

氏名

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】

- クラスはメソッドをもつことができない
- メソッドの戻り値は常に基本型である
- メソッドの戻り値はreturn文を用いて返す
- メソッドの引数は0個でもよい
- 仮引数とは、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
- クラスには複数のメソッドをメンバとして宣言できる
- 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
- コンストラクタはクラスのオブジェクトが生成されると最初に実行される手続きである
- コンストラクタの名前はクラス名と同じである
- コンストラクタから別のコンストラクタを呼び出すことができる
- クラス変数とはクラス単位で準備される変数である
- クラスメソッド内ではインスタンス変数をアクセスできない
- 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
- 参照型変数はオブジェクトへの参照を保持する
- 参照型変数はメソッドの仮引数にはできない
- 生成したオブジェクトをユーザの責任でメモリ上から解放する必要はない
- 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
- 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サブクラスという
- コンストラクタは必ず明示的に1つ以上宣言しなければならない
- 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

- privateメンバの宣言のための修飾子
- protectedメンバの宣言のための修飾子
- オブジェクトの生成を行うキーワード
- コンストラクタの宣言
- 値渡しを行うメソッドの宣言
- 参照渡しを行うメソッドの宣言
- ラップクラス
- クラスの拡張を行うためのキーワード
- スーパークラスのクラス名
- スーパークラスのコンストラクタの呼び出し
- 自身のオブジェクトへの参照
- 参照型変数の変数名

■コード1

```
class Worker{
  protected String name;
  protected int age;
  public Worker(String n, int a){
    name=n;
    age=a;
  }
  public void setWorker(String n){
    this.name=n;
  }
  public void setWorker(int a){
    this.age=a;
  }
}

class Driver extends Worker{
  private int career;
  public Driver(String n, int a, int c){
    super(n, a);
    career=c;
  }
  public void setDriver(int c){
    career=c;
  }
  public void showDriver(){
    System.out.println("〔運転手〕");
    System.out.println("名前："+name);
    System.out.println("年齢："+age);
    System.out.println("運転歴："+career);
  }
}

class Intermediate2{
  public static void main(String[] args){
    Driver drv=new Driver("福工大太郎", 20, 1);
    drv.showDriver();
  }
}
```

【問3】次に示すStringクラスのオブジェクトstr1とstr2について、(1)~(6)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12】

```
String str1="Hello";
String str2="75";
(1) System.out.print(str1.charAt(0));
(2) System.out.print(str1.length());
```

- System.out.print(str1.toUpperCase());
- System.out.print(str2.indexOf('5'));
- System.out.print(Integer.parseInt(str2));
- System.out.print(Math.max(5,10));

【問4】次はポイント进行管理するクラスPoint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保有ポイント数を解答欄に答えなさい。【各2計8】

■コード2

```
class Point{
  private static int max=20;
  private int pt;

  public Point(int p){
    pt=p;
  }
  public static void setMax(int m){
    Point.max=m;
  }
  public void showPoint(){
    System.out.print("保有ポイント：");
    if(pt>Point.max)
      System.out.println(Point.max);
    else
      System.out.println(pt);
  }
}

class Intermediate4{
  public static void main(String[] args){
    Point a=new Point(15);
    a.showPoint();
    Point b=new Point(22);
    b.showPoint();

    Point.setMax(10);

    Point c=new Point(7);
    c.showPoint();
    a.showPoint();
  }
}
```

【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AとBの宣言は下に示すものとします。【各2計6】

- A a=new A(0);
- A a=new B(0);
- A a=new B("A");

■コード3

```
class A{
  public A(){
    this(0);
    System.out.println("A0");
  }
  public A(int i){
    System.out.println("A1");
  }
}

class B{
  public B(){
    this(0);
    System.out.println("B0");
  }
  public B(String s){
    System.out.println("B1");
  }
}
```

```
class B extends A{
  public B(int i){
    super(i);
    System.out.println("B1");
  }
  public B(String s){
    System.out.println("B2");
  }
}
```

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)
A	L	J	B	M
(6)	(7)	(8)	(9)	(10)
E	G	F	×	H
(11)	(12)	(13)	(14)	
I	K	D	C	

【問3】

(1)	(2)	(3)
H	5	HELLO
(4)	(5)	(6)
1	75	10

【問4】

(1)	(2)
保有ポイント： 15	保有ポイント： 20
(3)	(4)
保有ポイント： 7	保有ポイント： 10

【問5】

(1)	(2)	(3)
A1	A1 B1	A1 A0 B2

お疲れ様でした!!