オブジェクト指向言語・中間テスト問題用紙
（2025年 06月 12日・ 09:35 〜 10:20） （訂正済）

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。）

Ⓐ `ActionListener` ⒶⒶ `addActionListener` ⒶⒺ `ActionEvent` Ⓚ `KeyListener`
ⒶⓀ `addKeyListener` ⓀⒺ `KeyEvent` Ⓜ `MouseListener` ⒶⓂ `addMouseListener`
ⓂⒺ `MouseEvent` ⓟⓛ `System.out.println` ⓟⓕ `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, 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 (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) Foo.java という Java のソースファイルは `main` メソッドを含み、標準ライブラリー以外には依存しない。これをコンパイルし、実行するためのコマンドの組み合わせで正しいのはどれか？一つ選べ。

	コンパイル	実行
(A).	`javac Foo.java`	`java Foo.class`
(B).	`javac Foo`	`java Foo.class`
(C).	`javac Foo.java`	`java Foo`
(D).	`javac Foo`	`java Foo`
(E).	`javac Foo.java`	`Foo`
(F).	`javac Foo`	`Foo`

(2) 次の文章のうち、正しいものはどれか? すべて選べ。

(A). Java は JavaScript に型宣言を仕様として追加した言語である。
(B). Java はコンパイル時に型チェックを行う、静的型付け言語である。
(C). Java は関数型言語に分類されるプログラミング言語の一つである。
(D). Java では（内部クラスを除くと）一つのソースファイルに 2 つ以上の public なクラスを定義することはできない。

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.lang.Character` クラスの `isDigit` 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を割愛したり、表現を変えたりしている。）

パッケージ java.util

クラス `LinkedList<E>`

`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す。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す。“deque” は “double ended queue” の省略であり、「デック」と発音されます。リストは順序付けられたコレクションです。

コンストラクターの詳細

LinkedList

```
public LinkedList()
```

空のリストを作成します。

メソッドの詳細

add

```
public boolean add(E e)
```

このリスト内の最後に指定された要素を追加します。

パラメーター:
e – このリストに追加される要素
戻り値:
true

add

```
public void add(int index, E e)
```

このリスト内の指定された位置に指定された要素を挿入します。 その位置の現在の要素（ある場合）とそれ以降の要素を右に移動します（インデックスに 1 を加算）
パラメーター:
index – 指定の要素が挿入される位置のインデックス
e – 挿入される要素
throws:
IndexOutOfBoundsException – インデックスが範囲外の場合

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
throws:
IndexOutOfBoundsException – インデックスが範囲外の場合

remove

```
public E remove(int index)
```

このリストの指定された位置にある要素を削除します。 後続の要素を左に移動します（インデックスから 1 を減算）。 リストから削除された要素が返されます。
パラメーター:
index – 削除される要素のインデックス
戻り値:
指定された位置に以前あった要素
throws:
IndexOutOfBoundsException – インデックスが範囲外の場合

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

パッケージ java.lang

クラス Character

Character クラスは、プリミティブ型 char の値をオブジェクトにラップします。クラス Character のオブジェクトには、タイプが char である単一のフィールドが含まれます。また、このクラスは、文字カテゴリ(小文字、数字など。)を決定するた

め、また大文字から小文字への変換やその逆のために多数の静的メソッドを提供します。

メソッドの詳細

isDigit

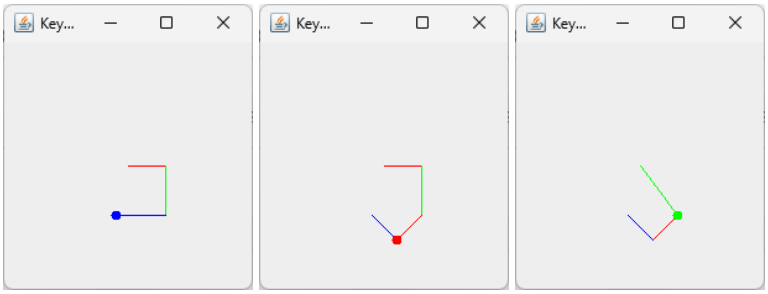
```
public static boolean isDigit(char ch)
```

指定された文字が数字かどうかを判定します。

パラメーター:
　　ch - 判定対象の文字。
戻り値:
　　文字が数字の場合は true、そうでない場合は false。

KeyDraw 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、いくつかの点を編集してそれらつなぐ線分を描画する。最初は点は一つだけで、(100, 100) にあり、現在の点はこの唯一の点を指している。そして'w' キーが押されると上、'a' キーは左、's' キーは下、'd' キーは右、に 10 ピクセルずつ"現在の"点を移動する。'N' キーは現在の点の後、'P' キーは現在の点の前、に現在の点と同じ座標を使って新しい点を挿入し、新しく挿入された点を現在の点とする。'n' キーは後、'p' キーは前、に（点があれば）現在の点を移動する。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、現在の点を削除して、後（なければ前）に現在の点を移動する。数字のキーが押されれば、現在の点の直前の線分の色を 0—BLACK, 1—RED, 2—YELLOW, 3—GREEN, 4—CYAN, 5—BLUE, 6—MAGENTA, 7—WHITE, 8—GRAY, 9—DARK_GRAY、にそれぞれ変更する。 また、現在の点は大きめの点（直径 8 の円盤）で描画する。

次のスクリーンショットは、それぞれ下に示すキーを入力したときの画面である。



NdddNssss3Naaaa5 さらに Pddss1　　さらに pp\b

KeyDraw 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 KeyDraw.java

```
1  import javax.swing.*;
2  import java.awt.*;
3  import static java.awt.Color.*;
4  import java.awt.event.*;
5  import java.util.LinkedList;
6
7  public class KeyDraw                    (1)                    {
8     private static final Color[] colors = {
9             BLACK, RED, YELLOW, GREEN, CYAN, BLUE, MAGENTA,
10            WHITE, LIGHT_GRAY, GRAY };
```

```
11    private LinkedList<int[]> points = new LinkedList<>();
12    private int idx = 0;
13
14    public KeyDraw() {
15       setPreferredSize(new Dimension(200, 200));
16       setFocusable(true);
17       points.add(new int[] { 100, 100, 1 });
18       addKeyListener(this);
19    }
20
21    public void keyTyped(KeyEvent e) {
22       char keyChar = e.getKeyChar();
23       int[] pt = points.get(idx);
24       int sz =                    (2)                    ;
25       switch (keyChar) {
26          case 'w': pt[1] -= 10; break;
27          case 's': pt[1] += 10; break;
28          case 'a': pt[0] -= 10; break;
29          case 'd': pt[0] += 10; break;
30          case 'N':
31             idx++; // fall through
32          case 'P':
33             int[] copy = new int[] { pt[0], pt[1], pt[2] }
34                                (3)                    ;
35             break;
36          case 'n':
37             if (idx < sz - 1) idx++;
38             break;
39          case 'p':
40             if (idx > 0) idx--;
41             break;
42          case '\b': // Backspace
43             if (sz > 1) {
44                                (4)                    ;
45                if (idx >= sz - 1) idx = sz - 2;
46             }
47             break;
48          default:
49             if (                (5)                    ) {
50                pt[2] = keyChar - '0';
51             }
52             break;
53       }
54       repaint();
55    }
56
57    public void keyPressed(KeyEvent e) {}
58
59    public void keyReleased(KeyEvent e) {}
60
61    @Override
62    public void paintComponent(Graphics g) {
63       super.paintComponent(g);
64       int sz =                    (2)                    ;
65       if (sz < 1) return;
66       int[] p0 = points.get(0);
67       for (int i = 1; i < sz; i++) {
68          int[] p1 = points.get(i);
69          g.setColor(colors[p1[2]]);
70          g.drawLine(p0[0], p0[1], p1[0], p1[1]);
71          p0 = p1;
72       }
73       int[] pt = points.get(idx);
74       g.setColor(colors[pt[2]]);
```

```
75          g.fillOval(pt[0] - 4, pt[1] - 4, 8, 8);
76      }
77      /* main は省略する */
78  }
```

(1) の空欄を埋めよ。

(2) には、points の要素の数を表す式が入る（2 箇所で共通）。この式を答えよ。

(3) には、points のインデックス idx に新しい要素 copy を挿入する式が入る。この式を答えよ。

(4) には、インデックス idx にある points の要素を削除する式が入る。この式を答えよ。

(5) には、変数 keyChar の値が、数字 ('0' ～ '9'）かどうかを判定する式が入る。この式を答えよ。

さらに、この KeyDraw.java を匿名クラスを用いて次のように同等のプログラム KeyDraw3.java に書き換える。やはり KeyDraw3 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: *KeyDraw3.java*

```
1  import java.awt.*;
2  import java.awt.event.*;
3  import javax.swing.*;
4
5  public class KeyDraw3 [          (6)          ] {
6      /* colors, points, idx は同じなので省略する */
7
8      public KeyDraw3() {
9          setPreferredSize(new Dimension(200, 200));
10         setFocusable(true);
11         points.add(new int[] { 100, 100, 1 });
12         addKeyListener([          (7)          ]);
13     }
14     /* paintComponent は同じなので省略する */
15     /* main は省略する */
16 }
```

(6) の空欄を埋めよ。

(7) の空欄に入る式を、匿名クラスを用いて書け。ただし、KeyDraw.java と同じ部分は、`/* 元の AB 行目から YZ 行目と同じ */` のように省略して書いて良い。

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

ファイル KeyTest.java

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

public class KeyTest extends JPanel implements KeyListener {
   private int x = 50, y = 20;
   public KeyTest() {
      setPreferredSize(new Dimension(150, 150));
      setFocusable(true);
      addKeyListener(this);
   }
   @Override
   public void paintComponent(Graphics g) {
      super.paintComponent(g);
      g.drawString("HELLO WORLD!", x, y);
   }
   public void keyTyped(KeyEvent e) {
      int k = e.getKeyChar();
      if (k == 'u')      y -= 10;
      else if (k == 'd') y += 10;
      repaint();
   }
   public void keyReleased(KeyEvent e) {}
   public void keyPressed(KeyEvent e) {}
   public static void main(String[] args) {/* 省略 */}
}
```

ファイル LeftRightButton3.java

```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();
         }
      });
      add(lBtn); add(rBtn);
   }
   @Override
   public void paintComponent(Graphics g) {
      super.paintComponent(g);
      g.drawString("HELLO WORLD!", x, 55);
   }
   public static void main(String[] args) {/* 省略 */}
}
```

オブジェクト指向言語・中間テスト解答用紙（2025 年 06 月 12 日）

学籍番号		氏名	

I. (3, 3)

(1)		(2)	

II. (3,3,3,3,3,3,6)

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

授業・テストの感想