

# オブジェクト指向言語・中間テスト問題用紙 (2021年 06月 17日・08:50 ~ 09:35)

## 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 40 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 KeyTest.java, LeftRightButton.java, LeftRightButton3.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 (i)~(ii) 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。

(i) int 型の変数 x の値が 4 のとき、次のうち「4 の 2 倍は、8 です。」と出力される文はどれか？

- (A). System.out.print('x の 2 倍は、x \* 2 です。');
- (B). System.out.print("\${x} の 2 倍は、\${x \* 2} です。");
- (C). System.out.printf("%d の 2 倍は、%d です。", x, x \* 2);
- (D). System.out.print(x + " の 2 倍は、" + x \* 2 + " です。");
- (E). System.out.print(x + " の 2 倍は、" + x + x + " です。");
- (F). System.out.print(x " の 2 倍は、" x \* 2 " です。");

(ii) 次の文章のうち、正しいものはどれか？

- (A). Java では、一つのファイルに public なクラスは一つしか定義できない。
- (B). Java から、型宣言を取り去ったインタプリタ方式の言語を JavaScript と呼ぶ。
- (C). 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。
- (D). Java の言語仕様は C 言語の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 java.util.LinkedList クラスのコンストラクターといくつかのメソッド、java.awt.Color クラスの getHSBColor 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。)

LinkedList クラスは両端キュー (Deque) とリスト (List) インタフェースの二重リンク・リストによる実装である。両端キューとは両端 (先頭と末尾) で挿入と削除をサポートするコレクションで、“deque” は “double ended queue” の省略であり、「デック」と発音される。リストは順序付けられたコレクションである。

```
java.util
```

```
クラス LinkedList<E>
```

```
LinkedList
```

```
public LinkedList()
```

```
空のリストを作成します。
```

```
addLast
```

```
public void addLast(E e)
```

```
指定された要素をこのリストの最後に追加します。
```

```
パラメーター:
```

```
    e – 追加する要素
```

```
removeFirst
```

```
public E removeFirst()
```

```
このリストの最初の要素を取得して削除します。
```

戻り値:  
このリストの最初の要素

### **removeLast**

```
public E removeLast()
```

このリストの最後の要素を取得して削除します。

戻り値:  
このリストの最後の要素

### **get**

```
public E get(int index)
```

このリスト内の指定された位置にある要素を返します。

パラメーター:  
index - 返される要素のインデックス

戻り値:  
このリスト内の指定された位置にある要素

### **size**

```
public int size()
```

このリスト内にある要素の数を返します。

戻り値:  
このリスト内の要素数

java.awt

## クラス **Color**

### **getHSBColor**

```
public static Color getHSBColor(float h, float s, float b)
```

HSB カラーモデルに指定された値に基づいて、Color オブジェクトを作成します。

s 成分と b 成分は、0 と 1 の間の浮動小数点値 (0.0 から 1.0 までの範囲の数値) にするように入力してください。h 成分は、任意の浮動小数点数値にできます。この数の下限が減算され、0 から 1 の間の小数部が作成されます。この小数点数に 360 が乗算され、HSB カラーモデルの色相角度が生成されます。

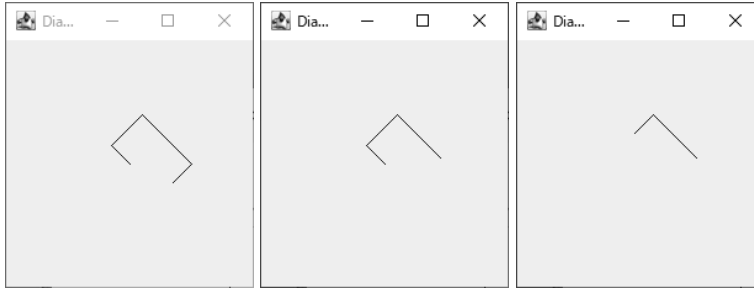
パラメーター:  
h - 色相成分  
s - 色の彩度  
b - 色の明度

戻り値:  
指定された色相、彩度、明度を持つ Color オブジェクト

次の DiaDraw クラスは、これらのメソッドを使って、キー入力を受け取り、'w' キーが押されれば左上、'r' キーが押されれば右上、'z' キーが押されれば左下、'c' キーが押されれば右下、へ約

14 ピクセル分の線分を描画して現在位置を移動する、バックスペースキー（'\b' という特殊文字に対応する）が押されれば、末尾の線分を削除する、DEL キー（'\177' という特殊文字に対応する）が押されれば、先頭の線分を削除する、's'、'd' キーが押されれば、それぞれ色相を -60°、60° 変化させる、という GUI アプリケーションである。

次のスクリーンショットは、順に、“wwrrrrrrcccccczzzs” というキーを入力したとき、そこから BackSpace キーを 4 回入力したとき、さらに DEL キーを 5 回入力したとき、の画面である。



実行例 1

実行例 2

実行例 3

DiaDraw 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 Dia.java

```

1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4 import java.util.LinkedList;
5
6 public class DiaDraw (i) {
7     LinkedList<int[]> points = new LinkedList<>();
8     int hue = 0, x = 100, y = 100;
9
10    public DiaDraw() {
11        setPreferredSize(new Dimension(200, 200));
12        setFocusable(true);
13        addKeyListener(this);
14    }
15
16    public void keyTyped(KeyEvent e) {
17        char k = e.getKeyChar();
18        switch (k) {
19            case '\b':
20                int[] p = (ii);
21                x = p[0]; y = p[1];
22                break;
23            case '\177':
24                (iii);
25                break;
26            case 's':
27                hue -= 60; break;
28            case 'd':
29                hue += 60; break;
30            case 'w':
31                (iv); x -= 10; y -= 10; break;
32            case 'r':
33                (iv); x += 10; y -= 10; break;
34            case 'z':
35                (iv); x -= 10; y += 10; break;
36            case 'c':
37                (iv); x += 10; y += 10; break;
38        }
39        repaint();
40    }
41    public void keyPressed(KeyEvent e) {}

```

```

42 public void keyReleased(KeyEvent e) {}
43
44 @Override
45 public void paintComponent(Graphics g) {
46     super.paintComponent(g);
47     int sz = (v);
48     if (sz < 1) return;
49     float h = hue / 360.0f;
50     g.setColor((vi));
51     int i = 0;
52     int[] p0 = (vii);
53     for (i = 1; i < sz; i++) {
54         int[] p1 = (vii);
55         g.drawLine(p0[0], p0[1], p1[0], p1[1]);
56         p0 = p1;
57     }
58     g.drawLine(p0[0], p0[1], x, y);
59 }
60
61 /* main は省略する */
62 }
63

```

(i) の空欄を埋めよ。

(ii), (iii) には、それぞれ、points の最後と最初の要素を削除して返す式が入る。この式を答えよ。

(iv) には、points の末尾に、現在の変数 x, y の値からなる整数の配列を追加する式が入る。この式を答えよ。

(v) には、points の要素の数を返す式が入る。この式を答えよ。

(vi) には、色相 h、彩度 1、明度 1 を持つ Color オブジェクトを表す式が入る。この式を答えよ。

(vii) には、points の i 番目の要素を返す式が入る。この式を答えよ。

さらに、この DiaDraw.java を匿名クラスを用いて次のように同等のプログラム DiaDraw2.java に書き換える。やはり DiaDraw2 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 DiaDraw2.java

```

1 import javax.swing.*;
2 import java.awt.*;
3 import java.util.Random;
4
5 public class DiaDraw2 (viii) {
6     LinkedList<int[]> points = new LinkedList<>();
7     int hue = 0, x = 100, y = 100;
8     public DiaDraw2() {
9         setPreferredSize(new Dimension(200, 200));
10        setFocusable(true);
11        (ix);
12    }
13
14    /* paintComponent, main は省略する */
15 }
16

```

(viii) の空欄を埋めよ。

(ix) の空欄に入る式を匿名クラスを用いて書け。ただし、DiaDraw.java の 17 行目から 39 行目と同じ部分は、☆と省略せよ。

以下に参考のために授業配布プリントの KeyTest.java, LeftRightButton.java, LeftRightButton3.java のソースを掲載する。

ファイル [KeyTest.java](#)

```
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.*;
4
5  public class KeyTest extends JPanel implements KeyListener {
6      private int x = 50, y = 20;
7
8      public KeyTest() {
9          setPreferredSize(new Dimension(150, 150));
10         setFocusable(true);
11         addKeyListener(this);
12     }
13
14     @Override
15     public void paintComponent(Graphics g) {
16         super.paintComponent(g);
17         g.drawString("HELLO WORLD!", x, y);
18     }
19
20     public void keyTyped(KeyEvent e) {
21         int k = e.getKeyChar();
22         if (k == 'u') {
23             y -= 10;
24         } else if (k == 'd') {
25             y += 10;
26         }
27         System.err.printf("key = %d\n", k);
28         repaint();
29     }
30     public void keyReleased(KeyEvent e) {}
31     public void keyPressed(KeyEvent e) {}
32
33     public static void main(String[] args) { /* 省略 */ }
34 }
35
```

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) { // lBtnが押された
28             x -= 10;
29         }
30         else if (source == rBtn) { // rBtnが押された
31             x += 10;
32         }
33         repaint();
34     }
35
```

```
35     public static void main(String[] args) { /* 省略 */ }
36 }
37
38
```

ファイル LeftRightButton3.java

```
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.*;
4
5  public class LeftRightButton3 extends JPanel {
6      private int x = 20;
7
8      public LeftRightButton3() {
9          setPreferredSize(new Dimension(200, 70));
10         JButton lBtn = new JButton("Left");
11         JButton rBtn = new JButton("Right");
12         lBtn.addActionListener(new ActionListener() {
13             public void actionPerformed(ActionEvent e) {
14                 x -= 10;
15                 repaint();
16             }
17         });
18         rBtn.addActionListener(new ActionListener() {
19             public void actionPerformed(ActionEvent e) {
20                 x += 10;
21                 repaint();
22             }
23         });
24         setLayout(new FlowLayout());
25         add(lBtn); add(rBtn);
26     }
27
28     @Override
29     public void paintComponent(Graphics g) {
30         super.paintComponent(g);
31         g.drawString("HELLO WORLD!", x, 55);
32     }
33
34     public static void main(String[] args) { /* 省略 */ }
35 }
36
```







