オブジェクト指向言語・期末テスト問題用紙
（2023 年 07 月 27 日・ 08:50 ～ 10:20）

（誤り訂正済）

解答上、その他の注意事項

1. 問題は、問 Ⅰ～Ⅶ まである。うち、<u>問 Ⅰ～Ⅲ は中間テストの代替問題である。</u>
   - 中間テストの得点が 7 割に達しなかったものは、 代替問題を解答すれば、7 割を上限として良いほうの点数を採用する。
   - 正当な事情があって中間テストを欠席した者も代替問題を解答すること。（この場合は、上限は設けない。）
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（うち中間テストの代替問題の問 Ⅰ～Ⅲの配点は 30 点）である。合格は「オブジェクト指向言語演習」の課題の得点を加点して、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 ⓐA addActionListener Ⓐ𝙴 ActionEvent Ⓚ KeyListener
ⓐK addKeyListener Ⓚ𝙴 KeyEvent Ⓜ MouseListener ⓐM addMouseListener
Ⓜ𝙴 MouseEvent ⓟl System.out.println ⓟf 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, Guruguru.java, Point.java, ColorPoint.java, Quadratic.kt, SequenceTest.kt のソースを掲載する。（GUI アプリケーションは 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) 変数 str に "132.25" という文字列（String 型）が代入されているとき、 132.25 という浮動小数点数（double 型）を返す式は次のうち、どれか? 一つ選べ。

(A). Double.doubleValue(str)　(B). Double.parseDouble(str)
(C). Double.toString(str)　(D). atof(str)

(2) 次の Java に関する文章のうち、正しいものはどれか? すべて選べ。

(A). Java は、オブジェクト指向言語に分類されるプログラミング言語の一つである。
(B). Java のクラス名には、アルファベット（英字）と数字（インド・アラビア数字）以外の文字は使用できない。
(C). Java は、コンパイル時に型チェックを行うため、実行時に null のメソッドにアクセスするエラーは起こらない。
(D). Java の中間言語の形式はコンパイル時の機種に依存している。
(E). Java のクラスで定義したメソッドは、必ず他のクラスからも呼び出すことができる。
(F). Java のコンパイラーは、C 言語のソースファイルもコンパイルできるようになっていなければならない。

II. 次の枠内の文章は java.awt.event.KeyEvent クラスの getKeyChar メソッド、java.awt.event.InputEvent クラスの getWhen, isShiftDown メソッド、java.lang.Character クラスの isDigit メソッド、 java.lang.Systemクラスの currentTimeMillis メソッドの Java API Reference からの抜粋である。 （解くのに関係ない部分は割愛し、説明を簡単にするために改変していることがある。）

パッケージ java.awt.event

クラス KeyEvent

public class KeyEvent extends InputEvent

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

getKeyChar

public char getKeyChar()

このイベントのキーに関連付けられた文字を返します。たとえば、Shift+「a」の KEY_TYPED イベントは値「A」を返します。

戻り値:
このキー・イベントに対して定義されている Unicode 文字。 このキー・イベントに対する有効な Unicode 文字がない場合、CHAR_UNDEFINED が返される。

パッケージ java.awt.event

クラス InputEvent

public class InputEvent extends ComponentEvent

すべてのコンポーネント・レベル入力イベントのルート・イベント・クラスです。

メソッドの詳細

getWhen

public long getWhen()

このイベントが発生した時刻と協定世界時の 1970 年 1 月 1 日深夜 0 時との差をミリ秒単位で返します。

戻り値:
　　イベントが発生した時刻と協定世界時の 1970 年 1 月 1 日深夜 0 時との間のミリ秒単位の差。

isShiftDown

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。

戻り値:
　　このイベントで Shift 修飾子が押されたかどうか。

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。

パッケージ java.lang

クラス System

System クラスには有用なクラス・フィールドおよびメソッドがあります。インスタンス化することはできません。 System クラスによって得られる機能には、標準入力、標準出力、およびエラー出力ストリーム、外部的に定義されたプロパティおよび環境変数へのアクセス、ファイルおよびライブラリのローディング方法、配列の一部をすばやくコピーするユーティリティ・メソッドがあります。

メソッドの詳細

currentTimeMillis

```
public static long currentTimeMillis()
```

ミリ秒で表される現在の時刻を返します。

戻り値:
ミリ秒で測定した、現在時刻と協定世界時の 1970 年 1 月 1 日深夜 0 時との差。

下に示す KeyTestEx クラスは、これらのメソッドを使って、キー入力を待受け、入力された文字、数字が入力された場合は以前の数字の入力からの経過時間、シフトキーが押されていたかどうかの判定、を表示するプログラムである。

ファイル名 KeyTestEx.java

```
1  import java.awt.*;
2  import java.awt.event.*;
3  import javax.swing.*;
4
5  public class KeyTestEx extends JPanel implements KeyListener {
6      char c = '?';
7      long current, previous = 0;
8      boolean isDigit, isShift;
9
10     public KeyTestEx() {
11         setPreferredSize(new Dimension(360, 180));
12         previous = (1);
13         setFocusable(true);
14         addKeyListener(this);
15     }
16
17     public void keyTyped(KeyEvent e) {
18         c       = (2);
19         current = (3);
20         isDigit = (4);
21         isShift = (5);
22         repaint();
23         return;
24     }
```

```
25      public void keyPressed(KeyEvent e) {}
26      public void keyReleased(KeyEvent e) {}
27
28      @Override
29      public void paint(Graphics g) {
30          super.paint(g);
31          g.drawString("ただいま入力された文字は '" + c + "' です。",
32                       10, 50);
33          if (isDigit) {
34              g.drawString("以前の数字のキー入力からの経過時間は " +
35                  (current - previous) / 1000.0 + " 秒でした。",
36                  10, 80);
37              previous = current;
38          } else if (isShift) {
39              g.drawString("シフトキーが押されていました。", 10, 80);
40          }
41      }
42
43      /* main は省略する */
44  }
```

以下の問に答えよ。

(1) の空欄に、現在の時刻を返す式を答えよ。

(2) の空欄に、キーイベントで入力された文字を返す式を答えよ。

(3) の空欄に、キーイベントが起こった時刻を返す式を答えよ。

(4) の空欄に、文字 c が数字かどうかを表す isDigit メソッドを使う式を答えよ。

(5) の空欄に、キーイベント時にシフトキーが押されていたかどうかを表す式を答えよ。

III. 次の EmojiBtn クラスは「⌦」というボタンと 5 つの文字列のボタンと 1 つのラベルを表示し、以下のように動作する GUI アプリケーションである。

- 5 つの文字列のボタンのいずれかをクリックすれば、（ラベルの文字列が空でなかった場合は区切りの空白文字と）そのボタンに表示されている文字列を、ラベルの文字列の最後に追加する。
- 「⌦」ボタンをクリックすれば、ラベルの文字列の最初の空白までの文字列（空白を含む）を削除する。空白が含まれない場合は、ラベルの文字列を空にする。

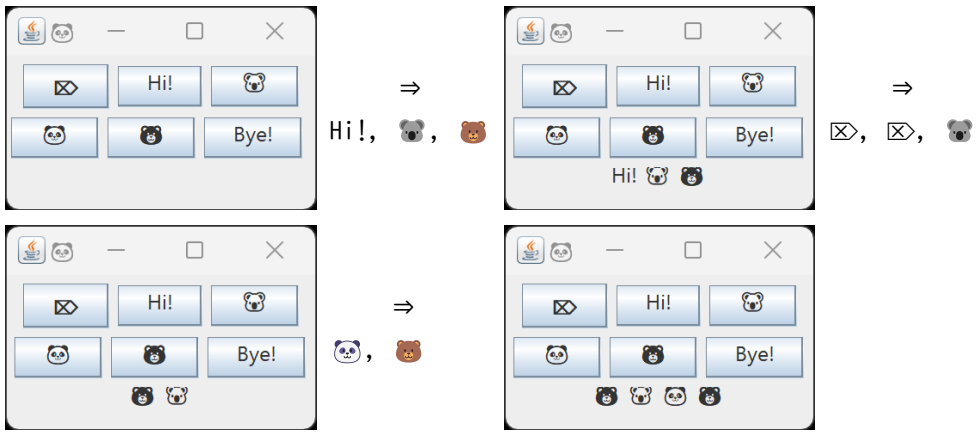
また EmojiBtn クラスは JPanel を継承している。

ファイル名 EmojiBtn.java

```
1   import java.awt.*;
2   import java.awt.event.*;
3   import javax.swing.*;
4
5   public class EmojiBtn                (1)                 {
6       private JButton del;
7       private JLabel label;
8       private String text = "";
```

```
9
10      public EmojiBtn() {
11          setPreferredSize(new Dimension(180, 90));
12          del = new JButton("⌦");
13          del.addActionListener(this);
14          add(del);
15          Font font = new Font("Segoe UI Emoji", Font.PLAIN, 12);
16          String[] strs  = { "Hi!", "🐻", "🐼", "🐻", "Bye!" };
17          for (String s: strs) {
18              JButton b = new JButton(s);
19              b.addActionListener(this);
20              b.setFont(font);
21              add(b);
22          }
23          label = new JLabel("");
24          label.setFont(font);
25          add(label);
26      }
27
28      public void actionPerformed(ActionEvent e) {
29          Object source = e.getSource();
30          if (source == del) {
31              int idx = text.indexOf(" ");
32              if (idx >= 0) {
33                  text = text.substring(idx + 1);
34              } else {
35                  text = "";
36              }
37          } else {
38              String str = ((JButton)source).getText();
39              if (text.length() == 0) {
40                  text = str;
41              } else {
42                  text += " " + str;
43              }
44          }
45          label.setText(text);
46      }
47
48      /* main は省略する */
49  }
```

実行例:



上の（1）の空欄を埋めよ。

このプログラムをラムダ式を用いて次のように書き換えるとき、(2) ～ (4) の空欄を埋めよ。なお EmojiBtn4 クラスも JPanel を継承している。(3), (4) では getSource メソッドは使用しないこと。また、EmojiBtn.java の 31 行目から 36 行目と同じ部分は△と、 39 行目から 43 行目と同じ部分は▽と省略して良い。

```
/* import は変わらないので省略する  */

public class EmojiBtn4 (2) {
    private String text = "";

    public EmojiBtn4() {
        setPreferredSize(new Dimension(180, 90));
        JButton del  = new JButton("⌦");
        JLabel label = new JLabel("");
        del.addActionListener(  (3)  );
        add(del);
        Font font = new Font("Segoe UI Emoji", Font.PLAIN, 12);
        String[] strs  = { "Hi!", "🐨", "🐼", "🐻", "Bye!" };
        for (String str: strs) {
            JButton b = new JButton(str);
            b.addActionListener(  (4)  );
            b.setFont(font);
            add(b);
        }
        label.setFont(font);
        add(label);
    }

    /* main は省略する */
}
```

IV. 次の (1) ～ (2) の Kotlin プログラムはどのように出力するか? 答えよ。なお、シーケンスの take(*n*) メソッドは先頭の *n* 個の要素からなる有限列を取り出す。

(1)

```
val foo = sequence {
    var i = 1
    while (true) {
        for (j in 1 .. i) {
            yield(Pair(j, i - j + 1))
        }
        i++
    }
}

fun main() {
    var sum = 0
    for ((x , y) in foo.take(10)) {
        sum += x * y
    }
    println("The sum is $sum.")
}
```

(2)

```
val bar = sequence {
    var a = 3;
    var b = 0;
    var c = 2;
    while (true) {
        yield(a)
        val d = a + b
        a = b
        b = c
        c = d
    }
}

fun main() {
    for (i in bar.take(20)) {
        print("$i, ")
    }
    println()
}
```

V. 次の（1）～（2）の多肢選択問題に答えよ。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）abc.Foo クラス（abc パッケージの Foo クラス），abc.Bar クラス（abc パッケージの Bar クラス），xyz.Baz クラス（xyz パッケージの Baz クラス）をそれぞれ次のように定義する

パス：abc/Foo.java

```
package abc;

public class Foo {
  public int x;
  int y;
  private int z;

  public Foo(int a, int b, int c) {
    x = a; y = b; z = c;
  }
}
```

パス：abc/Bar.java

```
package abc;
import xyz.Baz;

public class Bar {
  public static void main(String[] args) {
    Foo foo = new Foo(1, 2, 3);
    Baz baz = new Baz(4, 5, 6);
    System.out.println(            );
  }
}
```

パス：xyz/Baz.java

```java
package xyz;

public class Baz {
  public int x;
  int y;
  private int z;

  public Baz(int a, int b, int c) {
    x = a; y = b; z = c;
  }
}
```

abc/Bar.java の 8 行目の空欄に入れてエラーにならない式は次のうちどれか？すべて選べ。

(A). foo.x (B). foo.y (C). foo.z (D). baz.x (E). baz.y
(F). baz.z

(2) 次の選択肢の中で、一般的な処理系で実行前にコンパイルして型チェックを行う言語はどれか？すべて選べ。

(A). Ruby (B). Python (C). C (D). JavaScript (E). Java

VI. 育成ゲーム用のクラス階層を設計するために、テスト用のいくつかのクラスを作成する。次に定義されるクラス Creature を継承して、

ファイル名 Creature.java

```java
public class Creature {
    protected int age;

    public Creature() {
        age = 0;
    }

    public void grow() {
        age++;
    }

    public String getName() {
        return "something";
    }

    public int getNumLegs() {
        return 8;
    }

    public void selfIntroduce() {
        System.out.printf(
            "I am a %d week old %s. I have %d legs.%n",
            age, getName(), getNumLegs());
    }
}
```

3 つのサブクラス Frog, Earthworm, Butterfly を定義する。

ファイル名 Frog.java

```java
public class Frog extends Creature {
    public Frog() {
        super();
    }

    @Override
    public String getName() {
        if (age <= 3) {
            return "tadpole";      // オタマジャクシ
        }
        return "frog";             // カエル
    }

    @Override
    public int getNumLegs() {
        if (age <= 1) {
            return 0;
        } else if (age <= 3) {
            return 2;
        }
        return 4;
    }
}
```

ファイル名 Earthworm.java

```java
public class Earthworm extends Creature {
    public Earthworm() {
        super();
    }

    @Override
    public String getName() {
        return "earthworm";        // ミミズ
    }

    @Override
    public int getNumLegs() {
        return 0;
    }
}
```

ファイル名 Butterfly.java

```java
public class Butterfly extends Creature {
    public Butterfly() {
        super();
    }

    @Override
    public String getName() {
        if (age <= 2) {
            return "caterpillar"; // イモムシ
        } else if (age <= 4){
            return "pupa";         // サナギ
        }
        return "butterfly";        // チョウ
    }
```

```
16
17      @Override
18      public int getNumLegs() {
19          if (age <= 2) {
20              return 16;
21          } else if (age <= 4) {
22              return 0;
23          }
24          return 6;
25      }
26  }
```

さらに main メソッドを持つクラス CreatureTest を次のように定義する。

```
1   public class CreatureTest {
2       public static void main(String[] args) {
3           int i, j;
4           Creature[] creatures = new Creature [] {
5               new Butterfly(), new Frog(), new Earthworm()
6           };
7
8           for (i = 0; i < 4; i++) {
9               for (Creature c: creatures) {
10                  c.selfIntroduce();
11                  c.grow();
12                  c.grow();
13              }
14          }
15      }
16  }
```

CreatureTest 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。直前の行と同じ内容は「〃」と省略して良い。

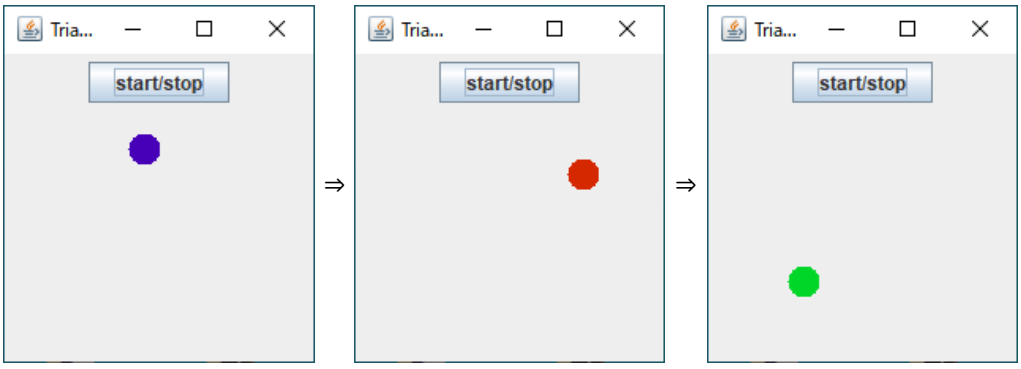
VII. 次の Triangle は点の動きのアニメーションを表示する GUI アプリケーションである。“start/stop” ボタンをクリックするとアニメーションを停止し、もう一度“start/stop” ボタンをクリックするとアニメーションを再開する。 Triangle クラスは JPanel を継承している。

ファイル名 Triangle.java

```
1   import javax.swing.*;
2   import java.awt.*;
3
4   public class Triangle extends JPanel implements Runnable {
5       int i = 0, x = 50, y = 0, z = 0, dx = -1, dy = 1, dz = 0;
6       Thread thread = null;
7
8       public Triangle() {
9           setPreferredSize(new Dimension(195, 195));
10          JButton btn = new JButton("start/stop");
11          btn.addActionListener(e -> toggleThread());
12          add(btn);
13          toggleThread();
14      }
```

```
15
16      public void toggleThread() {
17          if (thread == null) {
18              thread = [          (1)          ];
19              thread.start();
20          } else {
21              thread = [          (2)          ];
22          }
23      }
24
25      @Override
26      public void run() {
27          Thread me = [          (3)          ];
28          while (thread == me) {
29              x += dx;    y += dy;    z += dz;
30              repaint();
31              try {
32                  Thread.sleep(40);
33              } catch (InterruptedException e) {
34              }
35
36              if (++i >= 50) {
37                  i = 0;
38                  int tmp = dz; dz = dy; dy = dx; dx = tmp;
39              }
40          }
41      }
42
43      @Override
44      public void paintComponent(Graphics g) {
45          super.paintComponent(g);
46          g.setColor(new Color((int)(x * 255.0 / 50.0),
47                               (int)(y * 255.0 / 50.0),
48                               (int)(z * 255.0 / 50.0)));
49          g.fillOval(50 + (int)(x * 2), 50 + (int)(y * 2), 20, 20);
50      }
51      /* main は省略する */
52  }
```

実行例:



(1) ～ (3) の空欄を埋めてプログラムを完成せよ。

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

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) {
            x -= 10;
        } else if (source == rBtn) {
            x += 10;
        }
        repaint();
    }

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

ファイル LeftRightButton4.java

```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);
    }
```

```
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  }
```

ファイル Guruguru.java

```
1   import java.awt.*;
2   import javax.swing.*;
3
4   public class Guruguru extends JPanel implements Runnable {
5       private int r = 50;
6       private volatile int x = 110, y = 70 ;
7       private double theta = 0; // 角度
8       private volatile Thread thread = null;
9
10      public Guruguru() {
11          setPreferredSize(new Dimension(200, 180));
12          JButton startBtn = new JButton("start");
13          startBtn.addActionListener(e -> startThread());
14          JButton stopBtn = new JButton("stop");
15          stopBtn.addActionListener(e -> stopThread());
16          setLayout(new FlowLayout());
17          add(startBtn); add(stopBtn);
18          startThread();
19      }
20
21      private void startThread() {
22          if (thread == null) {
23              thread = new Thread(this);
24              thread.start();
25          }
26      }
27
28      private void stopThread() {
29          thread = null;
30      }
31
32      @Override
33      public void paintComponent(Graphics g) {
34          super.paintComponent(g); // スーパークラスの paintComponent を呼び出す
35          // 全体を背景色で塗りつぶす。
36          g.drawString("Hello, World!", x, y);
37      }
38
39      public void run() {
40          Thread thisThread = Thread.currentThread();
41          for (; thread == thisThread; theta += 0.02) {
42              x =  60 + (int)(r * Math.cos(theta));
43              y = 100 - (int)(r * Math.sin(theta));
44              repaint();  // paintComponent を間接的に呼出す
45              try {
46                  Thread.sleep(30); // 30 ミリ秒お休み
47              } catch (InterruptedException e) {}
48          }
49      }
50
51      public static void main(String[] args) {/* 省略 */}
52  }
```

ファイル Point.java

```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

```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;
    }
}
```

ファイル Quadratic.kt

```kotlin
import kotlin.math.*

fun quadratic(a: Double, b: Double, c: Double): Pair<Double, Double> {
    val d = b * b - 4 * a * c
    val sq = sqrt(d)
    return Pair((-b + sq) / (2 * a), (-b - sq) / (2 * a))
}

fun main() {
    val a = 1.0; val b = -1.0; val c = -1.0
    val (x1, x2) = quadratic(a, b, c)
    println("方程式の解は、$x1、$x2 です。")
    // 出力 ⇒ 方程式の解は、1.618033988749895、-0.6180339887498949 です。
}
```

ファイル SequenceTest.kt

```kotlin
val nums = generateSequence(1) { x -> x + 3 }

val fibs = sequence {
    var a = 1; var b = 1
    while (true) {
        yield(a)
        val c = a
        a = b; b += c
    }
}

fun main() {
    println(nums.take(10).toList())
    // 出力 ⇒ [1, 4, 7, 10, 13, 16, 19, 22, 25, 28]
    println(fibs.take(10).toList())
    // 出力 ⇒ [1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
}
```

（計算用紙）

（計算用紙）

オブジェクト指向言語・期末テスト解答用紙（2023 年 07 月 27 日）

学籍番号		氏名	

I. (3 × 2)

(1)		(2)	

II. (3 × 5)

(1)	
(2)	
(3)	
(4)	
(5)	

III. (1, 1, 3, 4)

(1)	
(2)	
(3)	
(4)	

IV. (4 × 2)

(1)	
(2)	

V. (4 × 2)

(1)		(2)	

VI. (12)

I am a week old . I have legs.
〃 〃 . 〃 〃 .

VII. (4 × 3)

(1)	
(2)	
(3)	

授業・テストの感想