

情報工学科 年 組 学籍番号
 氏名

問1 次の J a v a に関する記述である。各記述が正しい場合は○を、誤っている場合は×を解答欄に記述せよ。

【各2点 計40点】

- (1) クラスはフィールド(変数)とメソッドをもつことができる
- (2) メソッドは2つ以上の戻り値をもつことはできない
- (3) メソッドは必ず1つの引数を持たなくてはならない
- (4) 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
- (5) クラスの private メンバは、そのクラスからのみアクセス可能なメンバである
- (6) クラスは private メンバと public メンバを同時に持つことはできない
- (7) クラスの public メンバはどこからでもアクセスできるため、宣言する際には注意が必要である
- (8) クラスのカプセル化はポリモーフィズムを実現する
- (9) コンストラクタはクラスのオブジェクトが生成されと最初に実行される
- (10) コンストラクタはメソッドと同様にオーバーロードができる
- (11) クラス変数はクラス単位で準備される変数である
- (12) クラスメソッド内でインスタンス変数をアクセスしてもよい
- (13) ラップクラスとは、変数の基本型に関する様々な便利機能を集約したクラスである
- (14) クラス型の変数は、参照型ではなく、基本型である
- (15) 参照型の変数は、オブジェクトへの参照を保持する
- (16) 生成したクラスのオブジェクトはユーザの責任でメモリ上から解放する必要がある
- (17) 既存のクラスを拡張して新しいクラスを宣言することはできない
- (18) 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
- (19) 継承における既存のクラスをサブクラスという
- (20) コンストラクタはクラスのメンバではないため、継承されない

解答欄

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

問2 次の J a v a がもつ代表的な概念または機能である。それぞれの概念・機能に対応する宣言・箇所を下のコード中の記号から1つずつ選び解答欄に記述せよ。

【各2点 計20点】

- (1) メソッド
- (2) private メンバ
- (3) public メンバ
- (4) オーバーロード
- (5) クラス変数
- (6) クラスメソッド
- (7) 値渡し
- (8) 参照渡し
- (9) ガーベッジコレクション
- (10) コンストラクタ

解答欄

(1) d	(2) b	(3) e	(4) f
(5) a	(6) g	(7) h	(8) i
(9) j	(10) c		

```

class A{
    private static int s=0;
    private int i;
    private String n;
}

public A0{
    s++;
    i=s;
    n="none";
}

public int func10{
    return(i);
}

public String func20{
    return(n);
}

public void func3(int a){
    i=a;
}

public void func3(String a){
    n=a;
}

public static void func40{
    System.out.println(s);
}
    
```

```

class Intermediate2{
    public static void main(String[] args){
        A a=new A0();
        int i=10;
        String str="abc";
        a.func3(i);
        a.func3(str);
        a=null;
    }
}
    
```

問3 次の各 main()メソッドを実行した実行画面を正確に答えよ。クラス A の宣言は問2で用いたものであり、クラス B、C、D は下に示したものである。

【各3点 計24点】

- (1) class Intermediate3_1{
 public static void main(String[] args){
 A.func40();
 }
 }
- (2) class Intermediate3_2{
 public static void main(String[] args){
 A a1=new A0();
 A a2=new A0();
 A.func40();
 }
 }
- (3) class Intermediate3_3{
 public static void main(String[] args){
 B b=new B0();
 b.func(10);
 }
 }
- (4) class Intermediate3_4{
 public static void main(String[] args){
 B b=new B0();
 b.func(20,30);
 }
 }
- (5) class Intermediate3_5{
 public static void main(String[] args){
 B b=new B0();
 b.func(40.5);
 }
 }
- (6) class Intermediate3_6{
 public static void main(String[] args){
 C c=new C(10);
 }
 }
- (7) class Intermediate3_7{
 public static void main(String[] args){
 D d=new D0();
 }
 }
- (8) class Intermediate3_8{
 public static void main(String[] args){
 D d=new D(10);
 }
 }

解答欄

(1) 0	(2) 2	(3) M2	(4) M2 M3
(5) M4 M2 M1	(6) C1 C2	(7) C1 D1	(8) C1 C2 D2

```

class B{
    public void func0{
        func(10);
        System.out.println("M1");
    }
    public void func(int i){
        System.out.println("M2");
    }
    public void func(int i, int j){
        func(i);
        System.out.println("M3");
    }
    public void func(double d){
        System.out.println("M4");
        func0();
    }
}
    
```

```

class C{
    public C0{
        System.out.println("C1");
    }
    public C(int i){
        this0();
        System.out.println("C2");
    }
}
    
```

```

class D extends C{
    public D0{
        System.out.println("D1");
    }
    public D(int i){
        super(i);
        System.out.println("D2");
    }
}
    
```

問4 次に示す個人を管理するクラスを宣言するコードを答えよ。但し、下に示す3つの要望を有することとする。 【16点】

クラス名 : Individual
 メンバー一覧 :

[インスタンス変数]

識別子	型	意味
age	int	年齢(0以上の数値)
name	String	名前

[インスタンスメソッド]

識別子	戻り値	引数	機能
setAge	無し	int	年齢の設定
setName	無し	String	名前の設定
getAge	int	無し	年齢の取得
getName	String	無し	名前の取得

(要望)

1. クラスをカプセル化する
2. 不正な年齢(負値)が代入されないようにする
3. コンストラクタを用いて年齢は0に設定し、名前は"名無し"に設定する

解答欄

(裏面へ記述してください)

```
class Individual
{
    private int age;
    private String name;

    public Individual(){
        age=0;
        name="名無し";
    }
    public void setAge(int i){
        if(i>=0)age=i;
    }
    public void setName(String s){
        name=s;
    }
    public int getAge(){
        return(age);
    }
    public String getName(){
        return(name);
    }
}
```