

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】

- (1) クラスはメソッドをもつことができない
- (2) メソッドの戻り値は常に基本型である
- (3) メソッドの戻り値はreturn文を用いて返す
- (4) メソッドの仮引数は基本型変数のみを持つことができる
- (5) 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
- (6) 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
- (7) クラスのpublicメンバはそのクラス内外からアクセスできるメンバである
- (8) 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
- (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る
- (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) コンストラクタ同士は互いに呼出すことができる
- (14) インスタンス変数はオブジェクト単位で準備される変数である
- (15) 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
- (16) Stringクラスは数学的な計算を管理するクラスである
- (17) Mathクラスはラップクラスである
- (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仮引数にする時、参照渡しにされるという
- (20) 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
- (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) 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オーバーロードという
- (23) 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
- (24) 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できない
- (25) 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28】

- (1) **A**はクラスBookを宣言します
- (2) **B**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
- (3) **C**はアクセス制限をprivateに設定します
- (4) **D**はtitle型のStringという変数を宣言します

- (5) **E**の自身のオブジェクトへの参照です
- (6) **F**はオーバーロードの宣言です
- (7) **G**は変数titleの値を呼び出し元に返します
- (8) **H**は仮引数をもたないメソッドを宣言します
- (9) **I**はクラスの拡張を行うキーワードです
- (10) **J**は2つのクラス変数tとeを宣言します
- (11) **K**はint型の仮引数を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
- (12) **L**は戻り値の型をString型に宣言します
- (13) **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
- (14) **N**はクラスメソッドgetNum()を実行します

■コード1

```

class Book{
  static private int num=0;
  private String title;
  static int getNum(){
    return num;
  }
  public Book(String t){
    Book.num++;
    this.title = t;
  }
  public String getTitle(){
    return title;
  }
}

class Java extends Book{
  private int edition;

  public Java(String t, int e){
    super(t);
    this.edition = e;
  }
  public String getJava(){
    return getTitle()+"第"+edition+"版";
  }
}

class Intermediate2{
  public static void main(String[] args){
    Java bk1 = new Java("やさしいJava",6);
    Java bk2 = new Java("人体の限界",1);
    System.out.println(Book.getNum()+"冊");
    System.out.println(bk1.getJava());
    System.out.println(bk2.getJava());
  }
}
  
```

【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)~(4)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8】

- (1) System.out.print(str1.indexOf('a'));
- (2) System.out.print(str1.length());
- (3) System.out.print(str2.charAt(1));
- (4) System.out.print(Integer.parseInt(str2));

【問4】次はメソッドのオーバーロードに関するコードです。各コードの組においてオーバーロードとなる組には○を、そうではない組には×を解答欄に記入しなさい。【各2計10】

- (1) int func(){...}
void func(){...}
- (2) int func(int a){...}
int func(int b){...}
- (3) void func(int a){...}
int func(int a, double b){...}
- (4) int func(int a){...}
int func(double a){...}
- (5) void func(double b){...}
void func(Double b){...}

【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2に示します。【各2計8】

- (1) Building m1 = new Building();
m1.showBuilding();
- (2) Mansion m2 = new Mansion(12);
m2.showMansion();
- (3) Mansion m3 = new Mansion(6, 20);
m3.showMansion();
- (4) Mansion m4 = new Mansion(3, 10, 55);
m4.showMansion();

■コード2

```

class Building{
  private int floors;
  private int height;

  public Building(){
    floors = 1;
    height = 3;
  }
  public Building(int f){
    this();
    floors = f;
  }
  public Building(int f, int h){
    floors = f;
    height = h;
  }
  public void showBuilding(){
    System.out.print(floors+"階建/高さ"+height+"m/");
  }
}

class Mansion extends Building{
  private int families;

  public Mansion(){
    families = 0;
  }
  public Mansion(int f){
    super(f);
    families = 0;
  }
  public Mansion(int f, int h){
    super(f, h);
    families = 0;
  }
}
  
```

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)	(14)	

【問3】

(1)	(2)	(3)	(4)
-----	-----	-----	-----

【問4】

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

【問5】

(1)
(2)
(3)
(4)

お疲れ様でした!!