8. }
```

```
1. import test.Sample;
2.
3. public class Main extends Sample {
4.     public static void main(String[] args) {
5.         Main s = new Main();
6.
7.     }
8. }
```

- A. Mainクラスからはaとdの両方にアクセスできる
- B. Mainクラスからはdのみアクセスできる
- C. Mainクラスからはcとdの両方にアクセスできる
- D. Mainクラスからはa、c、およびdにアクセスできる

⇒ P560

- 14. 次のプログラムをコンパイルし、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. import java.util.Collection;
2.
3. public class A {
4.     private Collection collection;
5.     public void set(Collection collection) {
6.         this.collection = collection;
7.         System.out.println("A");
8.     }
9. }
```

```
1. import java.util.Collection;
2.
3. public class B extends A {
4.     public void set(Collection collection) {
5.         super.set(collection);
6.         System.out.println("B");
7.     }
8. }
```

```
1. import java.util.Collection;
2.
3. public class C extends B {
4.     @Override
5.     public void set(Collection collection) {
6.         super.super.set(collection);
7.         System.out.println("C");
8.     }
9. }
```

```
1. import java.util.List;
2.
3. public class Main {
4.     public static void main(String[] args) {
5.         new C().set(List.of(1, 2, 3));
6.     }
7. }
```

- A. 「A」「B」「C」と表示される
- B. 「A」「B」と表示される
- C. 「A」と表示される
- D. コンパイルエラーが発生する
- E. 実行時に例外がスローされる

⇒ P561

- 15. 次のプログラムを確認してください。

```
1. public class A {
2.     private int num;
3.     public A(int num) {
4.         this.num = num;
5.     }
6. }
```

次のプログラムの空欄に入るコードとして、正しいものを選びなさい。(1つ選択)

```
1. public class B extends A {
2.     private String val;
3.     public B(String val, int num) {
4.         
5.     }
6. }
```

- A. this(num);
- B. super(num);
- C. this(val);
- D. super(val);
- E. super();

⇒ P561

- 16. 次のプログラムの説明として、正しいものを選びなさい。(1つ選択)

```

1. package ex16;
2.
3. import java.util.*;
4.
5. public class Sample {
6.     private List<String> list = List.of("A", "B", "C");
7.     public List<String> getList() {
8.         return list;
9.     }
10. }
```

```

1. package ex16;
2.
3. public class Main {
4.     public static void main(String[] args) {
5.         Sample s = new Sample();
6.         for (String str : s.getList()) {
7.             System.out.println(str);
8.         }
9.     }
10. }
```

- A. 2つのクラスは同じパッケージに属するので、Mainクラスでjava.util.Listをインポートする必要はない
- B. java.util.Listのみがインポートされる
- C. コンパイルするには、Mainクラスでjava.util.Listをインポートしなければならない
- D. java.utilパッケージのすべてのクラスやインタフェースがSampleクラスにインポートされる

⇒ P561

- 17. 次のコードのうち、正常にコンパイルされるものを選びなさい。(2つ選択)

- A. `int[] array;`  
`array = new int[] {};`
- B. `int[] array2 = new int[3];`  
`array2 = {1, 2, 3};`
- C. `Double[] array3 = new Double[3] {};`
- D. `int[] array4 = {array4[0], array4[1]};`
- E. `Double[] array5 = {null, 1.0};`

⇒ P562

- 18. 次のプログラムを実行し、「A:B」と表示したい。Mainクラスの空欄に入るコードとして正しいものを選びなさい。(1つ選択)

```

1. public class Sample {
2.     static String header = "A:";
3.     private String value = "B";
4.     public static String getValue() {
5.         return new Sample().value;
6.     }
7. }
```

```

1. public class Main {
2.     public static void main(String[] args) {
3.         Sample s = new Sample();
4.         
5.     }
6. }
```

- A. `System.out.println(Sample.getValue + s.header);`
- B. `System.out.println(Sample.header + Sample.value);`
- C. `System.out.println(new Sample().header + new Sample().value);`
- D. `System.out.println(s.header + Sample.value);`
- E. `System.out.println(Sample.header + Sample.getValue());`
- F. `System.out.println(s.header + s.value);`

⇒ P563

- 19. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Main {
2.     public static void main(String[] args) {
3.         int a = 10;
4.         int b = 20;
5.         int c = b += a / 5;
6.         System.out.println(a + b + c);
7.     }
8. }
```

- A. 52が表示される
- B. 54が表示される
- C. 38が表示される
- D. 46が表示される

⇒ P563

- 20. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Main {
2.     public static void main(String[] args) {
3.         try {
4.             String[] array = new String[5];
5.             array[1] = "A";
6.             array[2] = "B";
7.             array[3] = "C";
8.             for (String str : array) {
9.                 System.out.println(str);
10.            }
11.        } catch (Exception e) {
12.            System.out.println("Error");
13.        }
14.    }
15. }
```

- A. 「A」「B」「C」と表示される
- B. 「null」「A」「B」「C」「null」と表示される
- C. 「Error」と表示される
- D. コンパイルエラーが発生する

⇒ P564

- 21. 次のプログラムをコンパイルし、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample {
2.     private int num;
3.     private int test() {
4.         for (int i = 0; i < 3; i++) {
5.             num += i;
6.         }
7.         return num;
8.     }
9.     public static void main(String[] args) {
10.        Sample s = new Sample();
11.        int num = s.test();
12.        s.test();
13.        System.out.println(num);
14.    }
15. }
```

- A. 3が表示される
- B. 6が表示される
- C. 9が表示される
- D. 実行時に例外がスローされる

⇒ P565

- 22. java.langパッケージを含むJava SEの基本的なAPIが提供されているモジュールとして正しいものを選びなさい。(1つ選択)

- A. classes
- B. jdk.base
- C. java.se
- D. java.base

⇒ P565



- 23. 次のプログラムをコンパイルし、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. package ex23.a;
2.
3. public class A {
4.     public int num = 10;
5. }
```

```
1. package ex23.b;
2.
3. import ex23.a.A;
4.
5. public class B extends A {
6.     int num = 20;
7. }
```

```
1. package ex23;
2.
3. import ex23.a.A;
4. import ex23.b.B;
5.
6. public class Main {
7.     public static void main(String[] args) {
8.         A a = new B();
9.         System.out.println(a.num);
10.    }
11. }
```

- A. Bクラスでコンパイルエラーが発生する
- B. Mainクラスでコンパイルエラーが発生する
- C. 10が表示される
- D. 20が表示される

⇒ P565

- 24. publicで修飾されたメソッドをオーバーライドするときに利用できるアクセス修飾子として、正しいものを選びなさい。(1つ選択)

- A. public
- B. public、protected
- C. public、protected、デフォルト
- D. 制限はない

⇒ P566

- 25. 次のプログラムをコンパイルし、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. package ex25.a;
2.
3. class Sample {
4.     public String value;
5.     public Sample(String value) {
6.         this.value = value;
7.     }
8. }
```

```
1. package ex25.b;
2. import ex25.a.Sample;
3.
4. public class Main {
5.     public static void main(String[] args) {
6.         Sample s = new Sample("Hello Java");
7.         System.out.println(s.value);
8.     }
9. }
```

- A. 「null」と表示される
- B. 「Hello Java」と表示される
- C. 何も表示されない
- D. コンパイルエラーが発生する
- E. 実行時に例外がスローされる

⇒ P567

- 26. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Main {
2.     int num;
3.     private static void test() {
4.         num++;
5.         System.out.println(num);
6.     }
7.     public static void main(String[] args) {
8.         Main.test();
9.         Main.test();
10.    }
11. }
```

- A. 「1」「2」と表示される
- B. 1が2回表示される
- C. testメソッドでコンパイルエラーが発生する
- D. mainメソッドでコンパイルエラーが発生する
- E. 実行時に例外がスローされる

⇒ P567

- 27. 次のプログラムをコンパイルし、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample {
2.     public static void main(String[] args) {
3.         String str = "abcde";
4.         str.replace('c', 'x');
5.         str.substring(2, 4);
6.         System.out.println(str);
7.     }
8. }
```

- A. 「abxde」と表示される
- B. 「abcde」と表示される
- C. 「cde」と表示される
- D. 「xde」と表示される

⇒ P567

- 28. 次のプログラムを実行して、1、2、3と順に表示したい。空欄に入るコードとして、正しいものを選びなさい。(1つ選択)

```
1. import java.util.List;
2.
3. public class Main {
4.     public static void main(String[] args) {
5.         List<Integer> list = List.of(1, 2, 3);
6.         
7.     }
8. }
```

- A. list.forEach(() -> System.out.println(x));
- B. list.forEach(x -> System.out.println(x));
- C. list.forEach(x -> { return x });
- D. list.forEach(list);

⇒ P568

- 29. 次のプログラムの説明として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample {
2.     private final int num;
3.     public Sample(String str) {
4.         this(Integer.parseInt(str));
5.     }
6.     public Sample(int num) {
7.         this.num = num;
8.     }
9.     public Sample() {}
10.    public int getNum() {
11.        return this.num;
12.    }
13. }
```

- A. 2行目でコンパイルエラーが発生する
- B. 4行目でコンパイルエラーが発生する
- C. 6行目でコンパイルエラーが発生する
- D. 9行目でコンパイルエラーが発生する

⇒ P568

- 30. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. interface Function {  
2.     void process();  
3. }
```

```
1. class Sample implements Function {  
2.     protected void process() {  
3.         System.out.println("A");  
4.     }  
5. }
```

```
1. public class Main extends Sample {  
2.     public void process() {  
3.         System.out.println("B");  
4.     }  
5.     public static void main(String[] args) {  
6.         Sample s = new Main();  
7.         Sample s2 = new Sample();  
8.         test(s);  
9.         test(s2);  
10.    }  
11.    public static void test(Function f) {  
12.        f.process();  
13.    }  
14. }
```

- A. 「A」「A」と表示される
- B. 「B」「A」と表示される
- C. Sampleクラスでコンパイルエラーが発生する
- D. Mainクラスでコンパイルエラーが発生する
- E. SampleクラスとMainクラスの両方でコンパイルエラーが発生する

⇒ P569

- 31. 次のプログラムを確認し、Bクラスの空欄に入るコードとして、正しいものを選びなさい。(1つ選択)

```
1. import java.util.Collection;  
2.  
3. public class A {  
4.     private Collection collection;  
5.     public void set(Collection collection) {  
6.         this.collection = collection;  
7.     }  
8. }
```

```
1. import java.util.Map;  
2.  
3. public class B extends A {  
4.     public void set(Map<String, String> map) {  
5.           
6.     }  
7. }
```

- A. set(map.values());
- B. super.set(List<String> map);
- C. set(super.map);
- D. map.forEach((k, v) -> set(v));

⇒ P569

- 32. 次の説明のうち、正しいものを選びなさい。(1つ選択)

- A. Javaでは、クラスとインタフェースは多重継承できない
- B. インタフェースで、抽象メソッドをオーバーライドして実装を提供できる
- C. 抽象メソッドをオーバーライドした抽象メソッドでは、戻り値型を変更することはできない
- D. インタフェースを継承したインタフェースでは、抽象メソッドの実装を提供しなければならない

⇒ P570



33. 次のプログラムをコンパイルし、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample {  
2.     public void test(Object[] val) {  
3.         System.out.println("A");  
4.     }  
5.     public void test(long[] val) {  
6.         System.out.println("B");  
7.     }  
8.     public void test(Object val) {  
9.         System.out.println("C");  
10.    }  
11. }
```

```
1. public class Main {  
2.     public static void main(String[] args) {  
3.         new Sample().test(new int[3]);  
4.     }  
5. }
```

- A. Aが表示される
- B. Bが表示される
- C. Cが表示される
- D. Sampleクラスでコンパイルエラーが発生する
- E. Mainクラスでコンパイルエラーが発生する

⇒ P570

34. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Main {  
2.     public static void main(String[] args) {  
3.         int[][] array = new int[2][4];  
4.         array[0] = new int[]{1, 2, 3, 4};  
5.         array[1] = new int[]{1, 2};  
6.         for (int[] a : array) {  
7.             for (int b : a) {  
8.                 System.out.print(b);  
9.             }  
10.            System.out.println();  
11.        }  
12.    }  
13. }
```

- A. 「1234」「12」と表示される
- B. 「12」「12」と表示される
- C. 「12」「1200」と表示される
- D. コンパイルエラーが発生する
- E. 実行時に例外がスローされる

⇒ P571

35. 次の説明のうち、もっとも適切なものを選びなさい。(1つ選択)

- A. Javaの実行環境は、異なるプログラミング言語のコンパイラによってコンパイルされた実行可能ファイルも処理できる
- B. コンパイラはリフレクションを使い、メソッドがターゲットプラットフォームのリソースによってサポートされているかどうかをテストする
- C. コンパイラは、特定のプラットフォームで実行するためにクラスを前処理する
- D. 実行時に、クラスは必要に応じてロードされる

⇒ P572



- 36. 次のプログラムをコンパイルし、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. import java.util.function.Consumer;
2.
3. public class Main {
4.     public static void main(String[] args) {
5.         final String val = "Hello, Lambda";
6.         Consumer<String> func = (val) -> System.out.println(val);
7.         func.accept("Hello, Java");
8.     }
9. }
```

- A. 「Hello, Lambda」と表示される
- B. 「Hello, Java」と表示される
- C. 何も表示されない
- D. コンパイルエラーが発生する
- E. 実行時に例外がスローされる

⇒ P573

- 37. 次のプログラムをコンパイルし、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Main {
2.     public static void main(String[] args) {
3.         String str;
4.         test(str);
5.     }
6.     private static void test(String str) {
7.         System.out.println(str);
8.     }
9. }
```

- A. 「null」と表示される
- B. 何も表示されない
- C. コンパイルエラーが発生する
- D. 実行時に例外がスローされる

⇒ P573

- 38. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Main {
2.     public static void main(String[] args) {
3.         int result = 30 - 12 / (2 * 5) + 1;
4.         System.out.println(result);
5.     }
6. }
```

- A. 2が表示される
- B. 3が表示される
- C. 28が表示される
- D. 29が表示される
- E. 30が表示される

⇒ P574

- 39. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample {
2.     public static void main(String[] args) {
3.         int[] array = {1, 2, 3, 4, 5};
4.         int key = 3;
5.         int cnt = 0;
6.         for (int i : array) {
7.             if (i != key) {
8.                 continue;
9.                 cnt++;
10.            }
11.        }
12.        System.out.println(cnt);
13.    }
14. }
```

- A. 3
- B. 2
- C. 1
- D. コンパイルエラーが発生する

⇒ P574

- 40. 次のプログラムをコンパイルし、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Main {
2.     public static void main(String[] args) {
3.         int num = 0;
4.         System.out.println(num++ + num + "," + ++num + num);
5.     }
6. }
```

- A. 「1,22」と表示される
- B. 「2,22」と表示される
- C. 「2,4」と表示される
- D. コンパイルエラーが発生する

⇒ P574

- 41. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Main {
2.     public static void main(String[] args) {
3.         String[] array = {"A", "B", "C", "D", "E"};
4.         for (String str : array) {
5.             if ("B".equals(str)) {
6.                 continue;
7.             }
8.             System.out.println(str);
9.             if ("C".equals(str)) {
10.                 break;
11.             }
12.         }
13.     }
14. }
```

- A. 「A」「C」と表示される
- B. 「A」「B」「C」と表示される
- C. 「C」「D」と表示される
- D. 「C」と表示される
- E. コンパイルエラーが発生する

⇒ P575

- 42. 次のプログラムをコンパイルし、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample implements AutoCloseable {
2.     @Override
3.     public void close() throws Exception {
4.         System.out.println("C");
5.     }
6. }
```

```
1. public class Main {
2.     public static void main(String[] args) {
3.         try (Sample s = new Sample()) {
4.             throw new Exception();
5.         } catch (Exception e) {
6.             System.out.println("A");
7.         } finally {
8.             System.out.println("B");
9.         }
10.     }
11. }
```

- A. 「A」「B」「C」と表示される
- B. 「A」「C」「B」と表示される
- C. 「C」「A」「B」と表示される
- D. コンパイルエラーが発生する

⇒ P576

- 43. モジュールシステムに関する説明として、正しいものを選びなさい。(2つ選択)

- A. module-info.javaは、module-path内のどこにでも配置できる
- B. java.baseをエクスポートすると、すべてのコアパッケージがエクスポートされる
- C. 利用するモジュールは、module-info.java内で宣言されていなければならない
- D. デフォルトで、モジュールは、モジュールが動作するディレクトリ内にはアクセスができる
- E. module-info.javaは空にはできない

⇒ P576

- 44. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. class A {  
2.     public A() {  
3.         System.out.print("A");  
4.     }  
5. }
```

```
1. class B extends A {  
2.     public B() {  
3.         System.out.print("B");  
4.     }  
5. }
```

```
1. class C extends B {  
2.     public C() {  
3.         System.out.print("C");  
4.     }  
5. }
```

```
1. public class Sample {  
2.     public static void main(String[] args) {  
3.         new C();  
4.     }  
5. }
```

- A. 「ABC」と表示される
- B. 「C」と表示される
- C. 「CBA」と表示される
- D. コンパイルエラーが発生する

⇒ P577

- 45. 次のプログラムの「// insert code here」に入るコードとして、正しいものを選びなさい。(3つ選択)

```
public abstract class Item {  
    int num;  
    // insert code here  
}
```

- A. public void print();
- B. public int calculate() {  
 return num \* 2;  
}
- C. public abstract int getNum() {  
 return num;  
}
- D. public abstract int test();
- E. public static void sample() {  
 // some codes  
}

⇒ P578

- 46. 次のプログラムをコンパイルし、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample {  
2.     public static void main(String[] args) {  
3.         int num = 2;  
4.         int result = (int) Math.round(Math.pow(3.1, num));  
5.         System.out.println(result);  
6.     }  
7. }
```

- A. 8が表示される
- B. 9が表示される
- C. 10が表示される
- D. 3が表示される
- E. コンパイルエラーが発生する
- F. 実行時に例外がスローされる

⇒ P578

- 47. 次のAクラスを継承したサブクラスに定義できるメソッドとして、正しいものを選びなさい。(1つ選択)

```
1. public class A {
2.     public Number test() {
3.         return 100;
4.     }
5. }
```

- A. `@Override`  
`public Integer test() {`  
 `return 0;`  
`}`
- B. `@Override`  
`public Number test(int num) {`  
 `return num;`  
`}`
- C. `@Override`  
`public void test() {`  
 `// do something`  
`}`
- D. `public Number test() throws Exception {`  
 `return 0;`  
`}`

⇒ P579

- 48. 次のプログラムを実行し、「sample」と表示したい。正しいコマンドを選びなさい。(2つ選択)

```
1. public class Sample {
2.     public static void main(String[] args) {
3.         System.out.println("sample");
4.     }
5. }
```

- A. `java Sample.class`  
 B. `javac Sample.java`  
`java Sample.class`  
 C. `java Sample`  
 D. `javac Sample.java`  
`java Sample`  
 E. `java Sample.java`

⇒ P579

- 49. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample {
2.     int a, b;
3.     public Sample(int a, int b) {
4.         init(a, b);
5.     }
6.     public void init(int a, int b) {
7.         this.a = a * a;
8.         this.b = b * b;
9.     }
10.    public static void main(String[] args) {
11.        int a = 2, b = 3;
12.        Sample s = new Sample(a, b);
13.        System.out.println(a + ", " + b);
14.    }
15. }
```

- A. 「4, 9」と表示される  
 B. 「0, 0」と表示される  
 C. 「2, 3」と表示される  
 D. コンパイルエラーが発生する

⇒ P580



50. 次のプログラムのコンパイルエラーを修正する方法として、正しいものを選びなさい。(2つ選択)

```
1. abstract class Sample {  
2.     protected void doProcess() {}  
3.     abstract void doTest();  
4. }
```

```
1. public class Test extends Sample {  
2.     void doProcess() {}  
3.     protected void doTest() {}  
4. }
```

- A. Sampleクラスの2行目のメソッドをpublicにする
- B. Sampleクラスの3行目のメソッドをpublicにする
- C. Testクラスの2行目のメソッドをpublicにする
- D. Testクラスの2行目のメソッドをprotectedにする
- E. Testクラスの3行目のメソッドをpublicにする

⇒ P580

51. 次のコードに記述されている変数valのデータ型として、正しいものを選びなさい。(1つ選択)

```
var list = new ArrayList<>();  
list.forEach(val -> System.out.println(val));
```

- A. String
- B. ArrayList
- C. Object
- D. var

⇒ P581

52. 次のプログラムを実行して「0」と表示したい。Mainクラスの空欄に入るコードとして、正しいものを選びなさい。(3つ選択)

```
1. class Test {  
2.     public int num;  
3.     public Test(int num) {  
4.         this.num = num;  
5.     }  
6.     public int getNum() {  
7.         return num;  
8.     }  
9.     public void modify(int x) {  
10.        num += x;  
11.    }  
12. }
```

```
1. public class Main {  
2.     public static void main(String[] args) {  
3.         Test t = new Test(100);  
4.           
5.         System.out.println(t.getNum());  
6.     }  
7. }
```

- A. this.num = 0;
- B. num = 0;
- C. t(0);
- D. t.num = 0;
- E. t.getNum() = 0;
- F. t.modify(0);
- G. t.modify(-t.num);
- H. t.modify(-t.getNum());

⇒ P581

53. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```

1. public class Main {
2.     private String val;
3.     private int num;
4.     public Main(int num) {
5.         this.num = num;
6.     }
7.     public Main() {
8.         this.val = "test";
9.         this.num = 10;
10.    }
11.    public static void main(String[] args) {
12.        Main m = new Main(20);
13.        System.out.println(m.val + ", " + m.num);
14.    }
15. }
```

- A. 「null, 20」と表示される
- B. 「test, 20」と表示される
- C. コンパイルエラーが発生する
- D. 実行時に例外がスローされる

⇒ P582

54. 次の中から、インタフェースに記述できるコードを選びなさい。(2つ選択)

- A. `public String methodA();`
- B. `final void methodB();`
- C. `public void methodC() {  
    System.out.println("C");  
}`
- D. `public int d;`
- E. `public abstract void methodE();`
- F. `private abstract void methodF();`
- G. `final void methodG() {  
    System.out.println("G");  
}`

⇒ P583

55. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```

1. interface A {
2.     public void sample();
3. }
```

```

1. interface B extends A {
2.     public void test();
3. }
```

```

1. abstract class C implements B {
2.     public void test() {
3.         System.out.println("A");
4.     }
5. }
```

```

1. class D extends C {
2.     public void test() {
3.         System.out.println("B");
4.     }
5. }
```

```

1. public class Main {
2.     public static void main(String[] args) {
3.         C c = new D();
4.         c.test();
5.     }
6. }
```

- A. 「A」と表示される
- B. 「B」と表示される
- C. Bインタフェースでコンパイルエラーが発生する
- D. Cクラスでコンパイルエラーが発生する
- E. Dクラスでコンパイルエラーが発生する

⇒ P584

- 56. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Main {
2.     public static void main(String[] args) {
3.         int a = 4;
4.         int b = 8;
5.         int c = b += a / 2;
6.         System.out.println(a + ", " + b + ", " + c);
7.     }
8. }
```

- A. 「4, 10, 8」と表示される
- B. 「4, 10, 10」と表示される
- C. 「4, 10, 6」と表示される
- D. 「4, 12, 6」と表示される

⇒ P585

- 57. 次のプログラムの空欄に入るコードとして、正しいものを選びなさい。なお、ex57.b.Sampleクラスのパスは次のとおりとする。(1つ選択)

/bin/ex57/b/Sample.class

```
1. package ex57.a;
2.
3. 
4.
5. public class Main {
6.     public static void main(String[] args) {
7.         Sample s = new Sample();
8.     }
9. }
```

- A. import ex57.b.Sample;
- B. import ex57.b;
- C. import bin.ex57.b.Sample;
- D. import bin.ex57.b.\*;

⇒ P585

- 58. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample {
2.     public static void main(String[] args) {
3.         String a = "A";
4.         a = a.concat("B");
5.         String b = "C";
6.         a = a.concat(b);
7.         a.replace('C', 'D');
8.         a = a.concat(b);
9.         System.out.println(a);
10.    }
11. }
```

- A. 「ABCD」と表示される
- B. 「ACD」と表示される
- C. 「ABCC」と表示される
- D. 「ABD」と表示される
- E. 「ABDC」と表示される

⇒ P586

- 59. 次のプログラムをコンパイルし、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample {
2.     public static void main(String[] args) {
3.         int num = 10;
4.         Integer val = Integer.valueOf(10);
5.         System.out.println(num == val);
6.     }
7. }
```

- A. trueが表示される
- B. falseが表示される
- C. コンパイルエラーが発生する
- D. 実行時に例外がスローされる

⇒ P586

60. Bクラス内からAクラスにアクセスするコードとして、正しいものを選びなさい。(1つ選択)

```
package sample;
public class A {
    // ...
}
```

```
package sample.sub;
public class B {
    // ...
}
```

- A. String val = sample.A.getVal();
- B. import sample.\*;
- C. import sample.sub.\*;
- D. String val = A.getVal();

⇒ P587

61. 次のプログラムをコンパイルし、以下に示すコマンドで実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
> java Sample a b c
```

```
1. public class Sample {
2.     public static void main(String[] args) {
3.         for (int i = 0; i < args.length; i++) {
4.             System.out.println(args[i]);
5.             switch(args[i]) {
6.                 case "a":
7.                     continue;
8.                 case "b":
9.                     i--;
10.                    continue;
11.                default:
12.                    break;
13.            }
14.        }
15.    }
16. }
```

- A. 「a」「b」「c」と表示される
- B. 無限ループになる
- C. コンパイルエラーが発生する
- D. 実行時に例外がスローされる

⇒ P587

62. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample {
2.     public static void main(String[] args) {
3.         Test a = new Test();
4.         Test b = new Test();
5.         a.sample();
6.         b.sample();
7.         System.out.println(a.value + ", " + b.value);
8.     }
9. }
```

```
1. class Test {
2.     static int value = 0;
3.     int count = 0;
4.     public void sample() {
5.         while(count < 5) {
6.             count++;
7.             value++;
8.         }
9.     }
10. }
```

- A. 「10, 10」と表示される
- B. 「5, 5」と表示される
- C. 「5, 10」と表示される
- D. コンパイルエラーが発生する

⇒ P588



63. 次のプログラムを実行して「54321」と表示したい。空欄に入るコードとして正しいものを選びなさい。(2つ選択)

```
1. public class Main {
2.     public static void main(String[] args) {
3.         int[] array = {1, 2, 3, 4, 5};
4.         int x = array.length;
5.         
6.     }
7. }
```

- A. `while( 0 <= x ) {  
 System.out.println(array[x]);  
 x--;  
}`
- B. `do {  
 x--;  
 System.out.println(array[x]);  
} while( 0 <= x );`
- C. `do {  
 System.out.println(array[x]);  
 x--;  
} while( 0 <= x );`
- D. `while (0 < x) {  
 x--;  
 System.out.println(array[x]);  
}`
- E. `while (0 < x) {  
 System.out.println(array[--x]);  
}`

⇒ P589

64. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample {
2.     public static void main(String[] args) {
3.         var i = 3;
4.         var j = 2;
5.         i += (j * 2 + j) / i - 2;
6.         System.out.println(i);
7.     }
8. }
```

- A. 9が表示される  
B. 6が表示される  
C. 3が表示される  
D. 12が表示される

⇒ P590

65. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample {
2.     public static void main(String[] args) {
3.         int a = 2;
4.         int b = 1;
5.         if (a++ > ++b) {
6.             System.out.print("A ");
7.         } else {
8.             System.out.print("B ");
9.         }
10.        System.out.println(a + ":" + b);
11.    }
12. }
```

- A. 「A 2:1」と表示される  
B. 「A 3:2」と表示される  
C. 「B 2:1」と表示される  
D. 「B 3:2」と表示される

⇒ P591

66. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. class Item {  
2.     String name;  
3.     public Item(String name) {  
4.         this.name = name;  
5.     }  
6. }
```

```
1. public class Sample {  
2.     public static void main(String[] args) {  
3.         Item[] items = new Item[3];  
4.         items[1] = new Item("A");  
5.         items[2] = new Item("B");  
6.         for (Item item : items) {  
7.             System.out.println(item.name);  
8.         }  
9.     }  
10. }
```

- A. A  
B
- B. null  
A  
B
- C. コンパイルエラーが発生する
- D. `ArrayIndexOutOfBoundsException`がスローされる
- E. `NullPointerException`がスローされる

⇒ P591

67. Javaが高いパフォーマンスを実現していることと関連する説明として、もっとも適切なものを選びなさい。(1つ選択)

- A. Javaの標準ライブラリは、パイプライン処理を行うための組み込み関数を持っている
- B. Javaは頻繁に実行されるコードを監視して、実行を最適化する
- C. Javaはガベージコレクションをアプリケーションのコードよりも優先して処理する
- D. Javaは自動的にコードを並列処理する

⇒ P592

68. 次のAクラスを継承したサブクラスに定義できるメソッドとして、正しいものを選びなさい。(2つ選択)

```
1. import java.util.*;  
2.  
3. public class A {  
4.     public List<Number> test(Set<CharSequence> s) {  
5.         return null;  
6.     }  
7. }
```

- A. `public ArrayList<Integer> test(Set<String> s) { ... }`
- B. `public List<Integer> test(TreeSet<String> s) { ... }`
- C. `public List<Integer> test(Set<String> s) { ... }`
- D. `public List<Object> test(Set<CharSequence> s) { ... }`
- E. `public List<Integer> test(Set<CharSequence> s) { ... }`
- F. `public ArrayList<Number> test(Set<CharSequence> s) { ... }`

⇒ P592

- 69. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```

1. public class Main {
2.     public static void main(String[] args) {
3.         String[] array = {"A", "B"};
4.         int i = 0;
5.         while( i < array.length ) {
6.             int j = 0;
7.             do {
8.                 ++j;
9.             } while ( j < array[i].length() );
10.            System.out.println(array[i] + ":" + j);
11.            i++;
12.        }
13.    }
14. }
```

- A. A:2  
B:2
- B. A:1  
B:3
- C. A:1  
B:1
- D. 実行時に例外がスローされる

⇒ P593

- 70. 次のプログラムの「// insert code here」に入るコードとして、正しいものを選びなさい。(1つ選択)

```

public class Sample {
    public static void main(String[] args) {
        // insert code here
        array[0] = 2;
        array[1] = 3;
        System.out.println(array[0] + "," + array[1]);
    }
}
```

- A. `int[] array = new int[2];`
- B. `int[] array;`  
`array = int[2];`
- C. `int array = new int[2];`
- D. `int array[2];`

⇒ P594

- 71. 次のプログラムのコンパイルエラーを修正する方法として、正しいものを選びなさい。(2つ選択)

```

1. public class Main {
2.     public static void main(String[] args) {
3.         try {
4.             if (args.length == 0) {
5.                 sample(null);
6.             } else {
7.                 sample(args[0]);
8.             }
9.         } catch (RuntimeException e) {
10.            System.out.println("error");
11.        }
12.    }
13.    private static void sample(String str) {
14.        if (str == null) throw new Exception();
15.        throw new RuntimeException();
16.    }
17. }
```

- A. mainメソッドの宣言にthrows Exceptionを追加する
- B. sampleメソッドの宣言にthrows Exceptionを追加する
- C. mainメソッドとsampleメソッドの両方にthrows Exceptionを追加する
- D. sampleメソッドの宣言にthrows Exceptionを追加し、catchブロックの型をExceptionに変更する
- E. mainメソッドの宣言にthrows Exceptionを追加し、catchブロックの型をExceptionに変更する

⇒ P594

72. 次のプログラムをコンパイルし、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample {
2.     public static final int length = 3;
3.     public static void main(String[] args) {
4.         int[] array = new int[length];
5.         for (int i = 0; i < 3; i++) {
6.             System.out.print(array[i]);
7.         }
8.         System.out.println();
9.         Integer[] array2 = new Integer[length];
10.        for (int i = 0; i < 3; i++) {
11.            System.out.print(array2[i]);
12.        }
13.    }
14. }
```

- A. nullnullnull  
000
- B. 000  
000
- C. 000  
nullnullnull
- D. nullnullnull  
nullnullnull
- E. 実行時に例外がスローされる

⇒ P595

73. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Main {
2.     public static void main(String[] args) {
3.         System.out.println("result=" + 2 + 3 + 4);
4.         System.out.println("result=" + 2 + 3 * 4);
5.     }
6. }
```

- A. result=9  
result=24
- B. result=9  
result=20
- C. result=234  
result=212
- D. result=212  
result=212

⇒ P595

74. 次のコードを確認し、trueを返すコードとして正しいものを選びなさい。(1つ選択)

```
char[][] array1 = {{'a', 'b'}, {'c', 'd'}};
char[][] array2 = array1.clone();
char[] array3 = array1[1].clone();
```

- A. array1[1] == array3
- B. array1[1] == array3[1]
- C. array1[1] == array2[1]
- D. array1.equals(array2)

⇒ P596



75. 次のSampleクラスに、引数で受け取った値でフィールドvalueを変更するメソッドを追加したい。メソッドの定義として、正しいものを選びなさい。(1つ選択)

```
1. public class Sample {  
2.     private String value;  
3. }
```

- A. 

```
public void setValue(String value) {  
    value = value;  
}
```
- B. 

```
public String setValue(String str) {  
    String value = str;  
    return value;  
}
```
- C. 

```
public void setValue(String value) {  
    this.value = value;  
}
```
- D. 

```
public void setValue() {  
    this.value = value;  
}
```

⇒ P596

76. 次のプログラムを実行し、「HELLO」と表示したい。空欄に入るコードとして、正しいものを選びなさい。(2つ選択)

```
1. import java.util.function.Function;  
2.  
3. public class Main {  
4.     public static void main(String[] args) {  
5.         Function<String, String> func =   
6.         System.out.println(func.apply("hello"));  
7.     }  
8. }
```

- A. `String::toUpperCase();`
- B. `x -> x.toUpperCase();`
- C. `x::toUpperCase();`
- D. `x::toUpperCase();`
- E. `(String x) -> { return x.toUpperCase(); }`

⇒ P597

77. 次のプログラムの説明として、正しいものを選びなさい。(1つ選択)

```
1. public class SampleException extends Exception {}
```

```
1. public class SubSampleException extends SampleException {}
```

```
1. public class Sample {  
2.     public void test(int num) throws SampleException {  
3.         if (num < 10) {  
4.             throw new SubSampleException();  
5.         }  
6.     }  
7. }
```

- A. testメソッドでは、SampleExceptionクラスか、そのサブクラスであればスローできる
- B. SampleExceptionは非検査例外である
- C. SampleExceptionはプログラムで対処しきれないときに使う例外である
- D. testメソッドでは、SampleExceptionしかスローできないためコンパイルエラーが発生する

⇒ P597

78. 次のプログラムをコンパイルすると、何行目でコンパイルエラーになるか。(1つ選択)

```
1. public class Main {
2.     public static void main(String[] args) {
3.         sample();
4.         int a = b;
5.         int b = num;
6.     }
7.     private static void sample() {
8.         System.out.println(num);
9.     }
10.     static int num;
11. }
```

- A. 3行目
- B. 4行目
- C. 5行目
- D. 8行目

⇒ P598

79. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Main {
2.     public static void main(String[] args) {
3.         int[] array = {2, 3, 4};
4.         int[] array2 = array.clone();
5.         array2[0] = 5;
6.         for (int i : array) {
7.             System.out.println(i);
8.         }
9.     }
10. }
```

- A. 「2」「3」「4」と表示される
- B. 「5」「3」「4」と表示される
- C. コンパイルエラーが発生する
- D. 実行時に例外がスローされる

⇒ P598

80. 次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Main {
2.     public static void main(String[] args) {
3.         int num = 0;
4.         String[] array = {"A", "B", "C", "D"};
5.         for (String s : array) {
6.             switch(s) {
7.                 case "D":
8.                     case "A":
9.                         num -= 1;
10.                        break;
11.                     case "B":
12.                         num++;
13.                     case "C":
14.                         num += 2;
15.             }
16.         }
17.         System.out.println(num);
18.     }
19. }
```

- A. 3が表示される
- B. 4が表示される
- C. -1が表示される
- D. コンパイルエラーが発生する

⇒ P599