

点/104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】

- クラスはメソッドのみをもつことができる
- 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
- メソッドの戻り値は 0 個でもよい
- 戻り値とは、呼び出し側から与えられた値をメソッド側で受け取るための変数である
- メソッドから直接コンストラクタの実行はできない
- クラスの protected 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
- クラスのメソッドは常に 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カプセル化を実現する
- コンストラクタはクラスのオブジェクトが削除されると実行される手続きである
- 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
- コンストラクタからそのクラスのメソッドを呼び出すことができる
- コンストラクタ同士は互いに呼出すことができる
- クラス変数はクラス単位で準備される変数である
- インスタンスメソッド内でクラス変数にアクセスしてもよい
- Math クラスは文字列を管理するクラスである
- Integer クラスはラッパクラスである
- クラス型の変数と配列変数は共に参照型の変数である
- クラス型の変数をメソッドの戻り値にする時、参照渡しにされるという
- 生成したオブジェクトはユーザの責任でメモリ上から解放しなければならない
- 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
- super(引数)を用いれば、スーパークラスの実行したいコンストラクタを指定できる

【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 はクラス Stationery を宣言します
- B は name 型の String という変数を宣言します
- C はコンストラクタのオーバーロードを宣言します

- D はガーベッジコレクションを実行します
- E のメソッドは戻り値を持ちません
- F は戻り値を呼び出し元に返します
- G はスーパークラスです
- H はクラス変数を宣言します
- I はアクセス制限を private に設定します
- J は String 型の戻り値を1つもつスーパークラスのコンストラクタを呼び出します
- K は戻り値の型を public 型に宣言します
- L はクラス変数やクラスメソッドを宣言するキーワードです
- M は配列要素の確保やクラスのオブジェクトを生成する命令です

■コード1

```
class Stationery
{
  private String name;
  public Stationery(){
    name="未定";
  }
  public Stationery(String n){
    this();
    name=n;
  }
  public String getStationery(){
    return name+",";
  }
}
class Whiteout extends Stationery
{
  private int amount;
  public Whiteout(String n, int a){
    super(n);
    amount=a;
  }
  public String getWhiteout(){
    return getStationery()+"容量"+amount+"ml";
  }
}
class Intermediate2
{
  public static void main(String[] args){
    Whiteout mp=new Whiteout("良く消え～",7);
    System.out.println("修正ペン,"+mp.getWhiteout());
  }
}
```

【問3】下に示す6つのメソッドの宣言に関して、次の各質問に該当するメソッド(複数該当する場合はすべて)を記号で答えなさい。【各2計10】

- 戻り値をもたないメソッドは?
- 戻り値の型が boolean であるメソッドは?
- メソッド名が set であるメソッドは?
- オーバーロードの関係になるメソッドの組は?
※○と○、のように組が分かるように解答すること
- 参照渡しを行う戻り値をもつメソッドは?

■メソッドの宣言

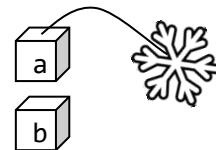
- int show(int i){...}
- void show(){...}
- int get(boolean b){...}

- double get(boolean b){...}
- boolean set(String s){...}
- void Set(int i, double d){...}

【問4】次はクラス型の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 Snow 型の変数とし、最初は両方とも null が代入されています。またどの変数からも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 Snow();
b = new Snow();
- a = new Snow();
a = b;
- a = new Snow();
b = a;
- a = null;
a = null;
- a = new Snow();
a = new Snow();
a = b;
- a = new Snow();
b = new Snow();
a = b;
- a = new Snow();
b = new Snow();
a = null;

■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。各クラスの宣言はコード3に示します。【各2計8】

- Ticket tc=new Ticket();
tc.showT();
- Ticket tc=new Ticket(-1000);
tc.showT();
- ThemeParkTicket tp=new ThemeParkTicket(2500);
tp.showT();
tp.showP();
- ThemeParkTicket tp=new ThemeParkTicket(5000, 2);
tp.showT();
tp.showP();

■コード3

```
class Ticket{
  private int price;
  public Ticket(){
    price=0;
  }
  public Ticket(int p){
    this();
    if(p>=0)price=p;
  }
  public void showT(){
    System.out.println(price+"円");
  }
}
class ThemeParkTicket extends Ticket{
  private int ppl;
}
```

```
public ThemeParkTicket(){
  ppl=0;
}
public ThemeParkTicket(int pr){
  super(pr);
  ppl=0;
}
public ThemeParkTicket(int pr, int pp){
  this(pr);
  ppl=pp;
}
public void showP(){
  System.out.println(ppl+"人用");
}
```

解答欄

【問1】

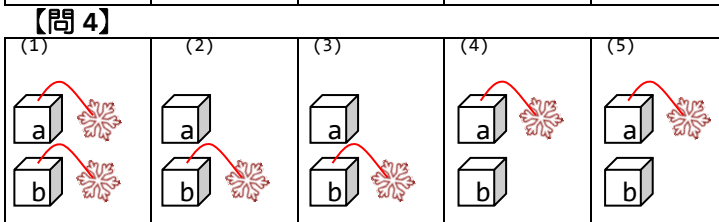
(1)	(2)	(3)	(4)	(5)
×	×	×	○	○
(6)	(7)	(8)	(9)	(10)
○	×	×	×	×
(11)	(12)	(13)	(14)	(15)
○	○	○	○	○
(16)	(17)	(18)	(19)	(20)
×	○	○	○	×
(21)	(22)	(23)	(24)	(25)
○	○	○	×	○

【問2】

(1)	(2)	(3)	(4)	(5)
○	×	○	×	○
(6)	(7)	(8)	(9)	(10)
○	○	×	○	○
(11)	(12)	(13)		
×	○	○		

【問3】

(1)	(2)	(3)	(4)	(5)
b, f	c, d	e	aとb	e



【問5】

(1)	(2)	(3)	(4)
0円	0円	2500円 0人用	5000円 2人用

お疲れ様でした!!