

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void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
- メソッドの仮引数がないときはvoidを指定する
- クラスには複数のメソッドをメンバとして宣言できる
- メソッドから別のメソッドの実行はできない
- 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
- クラスのメソッドのみがpublicメンバになることができる
- 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
- コンストラクタのオーバーロードはできない
- クラス変数はクラス単位で準備される変数である
- インスタンス変数はオブジェクト毎に異なる値を保持できる
- インスタンスメソッド内でクラス変数をアクセスしてもよい
- Stringクラスは文字列を管理するクラスである
- Mathクラスはラッパクラスである
- 参照型変数はオブジェクトへの参照を保持する
- 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はメモリから自動的に削除される
- 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
- 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継承という
- 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
- 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ができる

【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Giftを宣言します
- Bはint型のprivateという変数を宣言します
- Cは自身のオブジェクトへの参照です
- Dは実引数を宣言します
- Eはコンストラクタのオーバーロードを宣言します
- Fのメソッドは引数を持ちません
- Gは呼び出し元に戻り値を返します
- Hはサブクラスです

- Iは数学的な計算を扱うクラスです
- Jはスーパークラスのコンストラクタを呼び出します
- Kはスーパークラスから継承されたメソッドです
- Lはインスタンス変数やインスタンスメソッドを宣言するキーワードです
- MはクラスBirthdayのオブジェクトを作成します

■コード1

```
class Gift{
    private int price;
    public Gift(){
        this.price = -1;
    }
    public Gift(int p){
        this();
        this.price = p;
    }
    public String getGift(){
        return "ギフト("+price+"円)";
    }
}
class Birthday extends Gift{
    private String date;
    public Birthday(int p, String d){
        super(p);
        date = d;
    }
    public String getBirthday(){
        return getGift() + "/誕生日("+date+)";
    }
}
class Intermediate2{
    public static void main(String[] args){
        Birthday b = new Birthday(2500, "11月28日");
        System.out.println(b.getBirthday());
    }
}
```

【問3】次に示すStringクラスとMathクラス、ラッパクラスを用いたコード(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("Programming");
- System.out.println(str.length());
 - System.out.println(str.indexOf('b'));
 - System.out.println(str.charAt(1));
 - System.out.println(Math.pow(2,3));
 - System.out.println(Integer.parseInt("23"));

【問4】次はマイルを管理するクラスMileageの宣言と、これを利用するクラスです。このコードを実行したとき、(1)~(4)の各命令が画面に出力するマイル数を解答欄に答えなさい。【各2計8】

■コード2

```
class Mileage{
```

```
private static int rate = 1;
private int mile;

public Mileage(int m){
    mile = m;
}

public static void setRate(int r){
    Mileage.rate = r;
}

public void add(int m){
    mile += m * Mileage.rate;
}

public void show(){
    System.out.println(mile);
}

class Intermediate4{
    public static void main(String[] args){
        Mileage a = new Mileage(10);
        a.show();
        Mileage b = new Mileage(15);
        b.show();
        Mileage c = a;
        Mileage.setRate(3);
        a.add(2);
        Mileage.setRate(4);
        b.add(3);
        b.show();
        c.show();
    }
}
```

【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】

- First ft = new First();
- First ft = new First(1);
- Second sd = new Second();
- Third td = new Third(1);

■コード3

```
class First{
    public First(){
        System.out.println("F1");
    }
    public First(int a){
        System.out.println("F2");
    }
}

class Second extends First{
    public Second(){
        System.out.println("S1");
    }
    public Second(int a){
        this();
        System.out.println("S2");
    }
}

class Third extends Second{
    public Third(){
        System.out.println("T1");
    }
    public Third(int a){
        super(a);
        System.out.println("T2");
    }
}
```

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)
11	-1	r	8 or 8.0	23

【問4】

(1)	(2)	(3)	(4)
10	15	27	16

【問5】

(1)	(2)	(3)	(4)
F1	F2	F1 S1	F1 S1 S2 T2

お疲れ様でした!!