

オブジェクト指向言語・中間テスト問題用紙 (2023年 06月 01日・08:50 ~ 10:20)

解答上、その他の注意事項

1. 問題は、問 I ~ II までである。
2. 解答用紙の右上の欄に学籍番号・名前を記入すること。
3. 解答欄を間違えないよう注意すること。
4. 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。
5. 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。
6. 中間テストの配点は 30 点である。

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 LeftRightButton.java, LeftRightButton4.java, のソースを掲載する。(main メソッドは省略している。)

I. 次の (1)～(2) の Java に関する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。

(1) `int` 型の変数 `x` の値が 4 のとき、次のうち、エラーとならず、「4 の 3 倍は 12 です。」と出力して改行する文はどれか?

- (A). `System.out.println("${x} の 3 倍は ${x * 3} です。");`
- (B). `System.out.println(x + " の 3 倍は " + x * 3 + " です。");`
- (C). `System.out.printf("%d の 3 倍は %d です。%n", x, x * 3);`
- (D). `System.out.println(x, " の 3 倍は ", x * 3, " です。");`
- (E). `System.out.println(x + " の 3 倍は " + x + x + x + " です。");`
- (F). `System.out.println(String.format("%d の 3 倍は %d です。", x, x * 3));`

(2) 次の文章のうち、正しいものはどれか?

- (A). Java は、型の安全性を実行時にだけチェックする動的な型付け言語である。
- (B). Java のクラスは、（直接の）スーパークラスを、たかだか一つしか指定することができない。
- (C). Java はメモリー管理を自動化しているため、配列の範囲外をアクセスしてもエラーや例外にならない。
- (D). Java の言語仕様は JavaScript の言語仕様の上位互換になるように定められている。

II. 次の枠内の文章は `javax.swing.JTextComponent` クラスの `getText` メソッド、`javax.swing.JComponent` クラスの `setFont` メソッド、`javax.swing.JLabel` クラスの `setText` メソッド、`java.awt.Font` クラスの `decode` メソッド、`deriveFont` メソッドの Java™ API 仕様からの抜粋である。（問題を解くのに関係ない部分は割愛している。）なお、`javax.swing.JTextComponent` は `javax.swing.JTextField` のスーパークラス、`javax.swing.JComponent` は `javax.swing.JLabel` のスーパークラスである。

`javax.swing`

クラス `JTextComponent`

メソッドの詳細

`getText`

```
public String getText()
```

この `TextComponent` に格納されたテキストを返します。

戻り値:

テキスト

`javax.swing`

クラス `JComponent`

メソッドの詳細

setFont

```
public void setFont(Font font)
```

このコンポーネントのフォントを設定します。

パラメーター:

font - このコンポーネントに指定する Font

javax.swing

クラス JLabel

メソッドの詳細

setText

```
public void setText(String text)
```

このコンポーネントが表示する一行のテキストを定義します。

パラメーター:

text - このコンポーネントが表示する一行

java.awt

クラス Font

メソッドの詳細

decode

```
public static Font decode(String str)
```

str 引数で説明されている Font を返します。

パラメーター:

str - フォント名

戻り値:

str で指定される Font オブジェクト

deriveFont

```
public Font deriveFont(float size)
```

現在の Font オブジェクトを複製し、それに新しいサイズを適用することによって、新しい Font オブジェクトを返します。

パラメーター:

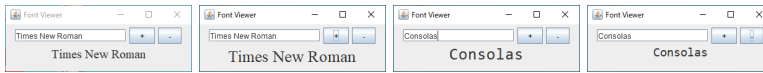
size - 新しい Font のサイズ

戻り値:

新しい Font オブジェクト

次の FontViewer クラスは、1つのテキストフィールドと1つのラベル、「+」ボタン、「-」ボタンを持つ GUI アプリケーションである。上記のメソッドを用いて、テキストフィールドに与えられた文字列（フォント名）で指定されたフォントでラベルにフォント名を表示し、「+」ボタンでラベルのフォントのサイズを 1.1 倍に変更し、「-」ボタンでラベルのフォントのサイズを 1.1 分の 1 に変更する。

次のスクリーンショットは、順に、最初の状態、何度か「+」をクリックしたとき、テキストフィールドに「Consolas」と入力したとき、何度か「-」をクリックしたとき、の画面である。



実行例 1

実行例 2

実行例 3

実行例 4

FontViewer クラスは JPanel クラスを継承している。以下の問に答えよ。

ファイル名: FontViewer.java

```

1  import java.awt.*;
2  import java.awt.event.*;
3  import javax.swing.*;
4
5  public class FontViewer (1) {
6      float size = 20;
7      Font font;
8      String name = "Times New Roman";
9
10     JTextField input; JLabel label;
11     JButton plus, minus;
12
13     public FontViewer() {
14         setPreferredSize(new Dimension(300, 75));
15         input = new JTextField(name, 16);
16         label = new JLabel(name);
17
18         font = (2);
19         font = (3);
20         (4);
21
22         plus = new JButton("+");
23         minus = new JButton("-");
24         input.addActionListener(this);
25         plus.addActionListener(this);
26         minus.addActionListener(this);
27         add(input); add(plus); add(minus); add(label);
28     }
29
30     public void actionPerformed(ActionEvent e) {
31         Object obj = e.getSource();
32         if (obj == input) {
33             name = input.getText();
34             (5);
35             font = (2);
36         } else if (obj == plus) {
37             size *= 1.1;
38         } else if (obj == minus) {
39             size /= 1.1;
40         }
41         font = (3);
42         (4);

```

```

43     }
44
45     /* main は省略する */
46 }

```

(1) の空欄を埋めよ。

(2) には、name 変数で与えられた文字列で指定されたフォントを表す式が入る。この式を答えよ。

(3) には、font 変数で与えられたフォントの複製に size 変数で与えられたサイズを適用した新しいフォントを表す式が入る。この式を答えよ。

(4) には、label オブジェクトのフォントを font 変数で与えられたフォントに設定する式が入る。この式を答えよ。

(5) には、label オブジェクトの表示する一行を name 変数で与えられた文字列にする式が入る。この式を答えよ。

さらに、この FontViewer.java をラムダ式を用いて次のように同等のプログラム FontViewer2.java に書き換える。やはり FontViewer2 クラスは JPanel クラスを継承している。ただし、空欄 (2), (3), (4) は FontViewer.java の該当する箇所と同じ式が入る。以下の問に答えよ。

ファイル名: FontViewer2.java

```

1  import java.awt.*;
2  import java.awt.event.*;
3  import javax.swing.*;
4
5  public class FontViewer2 [空欄 (6)] {
6      float size = 20;
7      Font font;
8      String name = "Times New Roman";
9
10     public FontViewer2() {
11         setPreferredSize(new Dimension(300, 75));
12         JTextField input = new JTextField(name, 16);
13         JLabel label = new JLabel(name);
14
15         font = [空欄 (2)];
16         font = [空欄 (3)];
17         [空欄 (4)];
18
19         JButton plus = new JButton("+");
20         JButton minus = new JButton("-");
21         input.addActionListener([空欄 (7)]);
22         plus.addActionListener([空欄 (8)]);
23         minus.addActionListener([空欄 (9)]);
24         add(input);
25         add(plus); add(minus);
26         add(label);
27     }
28
29     /* main は省略する */
30 }

```

(6) の空欄を埋めよ。

(7), (8) の空欄に入る式を、それぞれラムダ式を用いて書け。ただし、FontViewer.java の 33 行目から 35 行目と同じ部分は△と、41 行目から 42 行目と同じ部分は▽と省略して良い。なお、(9) は (8) とほとんど同じなので、解答しなくて良い。

以下に参考のために授業配布プリントの LeftRightButton.java, LeftRightButton4.java のソースを掲載する。

ファイル LeftRightButton.java

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.*;
4
5 public class LeftRightButton extends JPanel implements ActionListener {
6     private int x = 20;
7     private JButton lBtn, rBtn;
8
9     public LeftRightButton() {
10         setPreferredSize(new Dimension(200, 70));
11         lBtn = new JButton("Left");
12         rBtn = new JButton("Right");
13         lBtn.addActionListener(this);
14         rBtn.addActionListener(this);
15         setLayout(new FlowLayout());
16         add(lBtn); add(rBtn);
17     }
18
19     @Override
20     public void paintComponent(Graphics g) {
21         super.paintComponent(g);
22         g.drawString("HELLO WORLD!", x, 55);
23     }
24
25     public void actionPerformed(ActionEvent e) {
26         Object source = e.getSource();
27         if (source == lBtn) {
28             x -= 10;
29         } else if (source == rBtn) {
30             x += 10;
31         }
32         repaint();
33     }
34
35     public static void main(String[] args) { /* 省略 */ }
36 }
```

ファイル LeftRightButton4.java

```
1 import javax.swing.*;
2 import java.awt.*;
3
4 public class LeftRightButton4 extends JPanel {
5     private int x = 20;
6
7     public LeftRightButton4() {
8         setPreferredSize(new Dimension(200, 70));
9         JButton lBtn = new JButton("Left");
10        JButton rBtn = new JButton("Right");
11        lBtn.addActionListener(e -> {
12            x -= 10;
13            repaint();
14        });
15        rBtn.addActionListener(e -> {
16            x += 10;
17            repaint();
18        });
19        setLayout(new FlowLayout());
20        add(lBtn); add(rBtn);
21    }
22
23    @Override
24    public void paintComponent(Graphics g) {
```

```
25         super.paintComponent(g);
26         g.drawString("HELLO WORLD!", x, 55);
27     }
28
29     public static void main(String[] args) { /* 省略 */
30 }
```

オブジェクト指向言語・中間テスト解答用紙（2023 年 06 月 01 日）

学籍番号		氏名	
------	--	----	--

I. (3, 3)

(1)		(2)	
-----	--	-----	--

II. (2,3,3,3,3,2,4,4)

(1)			
(2)		(3)	
(4)		(5)	
(6)			
(7)			
(8)			

