

オブジェクト指向言語・中間テスト問題用紙

(2019年06月14日・11:00～12:00)

解答上、その他の注意事項

- I. 問題は、問I～III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。筆記用具・時計・学生証以外のものは、かばんの中などにしまうこと。
- VI. テストの配点は80点である。(中間テスト40点・期末テスト40点)
合格はレポートの得点を加点して、100点満点中60点以上とする。

すべ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。(必ず○で囲むこと。)

(A) ActionListener (aA) addActionListener (AE) ActionEvent
(K) KeyListener (aK) addKeyListener (KE) KeyEvent
(M) MouseListener (aM) addMouseListener (ME) MouseEvent
(pl) System.out.println (pf) System.out.printf

また、参考のために問題用紙の末尾に授業配布プリントの MouseTest.java, LeftRightButton3.java, LeftRightButton4.java, のソースを掲載する。

I. 次の各多肢選択問題に答えよ。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 一つとは限らない。

(i) 変数 x の値が 5 のとき、次のうち「5 の 3 倍は、15 です。」と出力される文はどれか？

- (A). `System.out.print("{x} の 3 倍は、{x * 3} です。");`
- (B). `System.out.printf("%d の 3 倍は、%d です。", x, x * 3);`
- (C). `System.out.print(x + " の 3 倍は、" + x * 3 + " です。");`
- (D). `System.out.print(x " の 3 倍は、" x * 3 " です。");`

(ii) 次の文章のうち正しいものはどれか？

- (A). Java 言語は、C 言語に対して、上位互換であることが仕様として要求されている。
- (B). Java 言語は、オブジェクト指向言語に分類され、通常、中間言語方式で実行される。
- (C). HTML に埋め込まれ、ブラウザで実行される Java のサブセットを JavaScript という。
- (D). Java 言語は、コンパイル時に型チェックを行わない、いわゆる動的な言語である。

II. 次の枠内の文章は `java.awt.event.KeyEvent` クラス（あるいはそのスーパークラス）のいくつかのメソッドの Java™ API 仕様からの抜粋（問題を解くのに関係ない部分は割愛）である。

```
java.awt.event
```

```
クラス KeyEvent
```

```
コンポーネント内でキー・ストロークが発生したことを示すイベントです。...
```

```
メソッドの詳細
```

```
public boolean isShiftDown()
```

```
このイベントで Shift 修飾子が押されたかどうかを返します。
```

```
public long getWhen()
```

```
このイベントが発生したタイムスタンプと協定世界時の UTC 1970 年 1 月 1 日深夜  
零時との差をミリ秒単位で返します。
```

これらのメソッドを使用するプログラムを次のように作成する。

ファイル名: KeyTrainer.java

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4
5 public class KeyTrainer extends JPanel (i) {
6     private int count = 0;
7     private long time;
8
9     public KeyTrainer() {
10        setPreferredSize(new Dimension(128, 64));
11        setFocusable(true);
12        addKeyListener(this);
13    }
14
15    public void paintComponent(Graphics g) {
16        super.paintComponent(g);
17        g.setFont(new Font(Font.SANS_SERIF, Font.PLAIN, 32));
18        g.drawString("" + count, 40, 40);
19    }
20
21    public void keyTyped(KeyEvent e) {
22        long next = (ii);
23        if ((iii) && next - time <= 300) {
24            count++;
25        } else if (next - time <= 100) {
26            count += 10;
27        }
28        time = next;
29        repaint();
30    }
31
32    /* 他の KeyListener のメソッドと main メソッドは割愛 */
33 }
```

このプログラムは、100 ミリ秒以内の間隔でキーボードのキーをタイプしたときに、10 点を加算する。ただし、Shift キーを押しながらタイプすると“easy mode”になり、300 ミリ秒以内の間隔でキーをタイプすれば 1 点を加算する。

実行例は次のようになる。



100 ミリ秒以内の間隔で
キーを 3 回タイプした時



さらに Shift キーを押しながら
300 ミリ秒以内の間隔で
キーを 2 回タイプした時

- 空欄 (i) を埋めよ。
- 空欄 (ii) には、変数 next にキーをタイプした時刻を代入するため、式が入る。この空欄を埋めよ。
- 空欄 (iii) には、キーがタイプされたときに Shift キーが押されていたかどうかを判定する式が入る。この空欄を埋めよ。

III. 次はテキストフィールドを1つ、「追加」と表示されたボタンを1つ、ラベルを一つ持ち、最初はラベルに“Hello”と表示していて、

- 「追加」ボタンをクリックすると、ラベルの文字列の後ろに絵文字 🐼 が追加される、
- テキストフィールドに文字列を入力すると、ラベルの文字列が入力された文字列に変わる、（「追加」された絵文字も消える）

という Java プログラムである。

実行例は次のようになる。



ファイル名: HelloEmoji.java

```

1 import java.awt.*;
2 import java.awt.event.*;
3
4 import javax.swing.*;
5
6 public class HelloEmoji (i) {
7     private JTextField input;
8     private JButton push;
9     private String text = "Hello";
10    private final String emoji = "🐼";
11    private JLabel label;
12
13    public HelloEmoji() {
14        setPreferredSize(new Dimension(512, 64));
15        input = new JTextField(text, 16);
16        push = new JButton("追加");
17        label = new JLabel(text);
18        label.setFont(new Font(Font.SERIF, Font.PLAIN, 24));

```

```

19     add(input); add(push); add(label);
20     input.addActionListener(this);
21     push.addActionListener(this);
22 }
23
24 public void actionPerformed(ActionEvent e) {
25     Object source = e.getSource();
26     if ( (ii) ) {
27         text = input.getText();
28     } else if ( (iii) ) {
29         text += emoji;
30     }
31     label.setText(text);
32 }
33
34 /* main メソッドは割愛 */
35 }

```

- 空欄 (i) ~ (iii) を埋めよ。

さらにこの HelloEmoji.java を匿名クラス・ラムダ式を用いて次のように同等のプログラム HelloEmoji2.java に書き換える。

ファイル名: HelloEmoji2.java

```

1 import java.awt.*;
2 import java.awt.event.*;
3
4 import javax.swing.*;
5
6 public class HelloEmoji2 (iv) {
7     /* HelloEmoji.java の 7~11 行目と同一のため省略 */
8     public HelloEmoji2() {
9         /* HelloEmoji.java の 14~19 行目と同一のため省略 */
10
11         input.addActionListener(
12             (v)
13         );
14         push.addActionListener(
15             (vi)
16         );
17     }
18
19     /* main メソッドは割愛 */
20 }

```

- 空欄 (iv) を埋めよ。

- 空欄 (v), (vi) を埋めて、HelloEmoji2 のコンストラクターの定義を完成させよ。ただし、空欄 (iv) には匿名クラスを、空欄 (v) にはラムダ式を使用せよ。適宜、`/* HelloEmoji.java の ~ 行目と同じ */` のように省略してよい。

以下に参考のために授業配布プリントの `MouseTest.java`, `LeftRightButton3.java`, `LeftRightButton4.java`, のソースを掲載する。(main メソッドは省略している。)

`MouseTest.java`

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

public class MouseTest extends JPanel implements MouseListener {
    private int x = 50, y = 20;

    public MouseTest() {
        setPreferredSize(new Dimension(150, 150));
        addMouseListener(this);
    }

    public void mouseClicked(MouseEvent e) {
        x = e.getX();
        y = e.getY();
        repaint();
    }

    public void mousePressed(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}

    @Override
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawString("HELLO_WORLD!", x, y);
    }

    /* main メソッドは割愛 */
}
```

`LeftRightButton3.java`

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

public class LeftRightButton3 extends JPanel {
    private int x = 20;

    public LeftRightButton3() {
        setPreferredSize(new Dimension(200, 70));
        JButton lBtn = new JButton("Left");
        JButton rBtn = new JButton("Right");
        lBtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                x -= 10;
                repaint();
            }
        });
        rBtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                x += 10;
                repaint();
            }
        });
        setLayout(new FlowLayout());
        add(lBtn); add(rBtn);
    }
}
```

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

/* main メソッドは割愛 */
}
```

LeftRightButton4.java

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

public class LeftRightButton4 extends JPanel {
    private int x = 20;

    public LeftRightButton4() {
        setPreferredSize(new Dimension(200, 70));
        JButton lBtn = new JButton("Left");
        JButton rBtn = new JButton("Right");
        lBtn.addActionListener(e -> {
            x -= 10;
            repaint();
        });
        rBtn.addActionListener(e -> {
            x += 10;
            repaint();
        });
        setLayout(new FlowLayout());
        add(lBtn); add(rBtn);
    }

    @Override
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawString("HELLO_WORLD!", x, 55);
    }

    /* main メソッドは割愛 */
}
```

オブジェクト指向言語・中間テスト解答用紙（2019年06月14日）

学籍番号		氏名	
------	--	----	--

I. (3×2)

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

II. (4, 6, 6)

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

III. (3, 1, 1, 3, 5, 5)

(i).	
(ii).	
(iii).	
(iv).	
(v).	<pre>/* 匿名クラス使用 */ input.addActionListener(----- ----- ----- ----- ----- ----- ----- ----- ----- -----);</pre>

(裏面に続く)

