

計算機ネットワーク I (Java 編) ・ テスト問題用紙

('08年 8月 1日 ・ 13:00 ~ 14:30)

解答上、その他の注意事項

- I. 問題は、問 I ~ IV までである。
- II. 解答用紙の右上の欄に学籍番号・名前を記入すること。
- III. 解答欄を間違えないよう注意すること。
- IV. 解答中の文字 (特に a と d) がはっきりと区別できるよう注意すること。
- V. 持ち込みは不可である。

全ての問に対する補足:

プログラムの空欄を埋める問題では、解答が長くなる可能性があるので、下の省略形 (○囲み文字) を用いても良い。例えば `this==null` と書く代わりに、 $\textcircled{T}==\textcircled{N}$ と書いて良い。

\textcircled{A} ActionListener \textcircled{C} class \textcircled{D} actionPerformed \textcircled{E} ActionEvent
 \textcircled{G} getSource \textcircled{I} implements \textcircled{J} JApplet \textcircled{L} addActionListener \textcircled{M} Math
 \textcircled{N} null \textcircled{P} public \textcircled{Q} equals \textcircled{S} System.out.println \textcircled{T} this \textcircled{V} void
 \textcircled{W} new \textcircled{X} extends

また、参考のために問題用紙の末尾に授業プリントの `UpDownButton.java`, `UpDownButton3.java` のソースを掲載する。

I. 次の各選択問題に答えよ。解答は各問の指示する選択肢から選べ。ただし、特に指定しない限り、選ぶべき選択肢は必ずしも 1つとは限らない。

(i) 次のうち、Java のクラスの名前として使用できないのは、どれか？

- (A). Prakeet! (B). Penguin's (C). Pigeon23 (D). Swallow_T

(ii) 次の Java に関する文章のうち間違っているものはどれか？

(A). Java の文法は C 言語に似ており、そのため、Java のコンパイラは C 言語のソースファイルをコンパイルすることも可能となっている。

(B). Java は中間言語方式を取っている。JDK では中間言語へのコンパイラが javac、中間言語のインタプリタが java である。

(C). Java はセキュリティを考慮して設計されており、遠隔のコンピュータにある実行コードを安全に実行することができる。

(D). Java と JavaScript の違いは主に実行方式 (Java — 中間言語方式、JavaScript — インタプリタ方式) であり、両者の文法はまったく同じである。

(iii) 要素の型が Font 型であるような、ArrayList 型 (サイズ変更可能な配列の型) の変数 fs を宣言したい。正しい書き方を 1 つ、以下の選択肢から選べ。

(A). `Font.ArrayList<> fs = new Font.ArrayList<>();`

(B). `ArrayList.Font<> fs = new ArrayList.Font<>();`

(C). `ArrayList fs = new ArrayList();`

(D). `Font<ArrayList> fs = new Font<ArrayList>();`

(iv) 0 から 5 までの数とその数を 4 で割った余りを次のよう出力したい。

```
x が 0 のとき x%4 は 0
x が 1 のとき x%4 は 1
x が 2 のとき x%4 は 2
x が 3 のとき x%4 は 3
x が 4 のとき x%4 は 0
x が 5 のとき x%4 は 1
```

次のプログラム (の一部) :

```
int x;
for(x=0; x<6; x++) {
    _____;
}
```

の空欄 _____ ? _____ にはいる式を下の選択肢の中から 1 つ選べ。

(A). `System.out.println("x が " + x + " のとき x%4 は " + x%4);`

(B). `System.out.println("x が " + x + " のとき x%4 は " + (x%4));`

(C). `System.out.println("x が 'x' のとき x%4 は 'x%4');`

(D). `System.out.println("x が +x+ のとき x%4 は +(x%4)+");`

II. 次のプログラムは、コマンドライン引数の中の4の倍数の個数をカウントする。

```
public class Count4sMultiple {
    public static void main(String[] args) {
        int i, count=0;
        for(i=0; i< (i) ; i++) {
            int a = (ii) ;
            if (a%4==0) {
                count++;
            }
        }
        System.out.printf("4の倍数は %d個です。%n", count);
    }
}
```

実行例は以下のようになる。

```
% java Count4sMultiple 10 20 30 40 50
4の倍数は 2個です。
% java Count4sMultiple 1 2 4 8 16
4の倍数は 3個です。
```

(i) ~ (ii) の空欄にふさわしい式を以下の選択肢から1つずつ選べ。

(i) の選択肢

(A). sizeof(args) (B). length(args) (C). (sizeof)args (D). args.length

(ii) の選択肢

(A). Integer.parseInt(args+i) (B). String.valueOf(args+i)
(C). Integer.parseInt(args[i]) (D). String.valueOf(args[i])

III. 次の文章は String クラスの substring メソッドの説明の Java™ API 仕様からの抜粋である。

```
public String substring(int beginIndex, int endIndex)
    この文字列の部分文字列である新しい文字列を返します。部分文字列は、指定された beginIndex から始まり、インデックス endIndex - 1 にある文字までです。したがって、部分文字列の長さは endIndex-beginIndex になります。
    例:
        "hamburger".substring(4, 8) では "urge" が返される。
        "smiles".substring(1, 5) では "mile" が返される。
    パラメータ:
        beginIndex - 開始インデックス (この値を含む)
        endIndex - 終了インデックス (この値を含まない)
    戻り値:
        指定された部分文字列
    例外:
        IndexOutOfBoundsException - beginIndex が負の値である場合、endIndex がこの String オブジェクトの長さより大きい場合、あるいは beginIndex が endIndex より大きい場合
```

このメソッドを使用し、テストするプログラムを次のように作成する。

```
public class SubstringTest {
    public static void main(String[] args) {
        String str = args[0];
        try {
            (i)
            System.out.println(str);
        } catch ((ii) e) {
            System.out.println("(The string is not long enough.)");
        }
    }
}
```

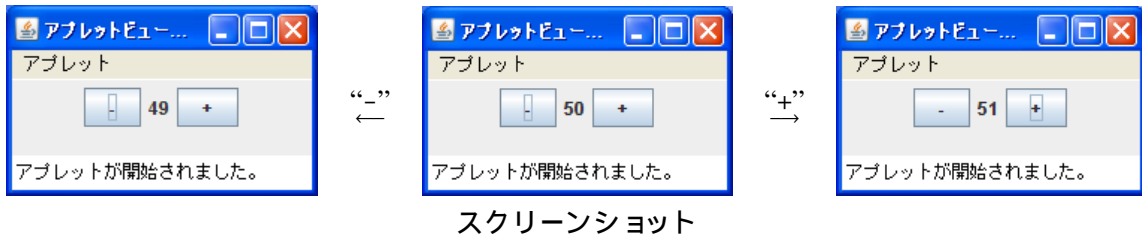
このプログラムは、コマンドライン引数を 1 つ受け取り、その中のインデックス 3 からインデックス 6 (ただし、最初の文字をインデックス 0 と数える) までの 4 文字を表示する。ただし、文字列の文字数が 7 文字未満の場合は、(The string is not long enough.) と表示する。

実行例は次のようになる。

```
prompt> java SubstringTest Tokushima
ushi
prompt> java SubstringTest Ehime
(The string is not long enough.)
```

上のプログラムの空欄を埋めて、プログラムを完成させよ。

IV. 次のプログラムは、“-”、“+” という 2 つのボタンと、整数の表示されたラベルを表示し、“-” ボタンを押せば、ラベル上の数がデクリメントされ、“+” ボタンを押せば、ラベル上の数がインクリメントされる、という Java アプレットである。



空欄を埋めて、プログラムを完成させよ。

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class PlusMinusButton (i) {
    JButton button1, button2;
    JLabel label;
    int x = 50;

    @Override
    public void init() {
        button1 = new JButton("-");
        label = new JLabel(""+x);
        button2 = new JButton("+");
        (ii)
        setLayout(new FlowLayout());
        add(button1); add(label); add(button2);
    }

    public void actionPerformed(ActionEvent e) {
        if ((iii-1)) {
            x--;
        } else if ((iii-2)) {
            x++;
        }
        label.setText(""+x);
    }
}

```

(次のページにつづく)

また、同じ動作をするプログラムを、匿名(無名)クラスを利用して実装する。空欄を埋めて、プログラムを完成させよ。

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class PlusMinusButton2 extends JApplet {
    JButton button1, button2;
    JLabel label;
    int x = 50;

    @Override
    public void init() {
        button1 = new JButton("-");
        label = new JLabel(""+x);
        button2 = new JButton("+");
        button1.addActionListener(
            (iv-1)
        );
        button2.addActionListener(
            (iv-2)
        );
        setLayout(new FlowLayout());
        add(button1); add(label); add(button2);
    }
}
```

参考: (例題 4.6.1: ボタン 2 つを使ってテキストを左右に移動する。)

ファイル UpDownButton.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class UpDownButton extends JApplet implements ActionListener {
    int x=20;
    JButton left, right;

    @Override
    public void init() {
        left = new JButton("Left");
        right = new JButton("Right");
        left.addActionListener(this);
        right.addActionListener(this);
        setLayout(new FlowLayout());
        add(left); add(right);
    }

    @Override
    public void paint(Graphics g) {
        super.paint(g);
        g.drawString("HELLO WORLD!", x, 55);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == left) { // Left が押された
            x-=10;
        } else if (e.getSource() == right) { // Right が押された
            x+=10;
        }
        repaint();
    }
}
```

参考: (例題 4.8.1: ボタン 2 つを使ってテキストを左右に移動する。匿名クラス版)

ファイル UpDownButton3.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class UpDownButton3 extends JApplet {
    int x=20;
    JButton left, right;

    @Override
    public void init() {
        left = new JButton("Left");
        right = new JButton("Right");
        left.addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    x-=10;
                    repaint();
                }
            }
        );
        right.addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    x+=10;
                    repaint();
                }
            }
        );
        setLayout(new FlowLayout());
        add(left); add(right);
    }

    @Override
    public void paint(Graphics g) {
        super.paint(g);
        g.drawString("HELLO WORLD!", x, 55);
    }
}
```


計算機ネットワーク I (Java 編) ・ テスト解答用紙 ('08 年 8 月 1 日)

学籍番号		氏名	
------	--	----	--

I. (4,4,4,4)

(i).		(ii).	
(iii).		(iv).	

II. (4,4)

(i).		(ii).	
------	--	-------	--

III. (5,5)

(i).	
(ii).	

IV. (4,4,4,4)

(i).	
(ii).	
(iii-1).	
(iii-2).	
(iv-1).	
(iv-2).	

(裏面に続く)

