

第6回 ループ文

(1) 第8章 01.

次のコードをコンパイル、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.         for(int i = 0; i < 5; i++) {  
4.             for(i = 5; i < 10; i++) {  
5.                 System.out.print(i);  
6.             }  
7.         }  
8.     }  
9. }
```

- A. 何も表示されない
- B. 「56789」と1回表示される
- C. 「56789」と5回表示される
- D. コンパイルエラーになる
- E. 実行時に例外が発生する

[B]

(2) 第8章 12.

次のコードをコンパイル、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.         for(int i = 0;      ; i++) {  
4.             int j = 0;  
5.             while(j <= 3)  
6.                 System.out.print(j++);  
7.         }  
8.     }  
9. }
```

- A. 「123」と1回表示される
- B. 「0123」と1回表示される
- C. 「1234」と1回表示される
- D. 「1234」が無限に表示される
- E. 「0123」が無限に表示される
- F. コンパイルエラーとなる

[E]

(3) 第8章 26.

次のコードをコンパイル、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 i = 4;  
4.         while(i >= 0)  
5.             System.out.print((i--) + " ");  
6.     }  
7. }
```

- A. 「4321」と表示される
- B. 「43210」と表示される
- C. 「321」と表示される
- D. 「3210」と表示される
- E. 「4」が無限に表示される
- F. 「3」が無限に表示される

[B]

(4) 第8章 33.

次のコードをコンパイル、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 i = 0;  
4.         do {  
5.             ++i;  
6.             System.out.println("hoge");  
7.         } while(i < 3);  
8.     }  
9. }
```

- A. 「hoge」が1回表示される
- B. 「hoge」が3回表示される
- C. 「hoge」が4回表示される
- D. コンパイルエラーになる
- E. 実行時に例外が発生する

[B]

第 7 回 クラス

(1) 第 6 章 1.

クラスの名前として適切なものを選びなさい。(2 つ選択)

- A. **EmployeeList**
- B. **\$EmployeeList**
- C. **1EmployeeList**
- D. Employee-List
- E. **%EmployeeList**

[] []

(2) 第 6 章 3.

クラス定義の記述として正しいものを選びなさい。(1 つ選択)

- A. class ClassA(int a){
 // クラス内の定義
 }
B. ClassA {
 // クラス内の定義
 }
C. class ClassA(){
 // クラス内の定義
 }
D. **class ClassA {**
 // クラス内の定義
 }

[]

(3) 第8章 17.

次のコードをコンパイル、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 Bridge {  
2.        String name;  
3.  
4.        public static void main(String[] args) {  
5.            Bridge b = new Bridge();  
6.  
7.            if(b.name == "")  
8.                b.name = "Brooklyn";  
9.  
10.           System.out.println(b.name);  
11.       }  
12.   }
```

- A. 何も表示されない
- B. 「null」と表示される
- C. 「Brooklin」と表示される
- D. コンパイルエラーになる
- E. 実行時に例外が発生する

[]

(4) 第8章 28.

次のコードをコンパイル、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 Airline {  
2.     String flightNumber;  
3.  
4.     public void printFlightNumber() {  
5.         System.out.println(flightNumber);  
6.     }  
7.  
8.     public static void main(String[] args) {  
9.         Airline a1 = new Airline();  
10.        Airline a2 = a1;  
11.        a1.flightNumber = "DAL027";  
12.        a2.flightNumber = "DAL305";  
13.  
14.        a1.printFlightNumber();  
15.        a2.printFlightNumber();  
16.    }  
17. }
```

- A. 「DAL027」「DAL305」と表示される
- B. 「DAL027」「DAL027」と表示される
- C. 「DAL305」「DAL305」と表示される
- D. コンパイルエラーになる
- E. 実行時に例外が発生する

[]

(5) 第8章 43.

次のような Item クラスと Book クラスが定義されている場合、Item クラスのインスタンスが生成されるコードとして正しいものを選びなさい。(2つ選択)

- 1. class Item { }
- 2. Class Book extends Item { }

- A. Item item = null;
- B. Item item;
- C. Item item = new Book();
- D. Item item = new Item();
- E. new Item();

[] []

第8回 カプセル化

(1) 第5章3.

「カプセル化の維持に欠かせない原則として、正しいものを選びなさい。(1つ選択)

- A. データ隠蔽
- B. 情報隠蔽
- C. 抽象化
- D. データ抽象

[]

(2) 第5章4.

データ隠蔽を実現するためには、フィールドをどのように修飾すればよいか。正しいものを選びなさい。(1つ選択)

- A. public final
- B. public static
- C. private
- D. private static
- E. private final

[]

(3) 第5章5.

データ隠蔽の説明として、もっとも適切なものを選びなさい。(1つ選択)

- A. 関係するデータとそのデータを扱う処理をひとまとめにする
- B. 属性の公開範囲を制限する
- C. getter や setter などのアクセサメソッドを提供する
- D. 実装の詳細を隠蔽する

[]

Java I 第 6 回小テスト

(4) 第 8 章 22.

以下の中から、private メソッドにアクセスできるものを選びなさい。(3 つ選択)

- A. サブクラスの public メソッド
- B. 同じクラスの private メソッド
- C. super() を使用するサブクラスのコンストラクタ
- D. オーバーロードされたメソッド
- E. 同じクラスのメソッド内に定義した自インスタンスを参照する this 変数

[] [] []

(5) 第 8 章 27.

適切にカプセル化し、維持するために必要な修飾子として、正しいものを選びなさい。(1 つ選択)

- A. final
- B. abstract
- C. public
- D. private

[]

第9回 オーバーライド

(1) 第8章 30.

以下の中から、情報隠蔽に関わりの深いキーワードを選びなさい。(3つ選択)

- A. ポリモーフィズム
- B. カプセル化
- C. アクセサメソッド
- D. パッケージ
- E. インタフェース
- F. インスタンス化

[] [] []

(2) 第8章 38.

継承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. 何らかのクラスを継承したクラスは継承できない
- B. 1つのクラスが複数のスーパークラスを継承できる
- C. 1つのスーパークラスから複数のサブクラスを定義できる
- D. サブクラスはスーパークラスのすべてのメソッドとフィールドを引き継ぐわけではない

[] []

(3) 第 8 章 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 SuperClass {  
2.     private int num = 10;  
3.     void test() {  
4.         System.out.println(num);  
5.     }  
6. }
```

```
1. public class SubClass extends SuperClass {  
2.     private int num = 20;  
3.     public static void main(String[] args) {  
4.         SubClass sub = new SubClass();  
5.         sub.test();  
6.     }  
7. }
```

- A. コンパイルエラーになる
- B. 実行時に例外がスローされる
- C. 10 が表示される
- D. 20 が表示される

[]

(4) 第8章 51.

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 SuperClass {  
2.     public void print() {  
3.         System.out.println("super");  
4.     }  
5. }
```

```
1. public class SubClass extends SuperClass {  
2.     public void print() {  
3.         System.out.println("sub");  
4.     }  
5.     public static void main(String[] args) {  
6.         SuperClass s = new SuperClass();  
7.         SubClass s2 = (SubClass) s;  
8.         s2.print();  
9.     }  
10. }
```

- A. 「super」と表示される
- B. 「sub」と表示される
- C. コンパイルエラーになる
- D. 実行時に例外がスローされる

[]

(5) 第8章 52.

次のプログラムを実行し、「subsuper」と表示されるようにしたい。空欄にあてはまるコードを選びなさい。(1つ選択)

```
1. public class SuperClass {  
2.     String val = "super";  
3. }
```

```
1. public class SubClass extends SuperClass {  
2.     String val = "sub";  
3.     public void test() {  
4.         System.out.println(val + "," + _____);  
5.     }  
6.     public static void main(String[] args) {  
7.         SubClass sub = new SubClass();  
8.         sub.test();  
9.     }  
10. }
```

- A. super(val)
- B. this.val
- C. super.val
- D. this(val)
- E. super().val
- F. SuperClass.val

[]