

氏名

点/102点

【問1】次はJavaに関する記述です。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】

- クラスはメソッドを複数もつことができない
- メソッドの戻り値がないときは宣言に null を指定する
- メソッドの戻り値の型は 2 つ以上同時に宣言できる
- メソッド名はクラス名と一致させなければいけない
- メソッドから別のメソッドの実行はできない
- メソッドには private や public などのアクセス制限をつけることができる
- 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
- クラスのフィールドは常に public メンバにしなければならない
- 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
- コンストラクタの名前はクラス名と同じである
- コンストラクタはオブジェクトの生成時に自動的に実行される手続きである
- コンストラクタはオーバーロードができる
- コンストラクタ同士は互いに呼出すことができる
- コンストラクタを明示的に宣言しないとき、デフォルトコンストラクタが暗黙的に宣言される
- 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
- String クラスは文字列を管理するクラスである
- Double クラスはラップクラスである
- クラス型の変数は参照型である
- 参照型変数をメソッドの仮引数にする時、参照渡しにされるという
- 参照されなくなったオブジェクトがメモリから自動的に削除されることをカプセル化という
- 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
- 拡張されたクラスが既存のクラスのメンバを受け継ぐことを継承という
- クラスの拡張における新しいクラスをサブクラスという
- スーパークラスの private メンバにサブクラスからのアクセスは可能である
- サブクラスはコンストラクタを持つことはできない

【問2】次はJavaで用いる用語や命令の説明です。コード1の記号と合せて、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26】

- A は基本型変数 Park を宣言しています
- B は参照型変数 name を宣言しています

- C や I はアクセス制限を宣言します
- D は自身のオブジェクトへの参照です
- E のコンストラクタは1つの仮引数を持ちます
- F と H はクラス名です
- G は Park を拡張して Garden を宣言することを意味します
- H はサブクラスです
- I は同じクラスからのみアクセスが可能となるアクセス制限を宣言します
- J は protected メンバの変数 plants です
- K はスーパークラスのコンストラクタを呼び出します
- L はメソッドのオーバーロードを宣言します
- M はクラス Garden のオブジェクトを作成します

■コード1

```
class Park{
  private String name;
  private int price;
  public Park(String n, int p){
    this.name = n;
    this.price = p;
  }
  public void showPark(){
    System.out.print(name);
    System.out.println(" (入場料"+price+"円)");
  }
}
class Garden extends Park{
  private int plants;
  public Garden(String n, int p, int pt){
    super(n, p);
    plants = pt;
  }
  public void showGarden(){
    showPark();
    System.out.print(plants);
    System.out.println("種の植物を展示!");
  }
}
class Intermediate2{
  public static void main(String[] args){
    Garden g=new Garden("鳥取花回廊", 1000, 400);
    g.showGarden();
  }
}
```

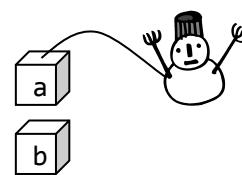
【問3】次に示すStringクラスとMathクラスを用いたコードについて(1)~(5)の各命令を実行したときの画面出力を答えなさい。【各2計10】

- ```
String str = new String("Java");
```
- System.out.print(str.charAt(str.length()-1));
  - System.out.print(str.indexOf('a'));
  - System.out.print(str.toLowerCase());
  - System.out.print(Math.pow(2,3));
  - System.out.print(Math.max(1, Math.min(2, 3)));

【問4】次はクラス型変数を用いたコードです。各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Snowman型の変数とし、最初は両方ともにnullが代入されている、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します。【各2計10】

- a = new Snowman();  
b = a;
- a = new Snowman();  
b = new Snowman();
- a = new Snowman();  
a = new Snowman();
- a = new Snowman();  
b = a;  
b = new Snowman();
- a = new Snowman();  
b = new Snowman();  
b = a;

■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解答欄に答えなさい。クラス Super と Sub の宣言は下に示すものとします。【各2計6】

- Super sup = new Super();
- Super sup = new Super("b");
- Sub sub = new Sub("d");

■コード2

```
class Super
{
 public Super(){
 System.out.print("a");
 }
 public Super(String s){
 this();
 System.out.print(s);
 }
}
class Sub extends Super
{
 public Sub(){
 super("b");
 System.out.print("c");
 }
 public Sub(String s){
 this();
 System.out.print(s);
 }
}
```

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)  | (2)  | (3)  | (4) | (5)  |
| (6)  | (7)  | (8)  | (9) | (10) |
| (11) | (12) | (13) |     |      |

【問3】

|     |     |     |     |     |
|-----|-----|-----|-----|-----|
| (1) | (2) | (3) | (4) | (5) |
|-----|-----|-----|-----|-----|

【問4】

|         |         |
|---------|---------|
| (1)<br> | (2)<br> |
| (3)<br> | (4)<br> |
| (5)<br> |         |

【問5】

|     |     |     |
|-----|-----|-----|
| (1) | (2) | (3) |
|-----|-----|-----|

お疲れ様でした!!