

過去問：11K225中間試験 (12K225-07-2)

2012年10月30日

ライフスタイルデザイン研究センター 金井秀明

1. 抽象クラス図形「Shape」がある。Shapeには「図形の面積」を計算する抽象メソッドvoid getArea()が定義されている。そのような抽象クラスを定義せよ。(配点6点)

(試験時間)

2011年7月05日(火) 2時限目

試験開始30分までは入室可能

試験開始30分以降は、退出可能。ただし、再入室は不可。

(範囲)

授業で行った内容(第7回, 抽象クラスまで, インタフェースは範囲外)

(持ち込み)

なし

(禁止事項)

- ・カンニングと見なされる行為
- ・他学生への迷惑行為(試験妨害)
- ・飲食

これらの事項が守られない場合、「不正行為」実行者として、当該学生を処理します。

2. 「問題1」の抽象クラス図形「Shape」を継承して、以下要領を満たす三角形を表すクラス「Triangle」を作成せよ。(配点10点)

- (1) 底辺 $a$ 、高さ $h$ のフィールド(double型)をもち、全てのフィールドをカプセル化する。ただし、仮にTriangleクラスのサブクラスを作成しても、それらのフィールドにはアクセスできないようにする。
- (2) コンストラクタ: 引数あり ( $a, h$ )
- (3) 面積 $S$ を求めるメソッドvoid getArea()  
三角形の面積 $S$ は「 $a*h/2.0$ 」で求まる。

3. 次のPrintStringクラスのprintStrメソッドを、2つのint型引数を持つメソッドにオーバーロードする。PrintStr(int x, int y)に引数に数値(1,2)を指定した場合、そのメソッドの実行結果は次のようになった。printやprintln文を使わずに処理部分を埋めて、プログラムを完成させなさい。(配点6点)

実行結果

```
1
2
```

```
class PrintString {
    void PrintStr(int x) {
        System.out.println(x);
    }

    void PrintStr(int x, int y) {
        処理部分
    }
}
```

・実行コード

```
class Mon3 {
    public static void main(String[] args) {
        PrintString ps = new PrintString();
        ps.PrintStr(1,2);
    }
}
```

4. 銀行口座を表すAccountクラスを定義せよ。クラスの仕様は以下である。

(配点15点：すべて正解の時)

(a) フィールド (各1点)

- 口座番号: private String型 id
- 預金残高: private int型 balance

(b) コンストラクタ (各2点)

- 引数なし: 口座番号は0000, 預金残高は0に初期化すること。
- 引数あり: 引数として口座番号, 預金残高を与え, それぞれのフィールド値を初期化する。

(c) メソッド (3点と4点)

- info(): 口座の現在の状態 (口座番号と預金残高) を文字列型のデータで返すメソッド。
- remittance: 送金処理を行うメソッド。第1引数に送金金額 (amount), 第2引数に送金先の口座を表すAccountインスタンス (remittee)。送金元, 送金先の口座状況はinfo()メソッドを用いて表示すること。

実行コード:

```
public class Mon4 {  
    public static void main(String[] args) {  
        Account remitter = new Account("123", 10000);  
        Account remittee = new Account("456", 10000);  
        remitter.remittance(2000, remittee);  
    }  
}
```

実行結果

```
rem: 2000  
remitter: account: 123 Balance: 8000  
remittee: account: 456 Balance: 12000
```

5. 「4」で定義したAccountクラスを継承して, VIP用口座を表す「VIPAccount」クラスを定義せよ。(配点6点: すべて正解の時)

(a) コンストラクタ (2点)

- 引数あり: 引数として口座番号, 預金残高を与え, それぞれのフィールド値を初期化する。

(b) メソッド (3点)

- remittance: 送金処理を行うメソッド。第1引数に送金金額 (amount), 第2引数に送金先の口座を表すAccountインスタンス (remittee)。動作や表示はAccountクラスのremittanceと同様である。ただし, 送金額が10000以上のときには, 「Thank you」とも表示する。

実行コード:

```
public class sample {  
    public static void main(String[] args) {  
        VIPAccount remitter = new VIPAccount("123", 100000);  
        Account remittee = new Account("456", 10000);  
        remitter.remittance(20000, remittee);  
    }  
}
```

実行結果

```
rem: 20000  
remitter: account: 123 Balance: 80000  
remittee: account: 456 Balance: 30000  
Thank you
```

6. 次のプログラムの実行結果を答えなさい。そのような実行結果を得る過程を説明せよ。（配点6点）

```
class Calc {
    int digit = 2;
    int add(int x) {
        return x+digit;
    }
}

class SubCalc extends Calc {
    void printAdd(int x) {
        System.out.println(x + "+" + digit + "=" + add(x));
    }
}

public class Mon6 {
    public static void main(String[] args) {
        SubCalc sCalc = new SubCalc();
        sCalc.printAdd(3);
    }
}
```

7. 次のプログラムの実行結果を答えなさい。さらに、そのような実行結果を得る過程を説明せよ。（配点6点）

```
class Calc {
    int digit = 2;
    int add(int x) {
        return x+digit;
    }
}

class SubCalc extends Calc {
    int digit = 3;
    int add(int x) {
        return (x+digit)*digit;
    }
}

public class Mon7 {
    public static void main(String[] args) {
        SubCalc sCalc = new SubCalc();
        System.out.println(sCalc.add(3));
    }
}
```

8. 次のプログラムの実行結果を答えなさい。さらに、そのような実行結果を得る過程を説明せよ。（配点8点）

```
class Calc {
    static double rate;
    Calc(double x) {
        rate = x;
    }
    private int weight = 100;
    private int calc(int x) {
        return (int)(x*rate)-weight;
    }
    void printCalc(int x) {
        System.out.println(x + "->" + calc(x));
    }
}

public class Mon8 {
    public static void main(String[] args) {
        Calc money = new Calc(1.05);
        Calc money2 = new Calc(1.15);
        money.printCalc(20000);
        System.out.println("Rate="+money.rate);
        money2.printCalc(20000);
        System.out.println("Rate="+money2.rate);
    }
}
```

9. 次のプログラムの実行結果を答えよ。その際、なぜ、そのような実行結果になったのかを説明せよ。説明では「実行したらこの結果になった」というような説明は不可とする。（配点6点）

```
class Parent {
    void display() {
        System.out.println("parent");
    }
}

class Child extends Parent {
    void display() {
        System.out.println("child");
    }
}

public class Mon9 {
    public static void main(String[] args) {
        Parent p = new Child();
        p.display();
    }
}
```

10. 次のプログラムの実行結果を答えなさい。そのような実行結果を得る過程を説明せよ。（配点8点）

```
class A {
    static int total =0;
    A() {System.out.print("1"); total =total+5;}
    A(int x) {System.out.print(x); total =total+4;}
}

class B extends A {
    B() {System.out.print("3"); total =total+3;}
    B(int x) {System.out.print("4"); total =total+x;}
    B(String x) {
        super(2);
        System.out.print("5"); total =total+1;
    }
}

class Mon10 {
    public static void main(String args[]) {
        new B("Mon10");
        new A(1);
        new A();
        new B();
        System.out.println();
        System.out.println(B.total);
    }
}
```

11. 車を表現するCarクラスを定義せよ。（配点6点：すべて正解の時）

(a)フィールド（1点）

- ・速度を表すprivateのフィールド: int speed

(b)メソッド（各2点）

- ・加速を行うメソッド accel(): このメソッドを呼び出す度に速度が5ずつ増加する.
- ・速度の値を返すメソッド getSpeed()

定義したCarクラスを利用した例を以下に示す。

・実行コード

```
class Mon12 {
    public static void main(String[] args) {
        Car c1 = new Car();
        System.out.println(c1.getSpeed());
        c1.accel();
        System.out.println(c1.getSpeed());
        c1.accel();
        System.out.println(c1.getSpeed());
    }
}
```

・実行結果

0
5
10

12. 「11」で定義した車クラスCarよりも速度が3倍で加速する「スーパーカー」を表現するスーパーカーのクラス(SuperCar)を作成せよ。ただし、Carクラスを継承して、メソッドのオーバーライドを少なくとも1つ行うこと。この問題で定義したSuperCarクラスを利用した例を以下に示す。(配点7点)

・実行コード

```
class Mon12 {
    public static void main(String[] args) {
        SuperCar sc1 = new SuperCar();
        System.out.println(sc1.getSpeed());
        sc1.accel();
        System.out.println(sc1.getSpeed());
        sc1.accel();
        System.out.println(sc1.getSpeed());
    }
}
```

・実行結果

```
0
15
30
```

13. 次に示す2つのクラス (CalcSum, CalcAvg) があります。それぞれコードの中に共通する処理が含まれています。その共通部分をスーパークラス (CalBase) で実装する形に、CalBaseを定義し、それに伴ってCalcSum, CalcAvgを修正せよ。(配点10点)

```
class CalcSum {
    public double calc(double[] nums) {
        double sum = 0;
        for(int i=0; i<nums.length;i++) {
            sum = sum + nums[i];
        }
        return sum;
    }
}
```

```
class CalcAvg {
    public double calc(double[] nums) {
        double sum = 0;
        for(int i=0; i<nums.length;i++) {
            sum = sum + nums[i];
        }
        return sum / nums.length;
    }
}
```

```
class Mon13 {
    public static void main(String[] args) {
        double[] nums = {1.1, 2.2, 3.3};
        CalcSum cs = new CalcSum();
        System.out.println("total = "+cs.calc(nums));

        CalcAvg ca = new CalcAvg();
        System.out.println("Average = "+ca.calc(nums));
    }
}
```