オブジェクト指向言語・期末テスト問題用紙
（2020年 07月 30日・ 08:50 ～ 10:20）

解答上、その他の注意事項

1. 問題は、問 Ⅰ～Ⅵ まである。
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. テストの配点は 70 点である。合格はレポート（創造工学部生は「オブジェクト指向言語演習」の課題も含む）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。）

Ⓐ `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 ChangeColor.java, LeftRightButton.java, LeftRightButton4.java, BubbleSort2.java, Point.java, ColorPoint.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) 次のうち、文法上 Java のクラス名として使うことができるのは、どれか?

(A). `123Class`　(B). `Test__`　(C). `Hayashi2217_20`　(D). `Takamatsu-City`

(ii) 次の文章のうち、正しい物はどれか?

(A). Java 言語は仕様を簡潔に保つため、1995 年に登場して以来一度も仕様を変更していない。

(B). Java は中間言語方式を採用しており `.class` ファイルは機種に依存していない。

(C). Java 言語をブラウザーで実行するために、ECMA で仕様を定めたのが JavaScript である。

(D). Java のクラスは、（直接の）スーパークラスを二つ以上指定することはできない。

II. 次の Python のリスト内包表記の値は何か？

(i) `[ x * x for x in [1,2,3,4,5,6] if (x * x) % 3 == 1 ]`

(ii) `[ (x,y) for x in [2,4,5] for y in [1,3,6] if x > y ]`

III. 次の (i)～(ii) の多肢選択問題に答えよ。解答は次の選択肢 (A)～(F) 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。**

(A). C　(B). Fortran　(C). Smalltalk　(D). Prolog　(E). Haskell　(F). Java

(i) この中で命令型（手続き型）に分類される言語はどれか？すべて選べ。

(ii) この中で論理型言語に分類される言語はどれか？すべて選べ。

IV. 次の枠内の文章は `java.util.Random` 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。）

java.util

クラスRandom

Random

```
public Random()
```

新規乱数ジェネレーターを作成します。このコンストラクターは、 乱数ジェネレーターのシードを、このコンストラクターの他の呼び出しと区別可能な値に設定します。

nextInt

```
public int nextInt(int bound)
```

この乱数ジェネレーターのシーケンスを使って、0から指定された値の範囲(0は含むが、その指定された値は含まない)で一様分布のint型の擬似乱数値を返します。（以降略）

パラメーター:
bound - 上限(含まない)。正の値でなければならない
戻り値:
この乱数ジェネレーターのシーケンスを使った、0(含む)からbound(含まない)の範囲で一様分布の、int型の次の擬似乱数値

nextBoolean

```
public boolean nextBoolean()
```

この乱数ジェネレーターのシーケンスを使って、一様分布のboolean型の次の擬似乱数値を返します。（以降略）

戻り値:
この乱数ジェネレーターのシーケンスを使って生成された、一様分布のboolean型の次の擬似乱数値

nextFloat

```
public float nextFloat()
```

この乱数ジェネレーターのシーケンスを使って、0.0から1.0の範囲で一様分布のfloat型の次の擬似乱数値を返します。（以降略）

戻り値:
この乱数ジェネレーターのシーケンスを使って生成された、0.0から1.0の範囲の一様分布のfloat型の次の擬似乱数値

java.awt

クラスColor

Color

```
public Color(int r, int g, int b)
```

範囲 (0 - 255) の指定された赤、緑、青の値を使って、不透明な sRGB カラーを生成します。（以下略）

パラメーター:
r - 赤色成分
g - 緑色成分
b - 青色成分

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。

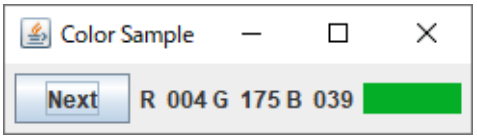
s 成分と b 成分は、0 と 1 の間の浮動小数点値 (0.0 から 1.0 までの範囲の数値) にするようにしてください。h 成分は、任意の浮動小数点数値にできます。この数の下限が減算され、0 から 1 の間の小数部が作成されます。この小数点数に 360 が乗算され、HSB カラーモデルの色相角度が生成されます。

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

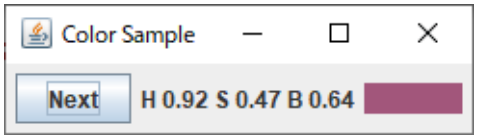
次の ColorSample クラスは、これらのメソッドを使って、“Next” ボタンを押すたびに、乱数で色を生成し、実際の色とともに、その RGB 成分または HSB 成分を表示する GUI アプリケーションである。ColorSample は JPanel を継承している。以下の問に答えよ。

ファイル名 ColorSample.java

```
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.*;
4  import java.util.Random;
5
6  public class ColorSample [          ( i )          ] {
7      private JLabel params, sample;
8      private Random gen;
9
10     public ColorSample() {
11         JButton btn = new JButton("Next");
12         params = new JLabel("R  256 G  256 B  256");
13         sample = new JLabel("                    ");
14         sample.setOpaque(true);
15         sample.setBackground(Color.WHITE);
16         gen = new Random();
17         btn.addActionListener(this);
18         add(btn); add(params); add(sample);
19     }
20
21     public void actionPerformed(ActionEvent e) {
22         if (      ( ii )      ) {
23             int r =      ( iii )      ;
24             int g =      ( iii )      ;
25             int b =      ( iii )      ;
26             params.setText(String.format("R  %03d G  %03d B  %03d",
27                                          r, g, b));
28             sample.setBackground(new Color(r, g, b));
29         } else {
30             float h =      ( iv )      ;
31             float s =      ( iv )      ;
32             float b =      ( iv )      ;
33             params.setText(String.format("H %.2f S %.2f B %.2f",
34                                          h, s, b));
35             sample.setBackground(      ( v )      );
36         }
37     }
38
39     /* main は省略する */
40 }
41
```



実行例 1



実行例 2

(i) の空欄を埋めよ。

(ii) には、乱数ジェネレーター gen を用いた、boolean型の疑似乱数値を表す式が入る。この式を答えよ。

(iii) には、乱数ジェネレーター gen を用いた、0 から 255 までの範囲（両端を含む）の int型の疑似乱数値を表す式が入る。この式を答えよ。

(iv) には、乱数ジェネレーター gen を用いた、0.0 から 1.0 の範囲の float 型の擬似乱数値を表す式が入る。この式を答えよ。

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

なお、getHSBColor メソッドのパラメーターは 1.0 を許すが、(iv) の擬似乱数値を表す式は 1.0 を含まなくてもよい。

さらに、この ColorSample.java をラムダ式を用いて次のように同等のプログラム ColorSample2.java に書き換える。やはり ColorSample2 は JPanel を継承している。以下の問に答えよ。

ファイル名 ColorSample2.java

```
1  import javax.swing.*;
2  import java.awt.*;
3  import java.util.Random;
4
5  public class ColorSample2 [          (vi)          ] {
6      public ColorSample2() {
7          JButton btn = new JButton("Next");
8          JLabel params = new JLabel("R  256 G  256 B  256");
9          JLabel sample = new JLabel("                  ");
10         sample.setOpaque(true);
11         sample.setBackground(Color.WHITE);
12         Random gen = new Random();
13         [          (vii)          ];
14         add(btn); add(params); add(sample);
15     }
16
17     /* main は省略する */
18 }
19
```

(vi) の空欄を埋めよ。

(vii) の空欄に入る式をラムダ式を用いて書け。ただし、ColorSample.java の 22 行目から 36 行目と同じ部分は、☆と省略せよ。

V. 次に定義されるクラス Insect を継承して、

ファイル名 Insect.java

```
1  public class Insect {
2      protected String name;
3      protected int age;
4      public Insect(String n) {
5          age  = 0;  name = n;
6      }
7      public void grow() {
8          age++;
9      }
10     public String firstName() {
11         return name + "の幼虫";
12     }
13     public String secondName() {
14         return name + "のサナギ";
15     }
16     public String lastName() {
17         return name;
18     }
```

```
    public String getName() {
        switch (age) {
            case 0:  return firstName();
            case 1:  return secondName();
            default: return lastName();
        }
    }
}

```

3つのクラス Butterfly, Cicada, Dragonfly を定義する。

ファイル名 Butterfly.java

```
public class Butterfly extends Insect {
    public Butterfly() {
        super("チョウ");
    }

    ( i )
    public String firstName() {
        return "イモムシ";
    }
}

```

ファイル名 Cicada.java

```
public class Cicada extends Insect {
    public Cicada() {
        super("セミ");
    }

    ( i )
    public String firstName() {
        return "ウゴウゴ"; /* ※ 方言です */
    }

    ( i )
    public String getName() {
        if (age < 7) {
            return firstName();
        }
        return lastName();
    }
}

```

ファイル名 Dragonfly.java

```
public class Dragonfly extends Insect {
    public Dragonfly() {
        super("トンボ");
    }

    ( i )
    public String firstName() {
        return "ヤゴ";
    }

    ( i )
    public String getName() {
        if (age == 0) {
            return firstName();
        }
        return lastName();
    }
}

```

ファイル名 InsectTest.java

```
1  public class InsectTest {
2      public static void main(String[] args) {
3          int i;
4          Insect[] data = new Insect[] {
5            new Butterfly(), new Dragonfly(), new Cicada()
6          };
7
8          for (i = 0; i < 3; i++) {
9              for (Insect m: data) {
10                 System.out.print(m.getName());
11                 System.out.print(" ");
12                 m.grow();
13             }
14             System.out.println();
15         }
16     }
17 }
18
```

(i) の空欄にすべて当てはまるアノテーションを答えよ。

(ii). Insect のフィールド name, age はいずれも protected と宣言されているが、これを private に変更すると（ここに掲載された 5 つのクラスをコンパイル・実行するとき）どうなるか？次の選択肢から選べ。

(A). name, age のどちらを private に変更してもエラーにならない。

(B). name は private に変更してもエラーにならないが、 age は private に変更するとエラーになる。

(C). name は private に変更するとエラーになるが、 age は private に変更してもエラーにならない。

(D). name, age のどちらを private に変更してもエラーになる。

(iii). InsectTest クラスの main メソッドを実行するとき、出力はどうなるか？

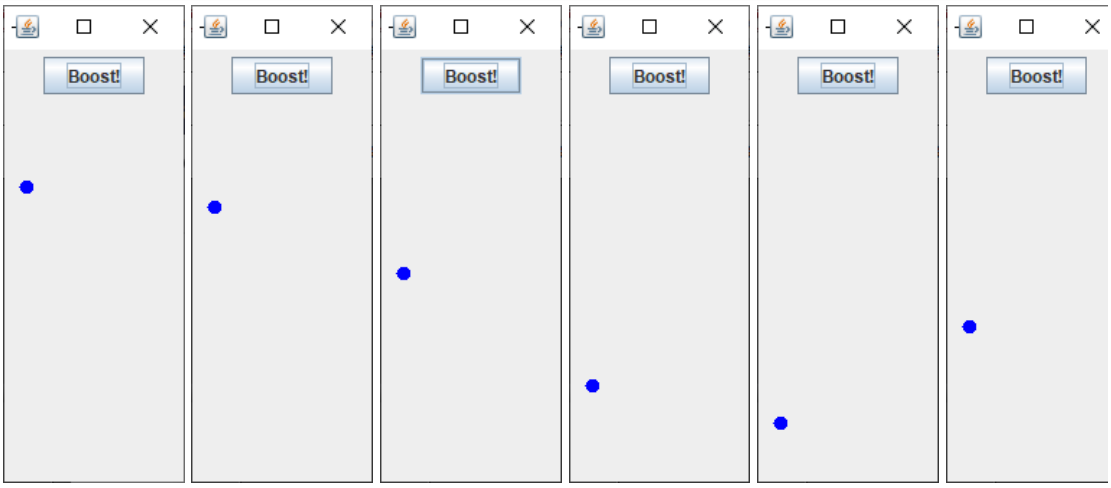
なお、解答用紙に記入するとき、空白の有無や数は気にしなくて良い。

VI. 次の BouncingBall クラスはボールが重力に従って落下し、地面にあたると（反発係数 0.8 で）バウンドする様子をアニメーションにする GUI アプリケーションである。BouncingBall は JPanel を継承している。なお、スレッドを停止したり、再開したりする手段は提供せず、プログラムを簡略化している。

ファイル名 BoucingBall.java

```
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.*;
4
5  public class BouncingBall [          ( i )          ] {
6      private double y = 200, vy = 0;
7      public BouncingBall() {
8          setPreferredSize(new Dimension(60, 300));
9          JButton b = new JButton("Boost!");
10         b.addActionListener(this);
11         add(b);
12         Thread thread = new Thread(this);
13         [          ( ii )          ];
14     }
```

```
15
16      @Override
17      protected void paintComponent(Graphics g) {
18          super.paintComponent(g);
19          g.setColor(Color.BLUE);
20          g.fillOval(10, 290 - (int)y, 10, 10);
21      }
22
23      public void actionPerformed(ActionEvent e) {
24          vy *= 4;
25      }
26
27      public void run() {
28          while (true) {
29              y += vy;
30              vy -= 0.5;
31              if (y < 0) {
32                  y *= -1; vy *= -0.8;
33              }
34                              (iii)                ;
35              try {
36                  Thread.sleep(30);
37              } catch (InterruptedException e) {}
38          }
39      }
40
41      /* main は省略する */
42  }
43
```



空欄 (i) ～ (iii) を埋めてプログラムを完成せよ。（空欄 (ii), (iii) は文法上、式である。）

以下に参考のために授業配布プリントのChangeColor.java, LeftRightButton.java, LeftRightButton4.java, BubbleSort2.java, Point.java, ColorPoint.java のソースを掲載する。

ファイル ChangeColor.java

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

public class ChangeColor extends JPanel
                         implements ActionListener {
    private final Color[] cs = {
    Color.RED, Color.BLUE, Color.GREEN, Color.ORANGE
    };
    private int i = 0;
    private JLabel label;

    public ChangeColor() {
        JButton b = new JButton("Next");
        b.addActionListener(this);          /* 1 */
        label = new JLabel("HELLO WORLD!");
        label.setForeground(cs[i]);         /* 前景色の変更 */
        setLayout(new FlowLayout());        /* 2 */
        add(b); add(label);                 /* 3 */
    }

    public void actionPerformed(ActionEvent e) {
        i = (i + 1) % cs.length;
        label.setForeground(cs[i]);
    }

    public static void main(String[] args) { /* 省略 */ }
}

```

ファイル LeftRightButton.java

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

public class LeftRightButton extends JPanel implements ActionListener {
    private int x = 20;
    private JButton lBtn, rBtn;

    public LeftRightButton() {
        setPreferredSize(new Dimension(200, 70));
        lBtn = new JButton("Left");
        rBtn = new JButton("Right");
        lBtn.addActionListener(this);
        rBtn.addActionListener(this);
        setLayout(new FlowLayout());
        add(lBtn); add(rBtn);
    }

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

    public void actionPerformed(ActionEvent e) {
        Object source = e.getSource();
        if (source == lBtn) {      // lBtnが押された
            x -= 10;
        }
        else if (source == rBtn) { // rBtnが押された
            x += 10;
        }
        repaint();
    }

    public static void main(String[] args) { /* 省略 */ }
}

```

ファイル LeftRightButton4.java

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

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

    public LeftRightButton4() {
        setPreferredSize(new Ðimension(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 WORLÐ!", x, 55);
    }

    public static void main(String[] args) { /* 省略 */ }
}

```

ファイル BubbleSort2.java

```
import javax.swing.*;

import java.awt.*;
import java.awt.event.*;

public class BubbleSort2 extends JPanel implements Runnable, ActionListener {
  /* フィールドは、ほとんど BubbleSort1 と同じ */
  private int[] args = new int[12];
  private final Color[] cs = {Color.REÐ, Color.ORANGE, Color.GREEN, Color.BLUE};
  private volatile Thread thread = null;
  private int i, j;

  private volatile boolean threadSuspended = true; /* 追加のフィールド */

  public BubbleSort2() {
    setPreferredSize(new Ðimension(320, 250));
    JButton step = new JButton("Step");
    step.addActionListener(this);
    setLayout(new FlowLayout());
    add(step);
    startThread();
  }
  /* startTread は BubbleSort1 と同じ */
  private void startThread() {
    if (thread == null) {
      thread = new Thread(this);
      thread.start();
    }
  }

  public void run() {
    while(true) {
      prepareRandomData();
      // バブルソートアルゴリズム
      for (i = 0; i < args.length - 1; i++) {
        for (j = args.length - 1; j > i; j--) {
          if (args[j - 1] > args[j]) { // スワップする。
            int tmp = args[j - 1];
            args[j - 1] = args[j];
            args[j] = tmp;
          }
          repaint();
          /* repaint の後で止まる */
          try {
            synchronized(this) {
              while (threadSuspended) {
```

```
              wait();
            }
            threadSuspended = true;
          }
        } catch (InterruptedException e) {}
      }
    }
  }
  public synchronized void actionPerformed(ActionEvent e) {
    threadSuspended = false;
    notify();
  }
  /* paintComponent, main などは BubbleSort1 と同じ */

  @Override
  public void paintComponent(Graphics g) {
    int k;

    super.paintComponent(g);
    g.setColor(Color.YELLOW);
    g.fillOval(5, 50 + j * 10, 10, 10);
    g.setColor(Color.CYAN);
    g.fillOval(5, 50 + i * 10, 10, 10);
    for (k = 0; k < args.length; k++) {
      g.setColor(cs[k % cs.length]);
      g.fillRect(20, 50 + k * 10, args[k] * 5, 10);
    }
  }

  private void prepareRandomData() {
    int len = args.length;
    for (int k = 0; k < len; k++) {
      args[k] = (int)(Math.random() * len * 4);  // 適当な範囲の乱数
    }
  }

  public static void main(String[] args) { /* 省略 */ }
}

```

ファイル Point.java

```
public class Point {
  // フィールド(メンバー変数)
  public int x;
  public int y;

  // メソッド（メンバー関数）
  public void move(int dx, int dy) {
    x += dx;
    y += dy;
  }

  public double distance() {
    return Math.sqrt(x * x + y * y);
  }

  public void print() {
    System.out.printf("(%d, %d)", x, y);
  }

  public void moveAndPrint(int dx, int dy) {
    print(); move(dx, dy); print();
  }

  // コンストラクター
  public Point(int x0, int y0) {
    x = x0; y = y0;
  }
}

```

ファイル ColorPoint.java

```
public class ColorPoint extends Point {
    public String[] cs = {
        "black", "red", "green", "yellow",
        "blue", "magenta", "cyan", "white" };
```

```
    public String color;

    @Override
    public void print() {
        System.out.printf("<font color='%s'>", getColor()); // 色の指定
        System.out.printf("(%d, %d)", x, y); // super.print(); でも可
        System.out.print("</font>");                        // 色を戻す
    }

    public void setColor(String c) {
        int i;
        for (i = 0; i < cs.length; i++) {
            if (c.equals(cs[i])) {
                color = c; return;
            }
        }
        // 対応する色がなかったら何もしない。
    }

    public ColorPoint(int x, int y, String c) {
        super(x, y);
        setColor(c);
        if (color == null) color = "black";
    }

    public String getColor() {
        return color;
    }
}


```

オブジェクト指向言語・期末テスト解答用紙（2020 年 07 月 30 日）

学籍番号		氏名	

I. (3 × 2)

(ⅰ)		(ⅱ)	

II. (3 × 2)

(ⅰ)		(ⅱ)	

III. (3 × 2)

(ⅰ)		(ⅱ)	

IV. (3,3,3,3,3,3,3,5)

(ⅰ)	
(ⅱ)	
(ⅲ)	
(ⅳ)	
(ⅴ)	
(ⅵ)	
(ⅶ)	

V. (3,5,6)

(ⅰ)		(ⅱ)	
(ⅲ)			

VI. (4 × 3)

(ⅰ)	
(ⅱ)	
(ⅲ)	

授業・テストの感想