

氏名

点/104点

【問1】次はJavaに関する記述である。各記述が正しい場合は○を、誤っている場合は×を解答欄に記入しなさい。【各2計50】

- (1) クラスはメソッドのみをもつことができる
- (2) メソッドの戻り値は常に基本型である
- (3) メソッドの戻り値の型は2つ以上宣言できる
- (4) メソッドの戻り値が無いときはnullを指定することができる
- (5) クラスには複数のメソッドをメンバとして宣言できる
- (6) クラスのprivateメンバは、そのクラスからのみアクセス可能なメンバである
- (7) クラスのpublicメンバは、そのクラスからはアクセスできないメンバである
- (8) クラスのメソッドのみがpublicメンバになることができる
- (9) メソッドのオーバーロードはポリモーフィズムを実現する
- (10) コンストラクタの名前はクラス名と同じである
- (11) コンストラクタはオーバーロードができない
- (12) コンストラクタからメソッドを呼び出すことができる
- (13) クラス変数とはクラス単位で準備される変数である
- (14) クラスメソッド内でインスタンス変数にアクセスしても良い
- (15) インスタンス変数はオブジェクト毎に異なる値を保持できる
- (16) Doubleクラスはラップクラスである
- (17) クラス型の変数は参照型である
- (18) 参照型変数はオブジェクトへの参照を保持する
- (19) 参照型変数をメソッドの戻り値にする時、値渡しにされるという
- (20) 参照型変数にはvoidを代入できる
- (21) 既存のクラスを拡張して新しいクラスを宣言することができる
- (22) 新しいクラスが既存のクラスのメンバを受け継ぐことをカプセル化という
- (23) クラスの拡張における既存のクラスをスーパークラスという
- (24) コンストラクタから別のコンストラクタを呼び出すことができる
- (25) デフォルトコンストラクタは戻り値をもたない

【問2】次はJavaで用いる用語である。それぞれの用語に対応する箇所を下のコード中から1つずつ選び記号を解答欄に記入しなさい。ただし、該当するものが無い2つの用語は解答欄に×を記入しなさい。【各2計28】

- (1) クラスを宣言するキーワード
- (2) privateメンバの宣言のための修飾子
- (3) publicメンバの宣言のための修飾子
- (4) オーバーロード

- (5) スーパークラスのコンストラクタの呼び出し
- (6) スーパークラスのクラス名
- (7) サブクラスのクラス名
- (8) ラップクラス
- (9) デフォルトコンストラクタ
- (10) クラスの拡張
- (11) 自身のオブジェクトへの参照
- (12) 参照型変数の変数名
- (13) 基本型変数の変数名
- (14) オブジェクトの生成

■コード1

```
class Stationery{
    private String name;
    public Stationery(){
        name=new String("名無し");
    }
    public Stationery(String n){
        name=n;
    }
    public void showStationery(){
        System.out.println("文具名："+name);
    }
}
class Whiteout extends Stationery{
    private int ml;
    public Whiteout(String n, int ml){
        super(n);
        this.ml=ml;
    }
    public void showWhiteout(){
        showStationery();
        System.out.println("容量(ml)："+ml);
    }
}
```

【問3】次はクラス変数dataとそれにアクセスを行う4つのメソッドをもつクラスCarの宣言である。コンパイルが成功するメソッドには○を、エラーになるメソッドには×を解答欄に記入しなさい。【各2計8】

■コード2

```
class Car{
    private static int data;
    public void add_1(int i){
        data += i;
    }
    public static void add_2(int i){
        data += i;
    }
    public static void add_3(int i){
        this.data += i;
    }
    public void add_4(int i){
        Car.data += i;
    }
}
```

メソッド1

メソッド2

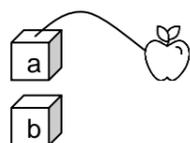
メソッド3

メソッド4

【問4】次はクラス型変数を用いたコードである。それぞれのコードを実行した直後の変数とオブジェクトの参照関係を解答例にしたがい図示しなさい。ここで変数aとbはクラスApple型の変数とし、最初は両方ともにnullが代入されていることとする。また、参照されないオブジェクトは図には示さないこととする。【各2計8】

- (1) a=new Apple();
a=b;
- (2) a=new Apple();
a=new Apple();
- (3) a=new Apple();
b=new Apple();
- (4) a=new Apple();
b=new Apple();
a=b;

■解答例



【問5】次の各コードを実行したときの画面出力を正確に答えなさい。クラスA~Cの宣言は下に示すものとする。【各2計10】

- (1) A a=new A();
- (2) A a=new A(0);
- (3) B b=new B();
- (4) B b=new B(0);
- (5) C c=new C();

■コード3

```
class A{
    public A(){
        this(0);
        System.out.println("A0");
    }
    public A(int i){
        System.out.println("A1");
    }
}
class B extends A{
    public B(){
        System.out.println("B0");
    }
    public B(int i){
        super(i);
        System.out.println("B1");
    }
}
class C extends B{
}
```

解答欄

【問1】

| | | | | |
|--------|--------|--------|--------|--------|
| (1) × | (2) × | (3) × | (4) × | (5) ○ |
| (6) ○ | (7) × | (8) × | (9) ○ | (10) ○ |
| (11) × | (12) ○ | (13) ○ | (14) × | (15) ○ |
| (16) ○ | (17) ○ | (18) ○ | (19) × | (20) × |
| (21) ○ | (22) × | (23) ○ | (24) ○ | (25) ○ |

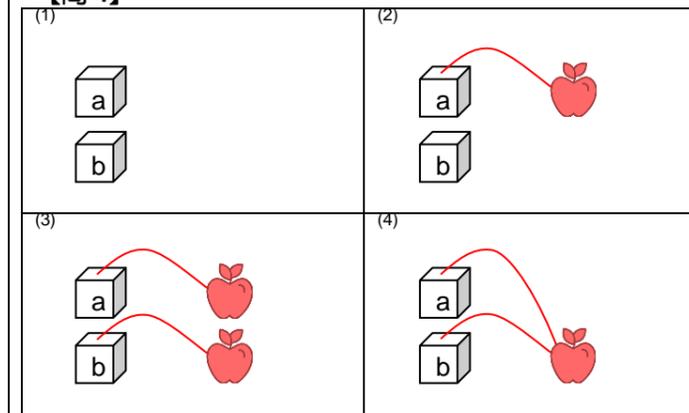
【問2】

| | | | | |
|--------|--------|--------|--------|--------|
| (1) A | (2) C | (3) F | (4) E | (5) K |
| (6) I | (7) G | (8) × | (9) × | (10) H |
| (11) L | (12) B | (13) J | (14) D | |

【問3】

| | | | |
|-------|-------|-------|-------|
| (1) ○ | (2) ○ | (3) × | (4) ○ |
|-------|-------|-------|-------|

【問4】



【問5】

| | | | | |
|--------------|--------|--------------------|--------------|--------------------|
| (1) A1 A0 | (2) A1 | (3) A1 A0 B0 | (4) A1 B1 | (5) A1 A0 B0 |
|--------------|--------|--------------------|--------------|--------------------|

お疲れ様でした!