知識情報・図書館学類　4学年　〔学籍番号〕201811528〔氏名〕春名航亨

情報科学 I 2021 期末試験問題

次の(1)—(5)の解答を問題文の下に記入せよ。教科書、参考書は見て良い。ただし、計算機類の使用は不可。

(1) **クリスマスツリー：演習で行った**直角三角形の例を少し変えて，直立した三角形，クリスマスツリーのような図形を考えてみましょう．高さ6とした例です．

```
     *
    ***
   *****
  *******
 *********
***********
```

ここで、規則性を考えましょう．

- 最下行にはない(0個)が，その上の行には1個，そのまた上の行には2個というように，「行の先頭に空白がある」．第1行には5(=6-1)個であるので，第k行目には6-k個あればよい．
- 星印の数は，1行目が1個，2行目が3個，3行目が5個というように2個ずつふえてゆく．よって第k行目の星印は2k-1個である．

このような「規則性の分析」ができれば，プログラムはできたも同然です．

あとは「空白の繰返し」と「星印の繰返し」で1行を作り，それらをさらに繰り返せばよいだけです．クリスマスツリーの規則性は，第k行(k=1, 2, …, 6)について

- 空白を6-k個
- 星印を2k-1個

でした．これをプログラムにしてみましょう．なお、高さはプログラム実行時に与えるようにしましょう。

〔解答欄〕

```java
class ChristmasTree {
    private static void drawTree(int h) {
        for (int i = 1; i <= h; i++) {
            System.out.println(" ".repeat(h - i) + "*".repeat(2 * i - 1));
        }
    }

    public static void main(String args[]) {
        if (args.length != 1) {
            System.err.println("Wrong Arguments");
            System.exit(1);
        }
        int Height = Integer.parseInt(args[0]);
        drawTree(Height);
    }
}
```

知識情報・図書館学類　4学年　〔学籍番号〕201811528〔氏名〕春名航亨

(2) 硬貨の入れ物のようなクラス、CoinCase を作成しなさい。

500 円、100 円、50 円、10 円、5 円、1 円が、それぞれ何枚あるかを管理する。

- addCoins メソッドで硬貨を追加する。
  引数は硬貨の種類(int)と枚数(int)。

- getCount メソッドで、指定した硬貨が、何枚あるかを取得する。
  引数は硬貨の種類(int)、戻り値は枚数(int)。

- getAmount メソッドで硬貨の総額を取得する。
  戻り値は硬貨の総額(int)。

CoinCase クラスを使用して次のプログラムを作成しなさい。

CoinCase クラスのインスタンスを作成する。
種類と枚数を入力し、addCoins メソッドで硬貨を追加することを 10 回繰り返す。
各硬貨が何枚あるかを表示する。
総額を表示する。
※ 硬貨の種類は 500 円なら整数の 500、100 円なら 100 とし、該当しない数が指定された場合には無視する
〔解答欄〕

```
import java.util.Arrays;
import java.util.Scanner;

class CoinCase {
   int[] num = new int[6];
   int[] sorts = { 500, 100, 50, 10, 5, 1 };

   CoinCase() {
       Arrays.fill(num, 0);
   }

   private int indexOf(int[] arr, int key) {
       for (int i = 0; i < arr.length; i++) {
           if (arr[i] == key)
               return i;
       }
       return -1;
   }

   void addCoins(int sort, int count) {
       int sortIndex = indexOf(sorts, sort);
       if (sortIndex > -1)
           this.num[sortIndex] += count;
   }
```

```
    int getCount(int sort) {
        int sortIndex = indexOf(sorts, sort);
        if (sortIndex > -1)
            return this.num[sortIndex];
        return 0;
    }

    int getAmount() {
        int sum = 0;
        for (int i = 0; i < 6; i++) {
            sum += num[i] * sorts[i];
        }
        return sum;
    }
}

class Coins {
    public static void main(String args[]) {
        CoinCase obj = new CoinCase();
        Scanner scanner = new Scanner(System.in);
        for (int i = 1; i < 11; i++) {
            System.out.println("Attempt:         " + i);
            System.out.print("Input Coin's sort>    ");
            int s = scanner.nextInt();
            System.out.print("Input Coin's count>   ");
            int c = scanner.nextInt();
            obj.addCoins(s, c);
                        System.out.println("Coins>                          "   +
Arrays.toString(obj.num));
            System.out.println("=".repeat(50));
        }
        System.out.println("Sum:    " + obj.getAmount() + "yen");
        System.out.print("(");
        for (int i = 0; i < 5; i++) {
            System.out.printf("%dy: %d, ", obj.sorts[i], obj.num[i]);
        }
        System.out.printf("%dy: %d.)\n", obj.sorts[5], obj.num[5]);

    }
}
```

（3）以下のルールでプログラムを作成せよ
1, 1から順番に数を表示する
2, その数が3で割り切れるなら"Fizz"、5で割り切れるなら"Buzz"、両方で割り切れるなら"FizzBuzz"と表示する
要するに"1 2 Fizz 4 Buzz Fizz 7 8 Fizz Buzz ･･･"と出力される

実行例
1 2 Fizz 4 Buzz Fizz 7 8 Fizz Buzz 11 Fizz 13 14 FizzBuzz 16 17 Fizz 19 Buzz Fizz 22 23 Fizz Buzz 26 Fizz 28 29 FizzBuzz 31 32 Fizz 34

[解答欄]

```
class FizzBuzz {
   public static void main(String args[]) {
       if (args.length != 1) {
           System.err.println("Wrong Arguments");
           System.exit(1);
       }
       int Times = Integer.parseInt(args[0]);
       for (int i = 1; i <= Times; i++) {
           System.out.print((i % 3 > 0 ? "" : "Fizz") + (i % 5 > 0 ? i % 3 > 0 ? i : "" : "Buzz") + " ");
       }
       System.out.println();
   }
}
```

(4)次の機能を持つPlacedRectangleを、Rectangle()のサブクラスとして宣言してください。

- 位置を表すint型のフィールドx, yを持つ
- ３つのコンストラクタを持つ
  (1) 引数なし
  (2) 位置付き
  (3) 位置と大きさ付き
- 位置と大きさ変更するメソッドsetParamsを持つ
- 標準的な文字列表現を返すメソッドtoStringを持つ
  x = 12, y = 34, width = 123, height = 45のとき、
  [(12, 34) [123, 45]]となるものとする。

ただし、Rectangleクラスは下記のように宣言されているとします。

Rectangleクラスの宣言

```
 1:  class Rectangle {
 2:        int width;
 3:        int height;
 4:        Rectangle() {
 5:             setSize(0, 0);
 6:        }
 7:        Rectangle(int width, int height) {
 8:          setSize(width, height);
 9:        }
10:        void setSize(int width, int height) {
11:             this.width = width;
12:             this.height = height;
13:        }
14:        public String toString() {
15:             return "[" + width + "," + height + "]";
16:        }
17:   }
```

```
class PlacedRectangle extends Rectangle {
    int x, y;

    PlacedRectangle() {
        super();
    }

    PlacedRectangle(int x, int y) {
        super();
        setPlace(x, y);
    }

    PlacedRectangle(int x, int y, int width, int height) {
        setParams(x, y, width, height);
    }

    void setParams(int x, int y, int width, int height) {
        setPlace(x, y);
        setSize(width, height);
    }

    void setPlace(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public String toString() {
        return String.format("[(%d, %d) [%d, %d]]", x, y, width, height);
    }
}
```

（5）

1　メソッドのオーバロードとオーバライドの違いを説明しなさい

2　superのキーワードの用途を3つあげなさい

3　次のコードをコンパイルするために必要となる最小限の修正はどうなるか理由をつけて説明せよ

```
final class Aaa
{
   int xxx;
   void yyy(){  xxx=1;  }
}

public class Bbb extends Aaa
{
   final Aaa finalref= new Aaa();
   final void yyy()
   {
```

```
        System.out.println("In method yyy()");
        finalref.xxx=12345;
    }
}
```

[解答欄]

1,

オーバーロードは、

- 同一クラス内で他メソッドと名前は重複していても引数の数や型が異なるメソッドを定義すること

だが、オーバーライドは、

- サブクラス内でスーパークラス内のメソッドと同名のメソッドを定義すること

である。

2,

superキーワードの用途は、

- サブクラスからスーパークラスのコンストラクタを実行すること(super())
- サブクラスからスーパークラスのフィールド変数にアクセスすること(super.xxx)
- サブクラスからスーパークラスのメソッドにアクセスすること(super.yyy())

の3つが挙げられる。

3,

1行目の「final class Aaa」を「class Aaa」とする