

プログラミング A 第7回・演習 回答

1 演習 1

- あ: (a) クラスフィールド x の初期化時に生成され、当該クラスフィールドにより参照されている。
- い: (c) GcQuiz2 のインスタンスは生成されていないため、インスタンスフィールド y が初期化されることはない。
- う: (a) 生成後に a[0] から参照され、a[1] が参照するインスタンス”え”のフィールド x から参照された後、a[2]=a[1] により (a[0] や a[1] からの参照がなくなった後も) a[2] から (インスタンス”え”を介して) 参照されている。
- え: (a) 生成後に a[1] から参照され、a[2]=a[1] により (a[1] からの参照がなくなった後も) a[2] から参照されている。
- お: (b) 生成後に a[2] から参照され、a[2]=a[1] により a[2] からの参照がなくなっている。
- か: (b) 生成後に、a[2] から参照される”お”のフィールド x から参照され、a[2]=a[1] により a[2] からの (インスタンス”お”を介した) 参照がなくなっている。
- き: (b) 一つ目の if 文の条件式の中で 2 回、異なるインスタンスとして生成されるがどこからも参照されていない。
- く: (c) 1 個目の new X(“き”) と 2 個目の new X(“き”) は、インスタンスとしては異なる。従って、一つ目の if 文の条件式が成り立たないため、生成されない。
- け: (a) 二つ目の if 文の条件式の中で生成されたときに、Y のクラスフィールド last から参照され、以降において Y のインスタンスは他には生成されないため、その参照が残っている。
- こ: (a) Y は X のサブクラスであるため、Y のインスタンスは X のインスタンスとも見なすことができる。従って、二つ目の if 文の条件式が成り立つため生成されて a[0] から参照され続けている。

2 演習 2

参考資料: ExceptionExam.java

```
class ExceptionExam {
    public static void main(String[] args) {
        int[] array = new int[3];
        for(int i = 0; i < 4; i++){
            try {
                array[i] = 0;
            } catch(ArrayIndexOutOfBoundsException e) {
                System.out.println(" 例外 : " + e);
            }
        }
        System.out.println(" プログラムを終了します。");
    }
}
```

3 演習 3

参考資料 : Length.java

```
public class Length {

    public static char[] encode(char[] data) {
        StringBuffer encodedData = new StringBuffer();
        int length;
        for (int i = 0; i < data.length; i++) {
            length = 1;
            while (i + 1 < data.length
                && data[i] == data[i + 1]
                && length < 9) {
                length++;
                i++;
            }
            encodedData.append(length);
            encodedData.append(data[i]);
        }
        return encodedData.toString().toCharArray();
    }

    public static char[] decode(char[] data) {
        StringBuffer decodedData = new StringBuffer();
        for(int i = 0; i < data.length; i += 2) {
            // int number = (int) data[i] - '0';
            // if(number > 9) throw new RuntimeException();
            int number = Integer.parseInt("" + data[i]);
            char c = data[i + 1];
            while(number-- != 0) {
                decodedData.append(c);
            }
        }
        return decodedData.toString().toCharArray();
    }

    public static void print(char[] data) {
        for (int i = 0; i < data.length; i++) {
            System.out.print(data[i]);
        }
        System.out.println();
    }

    public static void main(String[] args) {
        char input[] = {
            'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a',

```

```

        'b','b','b','b','b',
        'c','c',
    };
    print(input);
    print(encode(input));
    print(decode(encode(input)));
    print(decode(input));
}
}

```

4 演習 4

参考資料：Element.java

```

abstract class Element {
    abstract void print();
    void add(Element t) {
    }
}

```

参考資料：Text.java

```

class Text extends Element {
    String text = null;
    Text(String s) {
        text = s;
    }
    void print() {
        System.out.println("<t>" + text + "</t>");
    }
}

```

参考資料：Paragraph.java

```

class Paragraph extends Element {
    static int numberOfInstances = 0;
    Element[] children = new Element[10];
    int numberOfChildren = 0;
    int index = 0;
    Paragraph() {
        index += ++numberOfInstances;
    }
    void print() {
        System.out.println("<p" + index + ">");
        for(int i = 0; i < numberOfChildren; i++) {
            children[i].print();
        }
    }
}

```

```
        System.out.println("</p" + index + ">");
    }
    void add(Element t) {
        children[numberOfChildren] = t;
        numberOfChildren++;
    }
}
```
