

- 1. 次の2つのコードのクラスファイルを生成するためのコマンドとして、正しいものを選びなさい。(1つ選択)

```
1. package b;  
2.  
3. public class B {  
4.  
5. }
```

```
1. package a;  
2. import b.B;  
3.  
4. public class A {  
5.     public static void main(String[] args) {  
6.         B b = new B();  
7.     }  
8. }
```

- A. java a/A.java
- B. java b/B.java
- C. javac b/B.java
java a/A.java
- D. javac a/A.java
java a/A

⇒ P449

- 2. 次の3つのステートメントを記述する順番として正しいものを選びなさい。(1つ選択)

- A. package import class
- B. package class import
- C. import class package
- D. class package import
- E. import package class

⇒ P449

- 3. 次のプログラムを確認し、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. package com.sample;  
2. public class Sample {  
3.     public void sample() {  
4.         // any code  
5.     }  
6. }
```

```
1. package com.sample.test;  
2. public class Test {  
3.     public void test(int num) {  
4.         // any code  
5.     }  
6. }
```

```
1. package com;  
2.   
3. public class Main {  
4.     public static void main(String[] args) {  
5.         new Sample().sample();  
6.         int num = Integer.parseInt(args[0]);  
7.         new Test().test(num);  
8.     }  
9. }
```

- A. import java.lang.Integer;
import com.*;
- B. import com.sample.*;
- C. import com.sample.Sample;
import com.sample.test.*;
- D. import java.lang.*;

⇒ P449

- 4. 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. import java.util.function.Supplier;
2.
3. public class Sample {
4.     void sample() {
5.         int i = 0;
6.         Supplier<Integer> foo = () -> i;
7.         i++;
8.         System.out.println(foo.get());
9.     }
10. }
```

- A. コンパイルできるが、何も表示されない
- B. 実行時に例外がスローされる
- C. 0が表示される
- D. 1が表示される
- E. コンパイルエラーが発生する

⇒ P450

- 5. 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.     static int num = 23;
3.     public static void main(String[] args) {
4.         int num = num;
5.         System.out.println(num);
6.     }
7. }
```

- A. 23が表示される
- B. 想定外の結果が表示される
- 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

⇒ P451

- 6. 次の2つのモジュールの設定ファイルに関する説明として、正しいものを選びなさい。(2つ選択)

```

1. module A {
2.     requires B;
3. }
```

```

1. module B {
2.     requires A;
3. }
```

- A. どちらのモジュールも、パッケージがエクスポートされない
- B. どちらのモジュールもjava.baseモジュールを必要としない
- C. モジュールの名前が短かすぎる
- D. コンパイルエラーが発生する

⇒ P452

- 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 Main {
2.     public static void main(String[] args) {
3.         int data = 1;
4.         switch(data) {
5.             default: System.out.print("C");
6.             case 0:  System.out.print("A");
7.                     break;
8.             case 10: System.out.print("B");
9.                     break;
10.        }
11.    }
12. }
```

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

⇒ P453

- 8. 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 str = "hoge456test";
4.         int index = 5;
5.         if(Character.isAlphabetic(str.charAt(index))) {
6.             index = 0;
7.         } else if (Character.isDigit(str.charAt(index))) {
8.             index = 10;
9.         } else {
10.            ++index;
11.        }
12.        if (str.endsWith("test"))
13.            str = str.substring(0, index);
14.        System.out.println(str);
15.    }
16. }
```

- A. 「hoge」と表示される
- B. 「hoge456」と表示される
- C. 「456test」と表示される
- D. 「hoge456test」と表示される
- E. 「hoge456tes」と表示される

⇒ P453

- 9. クラスやモジュールの依存関係を調べるために使うコマンドとして、正しいものを選びなさい。(2つ選択)

- A. jmod describe
- B. jdeps --list-deps
- C. java Hello.java
- D. jar --show-module-resolution
- E. java --show-module-resolution

⇒ P454

- 10. 次のプログラムのコンパイルを成功させるための方法として、正しいものを選びなさい。(1つ選択)

```

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

```

1. public abstract class B /* position 1 */ {
2.     /* position 2 */
3.     public void x() {}
4.     public abstract void z();
5. }
```

```

1. public class C extends B implements A {
2.     /* position 3 */
3. }
```

- A. 以下をposition 3に記述する
@Override
public void x() {}
- B. 以下をposition 3に記述する
public void z() {}
- C. 以下をposition 2に記述する
void x() {}
- D. 以下をposition 1に記述する
implements A

⇒ P455

- 11. 次のプログラムを確認し、変数valueのデータ型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.         for (var value : args) {
4.             System.out.println(value);
5.         }
6.     }
7. }
```

- A. String
- B. String[]
- C. Character
- D. var

⇒ P455

- 12. 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 = "abcd ef gh";
4.         int x = str.indexOf("ef");
5.         str.substring(x + 3);
6.         x = str.indexOf("ef");
7.         System.out.println(str + " " + x);
8.     }
9. }
```

- A. 「ef gh 5」と表示される
- B. 「ef gh 4」と表示される
- C. 「abcd ef gh 4」と表示される
- D. 「abcd ef gh 5」と表示される
- E. 実行時に例外がスローされる

⇒ P456

- 13. 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.     static String str;
3.     public static void main(String[] args) {
4.         switch(str) {
5.             case "10" : str += "10";
6.             default : str += "def";
7.             case "20" : str += "20";
8.         }
9.         System.out.println(str);
10.    }
11. }
```

- A. 「10 def 20」と表示される
- B. 「def 20」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

⇒ P457

- 14. 次のプログラムを確認し、変数xのデータ型として正しいものを選びなさい。(1つ選択)

```
1. import java.util.Arrays;
2. import java.util.List;
3.
4. public class Sample {
5.     public static void main(String[] args) {
6.         List<String> list = Arrays.asList(new String[]{"a", "b", "c"});
7.         list.forEach(x -> {System.out.println(x)});
8.     }
9. }
```

- A. List<Character>
- B. String
- C. char
- D. List<String>

⇒ P458

- 15. 次のプログラムをコンパイルし、実行するためのコマンドとして、正しいものを選びなさい。(1つ選択)

```

1. package ex15;
2.
3. public class Sample {
4.     private final String name;
5.     public Sample(String name) {
6.         this.name = name;
7.     }
8.     public String getName() {
9.         return name;
10.    }
11.    public String toString() {
12.        return name;
13.    }
14. }
```

```

1. package ex15;
2.
3. import java.util.logging.Logger;
4.
5. public class Main {
6.     static final Logger logger = Logger.getAnonymousLogger();
7.     public static void main(String[] args) {
8.         var d = new Sample("sample");
9.         logger.info(d.toString());
10.    }
11. }
```

- A. `javac ex15/Sample.java ex15/Main.java`
`java -cp build ex15.Main`
- B. `javac -d build ex15/Sample.java ex15/Main.java`
`java -cp build ex15.Main`
- C. `javac -d build ex15/Sample.java ex15/Main.java`
`java ex15.Main`
- D. `javac -d build ex15/Sample.java ex15/Main.java`
`java build.ex15.Main`

⇒ P458

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

```

1. import java.time.LocalDate;
2. import static java.time.DayOfWeek.*;
3.
4. public class Sample {
5.     public static void main(String[] args) {
6.         var today = LocalDate.now().with(TUESDAY).getDayOfWeek();
7.         switch (today) {
8.             case SUNDAY:
9.             case SATURDAY:
10.                System.out.println("weekend");
11.                break;
12.             case MONDAY:
13.                 FRIDAY: System.out.println("working");
14.             default:
15.                 System.out.println("other");
16.         }
17.     }
18. }
```

- A. 「working」と表示される
- B. 「working」「other」と表示される
- C. 「Tuesday」と表示される
- D. 「other」と表示される

⇒ P459

- 17. 修飾子に関する説明として正しいものを選びなさい。(2つ選択)

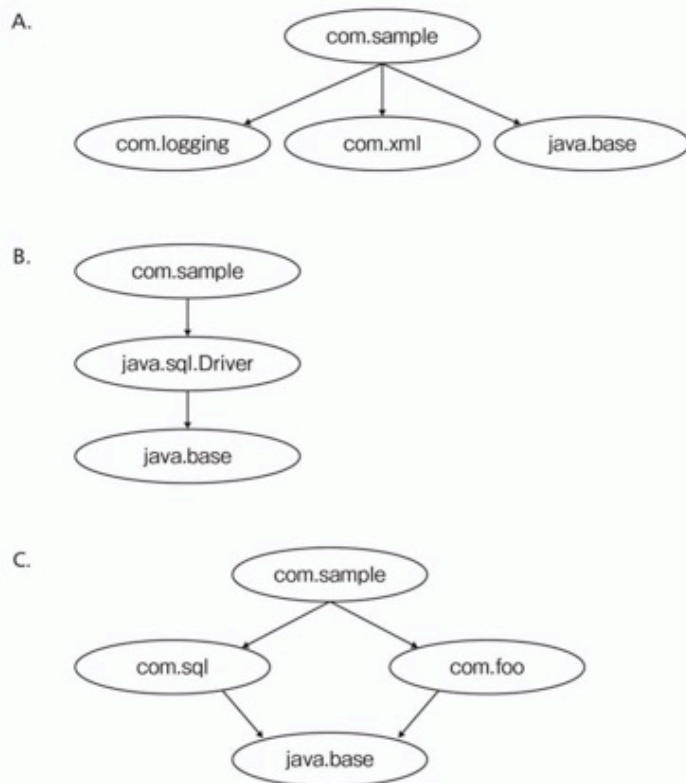
- A. ローカル変数は、finalを付けて宣言できる
- B. インタフェースは、protectedを付けて宣言できる
- C. インスタンス変数は、staticを付けて宣言できる
- D. 抽象メソッドは、privateを付けて宣言できる
- E. インナークラスは、publicを付けて宣言できる

⇒ P459

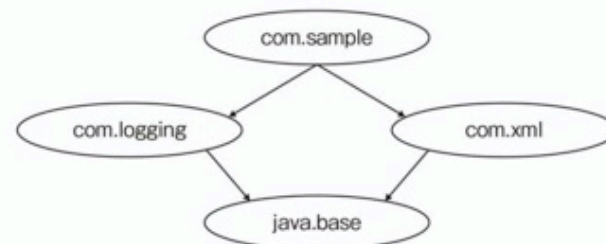
18. 次のコマンドを実行した結果、以下のような内容がコンソールに表示された。この内容を表すモジュールグラフとして、正しいものを選びなさい。(1つ選択)

```
> java --describe-module com.sample
```

```
com.sample
exports com.sql
exports com.foo
requires java.base mandated
requires com.logging transitive
requires com.xml transitive
uses java.sql.Driver
```



D.



⇒ P460

19. 次のプログラムを確認し、これらのクラスを利用するMainクラスをコンパイル、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 A {
2.     public void hello() {
3.         System.out.println("hello");
4.     }
5. }
```

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

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

- A. 「bye」と表示される
 B. 「null」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

⇒ P460

- 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 Sample {
2.     private char a;
3.     private int b = 1;
4.     String test(char a, int b) {
5.         return a + ", " + b;
6.     }
7.     public static void main(String[] args) {
8.         Sample app = new Sample();
9.         System.out.println(app.test('A'));
10.    }
11. }
```

- A. 「A, 0」と表示される
- B. 「A, 1」と表示される
- C. 「A, null」と表示される
- D. コンパイルエラーが発生する

⇒ P461

- 21. 多次元配列のインスタンス化と初期化のコードとして、正しいものを選びなさい。(2つ選択)

- A. `int[][] array = {{1,2,3},{4,5,6}};`
- B. `int[][][] array = {{1,2},{3,4},{5,6}};`
- C. `int[][] array = {0, 1};`
- D. `int[][] array = new int[][2];`
`array[0][0] = 1;`
`array[0][1] = 2;`
`array[1][0] = 3;`
`array[1][1] = 4;`
- E. `int[] array = {0, 1};`
`int[][][] array2 = new int[2][2][2];`
`array2[0][0] = array;`
`array2[0][1] = array;`
`array2[1][0] = array;`
`array2[1][1] = array;`

⇒ P461

- 22. 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 A {
2.     int num = 10;
3.     public double getValue() {
4.         return 0.0;
5.     }
6. }
```

```

1. public class B extends A {
2.     @Override
3.     public double getValue() {
4.         // ...
5.         return 1.0;
6.     }
7. }
```

```

1. public class C extends B {
2.     @Override
3.     public double getValue() {
4.         System.out.println(super.num);
5.         return super.getValue();
6.     }
7. }
```

- A. Bクラスのサブクラスでは、getValueメソッドでAクラスのフィールドにアクセスすることができる
- B. Bクラスのサブクラスでは、Bクラスのメソッドを呼び出すためにsuper.getValue()を使えるが、Aクラスのフィールドにはアクセスできない
- C. Bクラスのサブクラスでは、Bクラスのメソッドを呼び出すためにnew.getValue()を使う
- D. Bクラスのサブクラスでは、Bクラスのメソッドを呼び出すためにpublic.getValue()を使う

⇒ P462

□ 23. 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.Collection;
2.
3. public class A {
4.     public void sample(Collection arg) {
5.         System.out.println("A");
6.     }
7. }
```

```
1. import java.util.Collection;
2. import java.util.List;
3.
4. public class B extends A {
5.     public void sample(Collection arg) {
6.         System.out.println("B");
7.     }
8.     public void sample(List arg) {
9.         System.out.println("C");
10.    }
11. }
```

```
1. import java.util.ArrayList;
2. import java.util.List;
3.
4. public class Main {
5.     public static void main(String[] args) {
6.
7.         A a1 = new A();
8.         A a2 = new B();
9.         B b1 = new B();
10.
11.         List<String> list = new ArrayList<>();
12.
13.         // do something
14.
15.     }
16. }
```

- A. 「a1.sample(list)」を実行すると、Aが表示される
- B. 「a2.sample(list)」を実行すると、Aが表示される
- C. 「a2.sample(list)」を実行すると、Bが表示される
- D. 「b1.sample(list)」を実行すると、Bが表示される
- E. 「b1.sample(list)」を実行すると、Cが表示される

⇒ P463

□ 24. Dクラスの説明として正しいものを選びなさい。(1つ選択)

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

```
1. public interface B extends A {
2.     default void sample() {
3.         System.out.println("B");
4.     }
5. }
```

```
1. public interface C extends A {
2.     @Override
3.     default void sample() {
4.         System.out.println("C");
5.     }
6. }
```

```
1. public class D implements B, C {
2.     @Override
3.     public void sample() {
4.         super.sample();
5.     }
6. }
```

- A. Bインタフェースのsampleメソッドを実装している
- B. Cインタフェースのsampleメソッドを実装している
- C. Dクラスがsampleメソッドを使わないのであれば、オーバーライドをする必要はない
- D. コンパイルエラーが発生する

⇒ P464

- 25. 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 abstract class A {
2.     abstract void sample();
3. }
```

```
1. public class B extends A {
2.     @Override
3.     void sample(int i) {
4.         System.out.println(10);
5.     }
6.     public static void main(String[] args) {
7.         A a = new B();
8.         a.sample();
9.     }
10. }
```

- A. 0が表示される
- B. 10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

⇒ P465

- 26. 次のプログラムを確認し、コンパイルエラーとなるコードを選びなさい。(1つ選択)

```
public abstract class A {}
```

```
public interface B {}
```

```
public class C extends A implements B {}
```

```
public class D extends C {}
```

- A. `List<A> listA = new ArrayList<>();`
`listA.add(new D());`

- B. `List listB = new ArrayList<>();`
`listB.add(new C());`
- C. `List listC = new ArrayList<>();`
`listC.add(new D());`
- D. `List<D> listD = new ArrayList<>();`
`listD.add(new C());`
- E. `List<A> listE = new ArrayList<>();`
`listE.add(new C());`

⇒ P465

- 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 Main {
2.     public static void test(Integer a, Integer b) {
3.         System.out.println("A");
4.     }
5.     public static void test(double a, double b) {
6.         System.out.println("B");
7.     }
8.     public static void test(float a, float b) {
9.         System.out.println("C");
10.    }
11.    public static void test(int a, int b) {
12.        System.out.println("D");
13.    }
14.    public static void main(String[] args) {
15.        test(10, 20);
16.        test(10.0, 20.0);
17.    }
18. }
```

- A. 「D」「C」と表示される
- B. 「D」「B」と表示される
- C. 「A」「B」と表示される
- D. 「A」「C」と表示される

⇒ P466

□ 28. 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. public interface A {
2.     public A build(String str);
3. }
```

```
1. public class B implements A {
2.     public B(String str) {
3.         // ...
4.     }
5.     @Override
6.     public B build(String str) {
7.         return new B(str);
8.     }
9. }
```

- A. Aはfinalにできない
- B. Bはabstractにできない
- C. Bはfinalにできない
- D. AはBのサブタイプである
- E. Aはabstractにできない
- F. BはAのサブタイプである

⇒ P467

□ 29. JDKをモジュール化する理由の説明として、正しいものを選びなさい。(2つ選択)

- A. アプリケーションモジュールとJDKのモジュールのリンクが容易に構築できる
- B. Java言語を理解することが簡単になる
- C. 実装の詳細を簡単に公開しやすくなる
- D. セキュリティとメンテナンス性が向上する
- E. アプリケーションの堅牢性が向上する

⇒ P468

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

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

⇒ P469

- 31. 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.ArrayList;
2. import java.util.Arrays;
3.
4. public class Main {
5.     public static void main(String[] args) {
6.         var alphabet = new ArrayList<>{
7.             Arrays.asList(
8.                 new String[] {"A", "B", "C", "D", "E"}
9.             );
10.        alphabet.sort((var a, var b) -> -a.compareTo(b));
11.        alphabet.forEach(System.out::println);
12.    }
13. }
```

- A. 何も表示されない
- B. 「A」「B」「C」「D」「E」と表示される
- C. 「E」「D」「C」「B」「A」と表示される
- D. 「A」「B」と表示される

⇒ P469

- 32. 抽象クラスの定義として、正しいものを選びなさい。(1つ選択)

- A.

```
abstract class Item {
    public abstract int calcPrice(Item item);
    public void print(Item item) { /* do something */ }
}
```
- B.

```
abstract class Item {
    public int calcPrice(Item item);
    public void print(Item item);
}
```
- C.

```
abstract class Item {
    public int calcPrice(Item item);
    public final void print(Item item) { /* do something */ }
}
```

D.

```
abstract class Item {
    public abstract int calcPrice(Item item) {
        /* do something */
    }
    public abstract void print(Item item) {
        /* do something */
    }
}
```

⇒ P469

- 33. 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. import java.util.HashMap;
2. import java.util.List;
3. import java.util.Map;
4.
5. public class Main {
6.     static Map<String, String> map = new HashMap<>();
7.     static List<String> keys = List.of("A", "B", "C");
8.     static String[] values = {"1", "2", "3"};
9.     static {
10.        for (var i = 0; i < keys.size(); i++) {
11.            map.put(keys.get(i), values[i]);
12.        }
13.    }
14.     public static void main(String[] args) {
15.        keys.clear();
16.        values = new String[0];
17.        System.out.println(map.size() + ", " + keys.size() + ", " + values.length);
18.    }
19. }
```

- A. 「3,3,0」と表示される
- B. 「3,0,0」と表示される
- C. 「3,3,3」と表示される
- D. コンパイルエラーが発生する
- E. 実行時に例外が発生する

⇒ P470

- 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 Sample {
2.     public static void main(String[] args) {
3.         char c = 'b';
4.         int i = 0;
5.         switch (c) {
6.             case 'a':
7.                 i++;
8.                 break;
9.             case 'b':
10.                i++;
11.            case 'c' | 'd':
12.                i++;
13.            case 'e':
14.                i++;
15.                break;
16.            case 'f':
17.                i++;
18.                break;
19.            default:
20.                System.out.println(c);
21.        }
22.        System.out.println(i);
23.    }
24. }
```

- A. 3が表示される
- B. b1が表示される
- C. 1が表示される
- D. b2が表示される

⇒ P470

- 35. 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 A {
2.     private void print() {
3.         System.out.println("A");
4.     }
5.     public void a() {
6.         print();
7.     }
8. }
```

```
1. public class B extends A {
2.     private void print() {
3.         System.out.println("B");
4.     }
5.     public void b() {
6.         print();
7.     }
8.     public static void main(String... args) {
9.         B b = new B();
10.        b.a();
11.        b.b();
12.    }
13. }
```

- A. 「A」「B」と表示される
- B. 「B」「B」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

⇒ P471

- 36. 次の繰り返し処理のコードと同じ結果となるfor文を選びなさい。(1つ選択)

```
int x = 0;
while(x < 10) {
    System.out.print(x++);
}
```

- A. for (int a = 0; a < 10;) {
 System.out.print(a);
 a++;
}
- B. int b = 0;
 for (; b < 10;) {
 System.out.print(++b);
 }
- C. int c = 0;
 for (; c++) {
 System.out.print(c);
 }
- D. for (int d = 0; d < 10; d++) {
 System.out.print(d++);
}

⇒ P472

- 37. 次のうち、メソッドの定義として正しいものを選びなさい。(2つ選択)

- A. public void methodA(int x) {
 return ++x;
}
- B. public String methodB() {
 System.out.println("B");
}

- C. public boolean methodC(int a) {
 return a < 0;
}
- D. public char methodD(String str) {
 return str;
}
- E. public void methodE() {
 return;
}

⇒ P473

- 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. import java.util.List;
2.
3. public class Main {
4.     public static void main(String[] args) {
5.         List<Character> list = List.of();
6.         list.add('a');
7.         list.add('b');
8.         list.add('c');
9.         System.out.println(list);
10.    }
11. }
```

- A. [a, b, c]
- B. [c, b, a]
- 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

⇒ P473

- 39. 次のプログラムを以下に示した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 A "A B" A B
```

```
1. public class A {
2.     public static void main(String[] args) {
3.         for (String str : args) {
4.             System.out.print(str);
5.         }
6.     }
7. }
```

- A. 「AA BAB」と表示される
- B. 「A BAB」と表示される
- C. 「A BA B」と表示される
- D. 「A B A B」と表示される
- E. 実行時に例外がスローされる

⇒ P474

- 40. 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"};
4.         for (int i = 0; i < array.length; i++) {
5.             System.out.print(array[i] + " ");
6.             if (array[i].equals("C")) {
7.                 continue;
8.             }
9.             System.out.println("end");
10.            break;
11.        }
12.    }
13. }
```

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

⇒ P474

- 41. 次のプログラムを確認し、Dインタフェースでコンパイルエラーが発生する理由として考えられるものを選びなさい。(1つ選択)

```
1. public interface A {
2.     public Iterable a();
3. }
```

```
1. import java.util.Collection;
2.
3. public interface B extends A {
4.     public Collection a();
5. }
```

```
1. import java.nio.file.Path;
2.
3. public interface C extends A {
4.     public Path a();
5. }
```

```
1. public interface D extends B, C {
2.
3. }
```

- A. Dインタフェースが2つ以上継承しているため
- B. BインタフェースがAインタフェースとは異なる戻り値型の同名メソッドを定義しているため
- C. BとCインタフェースから継承したメソッドの戻り値型が異なるため
- D. Dインタフェースで継承したメソッドを再定義しなかったため

⇒ P475

42. 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. package ex42.a;
2.
3. public class A {
4.     public String x = "A";
5.     protected A() { }
6. }
```

```
1. package ex42.b;
2.
3. import ex42.a.A;
4.
5. public class B extends A {
6.     String x = "B";
7.     public B() {
8.         super();
9.     }
10. }
```

```
1. package ex42;
2.
3. import ex42.a.A;
4. import ex42.b.B;
5.
6. public class Main {
7.     public static void main(String[] args) {
8.         A a = new B();
9.         System.out.println(a.x);
10.     }
11. }
```

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

⇒ P476

43. 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 A {
2.     private boolean isValue(int val) {
3.         return true;
4.     }
5. }
```

```
1. public class B extends A {
2.     public int test(int num) {
3.         if (isValue(num)) {
4.             return num;
5.         }
6.         return 0;
7.     }
8.     public static void main(String[] args) {
9.         B b = new B();
10.        System.out.println(b.test(10));
11.    }
12. }
```

- A. 0が表示される
- B. 10が表示される
- C. 実行時に例外がスローされる
- D. コンパイルエラーが発生する

⇒ P476

- 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. public class Sample {
2.     String name;
3.     int num;
4.     public Sample(String name, int num) {
5.         this.name = name;
6.         this.num = num;
7.     }
8. }
```

```
1. public class SubSample extends Sample {
2.     int price;
3.     public SubSample(int price) {
4.         this.price = price;
5.     }
6.     public SubSample(String name, int num, int price) {
7.         super(name, num);
8.         this.price;
9.     }
10. }
```

```
1. public class Main {
2.     public static void main(String[] args) {
3.         SubSample s1 = new SubSample(100);
4.         SubSample s2 = new SubSample("sample", 200, 100);
5.         System.out.println(s1.name + ", " + s1.num + ", " + s1.price);
6.         System.out.println(s2.name + ", " + s2.num + ", " + s2.price);
7.     }
8. }
```

- A. sample 200 100
sample 200 100
- B. null 0 100
sample 200 100
- C. SubSampleクラスの3行目だけでコンパイルエラーになる
- D. SubSampleクラスの8行目だけでコンパイルエラーになる
- E. SubSampleクラスの3行目と8行目の両方でコンパイルエラーになる

⇒ P477

- 45. testメソッドのオーバーライドとして正しくないものを選びなさい。(1つ選択)

```
1. public interface Sample {
2.     void test();
3. }
```

- A.

```
public class SampleTest implements Sample {
    @Override
    public void test() {
        System.out.println("test");
    }
}
```
- B.

```
public class SampleTest2 {
    @Override
    public void test() {
        System.out.println("test");
    }
}
```
- C.

```
public abstract class SampleTest3 implements Sample {
    @Override
    public abstract void test();
}
```
- D.

```
public abstract class SampleTest4 implements Sample {
    @Override
    public void test() {
        System.out.println("test");
    }
}
```

⇒ P478

- 46. 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。なお、現在の日付は2019年8月14日とする。(1つ選択)

```
1. import java.time.LocalDate;
2.
3. public class Diary {
4.     private LocalDate now = LocalDate.now();
5.     public LocalDate getNow() {
6.         return this.now;
7.     }
8. }
```

```
1. public class Main {
2.     public static void main(String[] args) {
3.         Diary d = new Diary();
4.         System.out.println(d.getNow());
5.     }
6. }
```

- A. 「2019-08-14」と表示される
- B. 「08-14-2019」と表示される
- C. 「14-08-2019」と表示される
- D. 「08-14」と表示される
- E. 「2019-08」と表示される

⇒ P479

- 47. 次の数式を表したコードとして、正しいものを選びなさい。(1つ選択)

$$a = b \frac{c(1+c)^n}{(1+c)^n - 1}$$

- A. `a = b * (c * Math.pow(1 + c, n) / (Math.pow(1 + c, n) - 1));`
- B. `a = b * c * Math.pow(1 + c, n) / Math.pow(1 + c, n) - 1;`
- C. `a = b * c * (Math.pow(1 + c, n) / (Math.pow(1 + c, n) - 1));`
- D. `a = b * Math.pow(1 + c, n) - 1 / (c * Math.pow(1 + c, n));`

⇒ P479

- 48. 次のプログラムを以下に示した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 Main a b c
```

```
1. public class Main {
2.     public static void main(String[] args) {
3.         int i = 1;
4.         for (String s : args) {
5.             System.out.println((i++) + " " + s);
6.         }
7.     }
8. }
```

- A. 1) a
2) b
3) c
- B. 2) a
3) b
4) c
- C. 1) Main
2) a
3) b
4) c
- D. 2) Main
3) a
4) b
5) c

⇒ P479

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;
3.     int b;
4.     int c;
5.     int d;
6.     void setA(int a) {
7.         a = a;
8.     }
9.     void setB() {
10.        this.b = b;
11.    }
12.    int setC() {
13.        return c;
14.    }
15.    int setD(int num) {
16.        d = num;
17.        return d;
18.    }
19.    void setAll(int x) {
20.        a = b = this.c = setD(x);
21.    }
22.    @Override
23.    public String toString() {
24.        return "Sample [a=" + a + ", b=" + b + ", c=" + c + ", d=" + d + "];"
25.    }
26.    public static void main(String[] args) {
27.        Sample s = new Sample();
28.        s.setAll(10);
29.        System.out.println(s);
30.    }
31. }
```

- A. Sample [a=10, b=10, c=10, d=10]
- B. Sample [a=0, b=10, c=10, d=10]
- C. Sample [a=0, b=0, c=0, d=10]
- 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

⇒ P480

50. 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.         do {
5.             num++;
6.             if (num == 1) {
7.                 continue;
8.             }
9.             System.out.println(num);
10.        } while (num < 1);
11.    }
12. }
```

- A. 「0」「1」と表示される
- B. 「0」と表示される
- C. 「1」と表示される
- D. 何も表示されない

⇒ P481

51. 例外処理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. finallyブロックは、tryブロックまたはcatchブロックのすぐ後ろに配置しなければならない
- B. try-with-resourcesのリソースを閉じる前に、finallyブロックが実行される
- C. tryブロックは複数のcatchブロックを持つことができる
- D. catchブロックは、一般的な型から特定の型への順に並べなければならない
- E. tryブロックは、catchブロックとfinallyブロックを持たなければならない

⇒ P481

52. 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.Arrays;
2.
3. public class Main {
4.     public static void main(String[] args) {
5.         int[] array1 = {1, 2, 3, 4, 5};
6.         int[] array2 = {1, 2, 4, 5, 3};
7.         int result1 = Arrays.mismatch(array1, array2);
8.         int result2 = Arrays.compare(array1, array2);
9.         System.out.println(result1 + ":" + result2);
10.    }
11. }
```

- A. 「-1:2」と表示される
 B. 「2:3」と表示される
 C. 「2:-1」と表示される
 D. 「3:0」と表示される

⇒ P482

53. 次のプログラムを実行したときに「2.5」と表示されるようにしたい。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. public class Main {
2.     public static void main(String[] args) {
3.         byte x = 5, y = 2;
4.         
5.         System.out.println(a);
6.     }
7. }
```

- A. float a = Math.round((float) x / y * 100) / (float) 100;
 B. float a = (float) (Math.round((float) x / y * 100)/100);
 C. float a = Math.round((int) (x / y), 2);
 D. float a = Math.round((float) x / y, 2);

⇒ P482

54. 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 interface Sample {
2.     void test();
3. }
```

```
1. class SampleImpl implements Sample {
2.     @Override
3.     public void test() {
4.         // do something
5.     }
6. }
```

```
1. class SubSampleImpl extends SampleImpl {
2.     void test(int x) {
3.         // do something...
4.     }
5. }
```

- A. Sampleインタフェースでコンパイルエラーが発生する
 B. SampleImplクラスでコンパイルエラーが発生する
 C. SubSampleImplクラスでコンパイルエラーが発生する
 D. 正常にコンパイル、実行できる

⇒ P483

55. 次のステートメントのうち、正しいものを選びなさい。(4つ選択)

- A. int[][] a = {{1,1}, {2,2}};
 B. short b = (short)'A';
 C. byte c1 = 10;
 char c2 = c1;
 D. boolean d = (10 == 10);
 E. float e = 1.99;
 F. int f = 12_34;
 G. String g = 'a';

⇒ P484

56. 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[][] array = {
4.             {"A", "B"},
5.             {"C"},
6.             {"D", "E", "F", "G"}
7.         };
8.         for (int i = 0; i < array.length; i++) {
9.             int j = 0;
10.            for (; j < array[i].length; j++) {
11.                System.out.println "[" + i + ", " + j + "] = " + array[i][j]);
12.            }
13.        }
14.    }
15. }
```

- A. [0, 0] = A
[0, 1] = B
[1, 0] = C
[1, 1] = D
[2, 0] = E
[2, 1] = F
[2, 2] = G
- B. [0, 0] = A
[0, 1] = B
[1, 0] = C
[2, 0] = D
[2, 1] = E
[2, 2] = F
[2, 3] = G

- C. [0, 0] = A
[0, 1] = B
[1, 0] = C
[1, 1] = D
[2, 0] = E
[2, 1] = F
[3, 0] = G
- D. [0, 0] = A
[0, 1] = B
[0, 2] = C
[1, 0] = D
[1, 1] = E
[1, 2] = F
[1, 3] = G

⇒ P485

57. 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.             throw new Exception();
5.         } catch (Exception e) {
6.             throw new RuntimeException();
7.         } catch (RuntimeException e) {
8.             System.out.println("A");
9.         } finally {
10.            System.out.println("B");
11.        }
12.    }
13. }
```

- A. Aが表示される
B. B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

⇒ P486

- 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 Main {
2.     public static void main(String[] args) {
3.         int num = 9;
4.         if (num++ < 10) {
5.             System.out.println(num);
6.         } else {
7.             System.out.println("B");
8.         }
9.     }
10. }
```

- A. 10が表示される
- B. 8が表示される
- C. 9が表示される
- D. コンパイルエラーが発生する

⇒ P486

- 59. 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 = 2;
5.         System.out.println(a + b + " = (a + b) = " + a + b);
6.     }
7. }
```

- A. 「6 = (a + b) = 42」と表示される
- B. 「6 = (a + b) = 6」と表示される
- C. 「42 = (a + b) = 42」と表示される
- D. 「42 = (a + b) = 6」と表示される

⇒ P486

- 60. 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 s1 = 10;
4.         Integer s2 = 20;
5.         Long s3 = (long) s1 + s2;
6.         String s4 = (String) (s3 + s2);
7.         System.out.println(s4);
8.     }
9. }
```

- A. 30が表示される
- B. 5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. 5行目でClassCastExceptionがスローされる
- E. 6行目でClassCastExceptionがスローされる

⇒ P487

- 61. 次のプログラムをコンパイル、実行した結果として、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 s1 = new String("Java");
4.         String s2 = "Java";
5.         String s3 = s2.intern();
6.         System.out.print((s1 == s2) + ", ");
7.         System.out.print((s2 == s3) + ", ");
8.         System.out.println(s1 == s3);
9.     }
10. }
```

- A. 「false, true, false」と表示される
- B. 「true, true, true」と表示される
- C. 「true, false, false」と表示される
- D. 「true, true, false」と表示される
- E. 「false, false, false」と表示される

⇒ P488

62. 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.         char[][] arrays = {{'a', 'd'}, {'b', 'e'}, {'c', 'f'}};
4.         for (char[] array : arrays) {
5.             for (char ch : array) {
6.                 System.out.print(ch);
7.             }
8.             System.out.print(" ");
9.         }
10.    }
11. }
```

- A. 「ad be cf」と表示される
- B. 「abc def」と表示される
- C. 「a d b e c f」と表示される
- D. 「adb ecf」と表示される

⇒ P489

63. 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 x;
4.         int y = 3;
5.         if (y > 2) {
6.             x = ++y;
7.             y = x + 5;
8.         } else {
9.             y++;
10.        }
11.        System.out.println(x + ", " + y);
12.    }
13. }
```

- A. 「4, 9」と表示される
- B. 「3, 8」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

⇒ P490

64. 次のプログラムを以下に示したコマンドで実行した結果として、正しいものを選びなさい。(1つ選択)

```
> java Main 1 2 3
```

```
1. public class Main {
2.     public static void main(int[] args) {
3.         System.out.println("A");
4.     }
5.     public static void main(Object[] args) {
6.         System.out.println("B");
7.     }
8.     public static void main(String[] args) {
9.         System.out.println("C");
10.    }
11. }
```

- 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. 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

⇒ P490

65. 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 A {  
2.     protected A() {  
3.         this(2);  
4.         System.out.print("1");  
5.     }  
6.     protected A(int a) {  
7.         System.out.print(a);  
8.     }  
9. }
```

```
1. public class B extends A {  
2.     B() {  
3.         this(4);  
4.         System.out.print("3");  
5.     }  
6.     B(int b) {  
7.         System.out.print(b);  
8.     }  
9. }
```

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

- A. 2143が表示される
- B. 1234が表示される
- C. 4321が表示される
- D. 3421が表示される

⇒ P490

66. 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 Sample {  
2.     public static void print(Integer num) {  
3.         System.out.println(num);  
4.     }  
5. }
```

```
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.         list.forEach( );  
7.     }  
8. }
```

- A. new Sample()::print
- B. Sample.print
- C. Sample:print
- D. Sample::print

⇒ P491

67. Javaの開発環境のセットアップに関する説明として、正しいものを選びなさい。(1つ選択)

- A. JDKをインストールするときには、特定のオペレーティングシステム用のJava開発環境をセットアップする
- B. Javaの開発環境はデフォルトですべてのOS用に設定されている
- C. Javaの開発環境をセットアップするにはJREが必要である
- D. JDKよりも前にIDE（統合開発環境）をインストールすれば、Javaの開発環境はセットアップされる

⇒ P492

68. 次のクラスのコンストラクタのうち、コンパイル可能で、かつarrayフィールドに参照をセットすることができるものを選びなさい。(2つ選択)

```
1. public class Sample {
2.     private String[] array;
3. }
```

- A. `public Sample(String[] array) {
 this.array = array;
}`
- B. `public Sample(String... array) {
 array = array;
}`
- C. `public Sample(String str) {
 this.array = str;
}`
- D. `public Sample(String[] array) {
 array = array;
}`
- E. `public Sample(String... array) {
 this.array = array;
}`
- F. `public Sample(Object[] array) {
 this.array = array;
}`

⇒ P492

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. import java.util.List;
2. import java.util.ArrayList;
3.
4. public class Main {
5.     public static void main(String[] args) {
6.         List<String> list = new ArrayList<>();
7.         try {
8.             while(true) {
9.                 list.add("hello");
10.            }
11.        } catch (RuntimeException e) {
12.            System.out.println("A");
13.        } catch (Exception e) {
14.            System.out.println("B");
15.        }
16.        System.out.println("C");
17.    }
18. }
```

- 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. 実行時に例外がスローされる
- E. throws宣言がないためコンパイルエラーが発生する

⇒ P493

70. com.exampleモジュール内にあるmainメソッドを持つcom.sample.Mainクラスをコマンドラインから実行するためのコマンドとして、正しいものを選びなさい。(1つ選択)

- A. `java -classpath com.example.jar -m com.example/com.sample.Main`
- B. `java -classpath com.example.jar com.sample.Main`
- C. `java --module-path mods -m com.example/com.sample.Main`
- D. `java --module-path mods com.example/com.sample.Main`

⇒ P493

71. 次のプログラムを実行し、「abcd」と表示したい。「// 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) {
        String[][] array = new String[2][2];
        array[0][0] = "a";
        array[0][1] = "b";
        array[1][0] = "c";
        array[1][1] = "d";
        // insert code here
    }
}
```

- A.

```
for (int i = 1; i < 2; i++) {
    for (int j = 1; j < 2; j++) {
        System.out.print(array[i][j]);
    }
}
```
- B.

```
for (int i = 0; i < 2; ++i) {
    for (int j = 0; j < i; ++j) {
        System.out.print(array[i][j]);
    }
}
```
- C.

```
for (String a : array) {
    for (String b : array) {
        System.out.println(b);
    }
}
```
- D.

```
for (int i = 0; i < 2;) {
    for (int j = 0; j < 2;) {
        System.out.print(array[i][j]);
        j++;
    }
    i++;
}
```

⇒ P494

72. 次のプログラムを実行して「A」「X」「C」「D」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。(1つ選択)

```
1. public class Sample {
2.     public static void main(String[] args) {
3.         StringBuilder sb = new StringBuilder("ABCD");
4.         
5.         System.out.println(sb);
6.     }
7. }
```

- A. `sb.replace(sb.indexOf("B"), sb.indexOf("B"), "X");`
- B. `sb.replace(sb.indexOf("B"), sb.indexOf("C"), "X");`
- C. `sb.replace(sb.indexOf("A"), sb.indexOf("B"), "X");`
- D. `sb.replace(sb.indexOf("A"), sb.indexOf("C"), "X");`

⇒ P495

73. 次のプログラムを実行し、「135」と表示したい。4行目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 Main {
2.     public static void main(String[] args) {
3.         int[] array = { 1, 2, 3, 4, 5 };
4.         for (  ) {
5.             System.out.print(array[i]);
6.         }
7.     }
8. }
```

- A. `int i = 0; i <= 4; i++`
- B. `int i = 0; i < 5; i += 2`
- C. `int i = 1; i <= 5; i += 1`
- D. `int i = 1; i < 5; i +=2`

⇒ P496

74. 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[] str = new String[2];
4.         int i = 0;
5.         for (String s : str) {
6.             str[i].concat("e" + i);
7.             i++;
8.         }
9.         for (i = 0; i < str.length; i++) {
10.            System.out.println(str[i]);
11.        }
12.    }
13. }

```

- A. e0
e1
- B. null e0
null e1
- C. null
null
- D. 実行時に例外がスローされる

⇒ P496

75. モジュールaはモジュールbに依存し、ほかのすべてのモジュールからcom.sample/パッケージにアクセスできるようにしたい。この要件に合致するmodule-info.javaを選びなさい。(1つ選択)

- A.

```
module a {
    requires com.sample;
    exports b;
}
```
- B.

```
module a {
    requires b;
    exports com.sample;
}
```

- C.

```
module a {
    import com.sample;
    export b;
}
```
- D.

```
module a {
    import b;
    export com.sample;
}
```

⇒ P497

76. 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 void test(Object[] array) {
3.         System.out.println("A");
4.     }
5.     public void test(String[] array) {
6.         System.out.println("B");
7.     }
8.     public void test(Object obj) {
9.         System.out.println("C");
10.    }
11.    public static void main(String[] args) {
12.        new Main().test(args);
13.    }
14. }

```

- 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. 実行時に例外がスローされる
- E. コンパイルエラーが発生する

⇒ P497

77. 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() {
3.         System.out.println("Sample#test()");
4.     }
5. }
```

```
1. public class SubSample extends Sample {
2.     @Override
3.     public void test() {
4.         System.out.println("SubSample#test()");
5.     }
6. }
```

```
1. public class Main {
2.     public static void main(String[] args) {
3.         SubSample sub = new SubSample();
4.         Sample sample = new SubSample();
5.         sub = sample;
6.         sub.test();
7.         sample.test();
8.     }
9. }
```

- A. Sample#test()
Sample#test()
- B. SubSample#test()
SubSample#test()
- C. SubSample#test()
Sample#test()
- D. 実行時に例外がスローされる
- E. コンパイルエラーが発生する

⇒ P498

78. 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。(2つ選択)

```
1. package ex78.a;
2.
3. public abstract class A {
4.     protected abstract void test();
5. }
```

```
1. package ex78.b;
2.
3. import ex78.a.A;
4.
5. public class B extends A {
6.     
7. }
```

- A. public void test() {}
- B. private void test() {}
- C. void test() {}
- D. protected void test() {}

⇒ P498

79. モジュール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. コマンドラインからモジュールのエクスポートを設定することはできない
- B. Javaの基本APIはjava.baseモジュールにある
- C. JDKがモジュール化されたため、APIは非推奨になった
- D. モジュール式JDKで実行するには、アプリケーションをモジュールとして構造化する必要がある

⇒ P498

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 Test {  
2.     public static void main(String[] args) {  
3.         String[][] array = new String[2][];  
4.         array[0] = new String[2];  
5.         array[1] = new String[5];  
6.         int i = 97;  
7.         for (int a = 0; a < array.length; a++) {  
8.             for (int b = 0; b < array.length; b++) {  
9.                 array[a][b] = "" + i;  
10.                i++;  
11.            }  
12.        }  
13.        for (String[] tmp : array) {  
14.            for (String s : tmp) {  
15.                System.out.print(s + " ");  
16.            }  
17.            System.out.println();  
18.        }  
19.    }  
20. }
```

- A. 97 98
99 100 null null null
- B. 97 98
99 100 101 102 103
- C. コンパイルエラーが発生する
- D. 実行時にNullPointerExceptionがスローされる
- E. 実行時にArrayIndexOutOfBoundsExceptionがスローされる

⇒ P499