オブジェクト指向言語・期末テスト問題用紙
（2022年 07月 28日・ 08:50 ～ 10:20）

解答上、その他の注意事項

1. 問題は、問 Ⅰ～Ⅶ まである。うち、問 Ⅰ～Ⅲ は中間テストの代替問題である。

   ○ 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, LeftRightButton2.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（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）Java のクラス名として文法上使用可能な識別子は次のうち、どれか? すべて選べ。

(A). `Covid19` (B). `X_Y_Z_` (C). `ABC-MART`
(D). `2X4` (E). `for_a_while` (F). `QQQQQ`

（ii）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。
(B). Java のクラスが実装（implement）できるインタフェースは多くて 1 つである。
(C). Java の public なクラスは、その名前に .java という拡張子をつけたファイルに定義する必要がある。
(D). Java は JavaScript に型宣言を仕様として追加したプログラミング言語である。
(E). Java のクラスでは、他のクラスのコードからは参照できないメソッドを定義することができる。
(F). Java の配列は範囲外をアクセスしても、C と同じでエラーや例外にはならない。

II. 次の枠内の文章は `org.jsoup.Jsoup` クラスの `parse` メソッド、`org.jsoup.nodes.Documents` クラス、 `org.jsoup.nodes.Element` クラスの `select`, `after`, `before`, `firstElementChild`, `empty` メソッド、`org.jsoup.select.Elements`クラスの jsoup API Reference 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。）

Package `org.jsoup`

Class **Jsoup**

`public class Jsoup extends Object`

Method Details

parse

`public static Document parse(File file) IOException`

Parse the contents of a file as HTML. …

Parameters:
`file` - the file to load HTML from.
Returns:
sane HTML
Throws:
`IOException` - if the file could not be found or read.

---

sane – 健全な、まっとうな

Package `org.jsoup.nodes`

Class **Document**

`public class Document extends Element`

An HTML Document.

Package `org.jsoup.nodes`

Class **Element**

```
public class Element extends Node
```

An HTML element consists of a tag name, attributes, and child nodes (including text nodes and other elements). From an Element, you can extract data, traverse the node graph, and manipulate the HTML.

Method Details

select

```
public Elements select(String cssQuery)
```

Find elements that match the Selector CSS query, with this element as the starting context. Matched elements may include this element, or any of its children. …. Also known as `querySelectorAll()` in the Web DOM.

Parameters:
`cssQuery` - a Selector CSS-like query
Returns:
an Elements list containing elements that match the query (empty if none match)

after

```
public Element after(String html)
```

Insert the specified HTML into the DOM after this element (as a following sibling).

Parameters:
`html` - HTML to add after this element
Returns:
this element, for chaining

firstElementChild

```
public Element firstElementChild()
```

Gets the first child of this Element that is an Element, or null if there is none.

Returns:
the first Element child node, or null.

before

```
public Element before(String html)
```

Insert the specified HTML into the DOM before this element (as a preceding sibling).

Parameters:
`html` - HTML to add before this element
Returns:
this element, for chaining

empty

public Element empty()

Remove all of the element's child nodes. Any attributes are left as-is.

Returns:
this element

Package org.jsoup.select

Class **Elements**

public class Elements extends ArrayList<Element>

A list of Elements, with methods that act on every element in the list.

下に示す JsoupTest クラスは、これらのメソッドを使って、HTML ファイルを読み込み、<h1> タグの前に <hr> を挿入し、 タグの最初の子要素の後に ???? という要素を挿入し、 <address> タグの子ノードをすべて除去した HTML を別のファイルに保存するプログラムである。つまり、次のような HTML ファイルを

```
<!doctype html>
<html>
 <head>
  <title>Test Document</title>
 </head>
 <body>
  <h1>Componies</h1>
  <ul>
   <li>Oracle</li>
   <li>JetBrains</li>
   <li>Mozilla</li>
  </ul>
  <address>Taro Yamada</address>
 </body>
</html>
```

次のような HTML ファイルに書き換える。（変わっているのは、7, 11, 15 行目、ただし空白や改行を適宜挿入／削除して整形している。）

```
<!doctype html>
<html>
 <head>
  <title>Test Document</title>
 </head>
 <body>
  <hr>
  <h1>Componies</h1>
  <ul>
   <li>Oracle</li>
   <li>????</li>
   <li>JetBrains</li>
   <li>Mozilla</li>
  </ul>
  <address></address>
 </body>
</html>
```

ファイル名 JsoupTest.java

```java
import org.jsoup.Jsoup;
import org.jsoup.nodes.*;
import org.jsoup.select.*;

import java.io.*;

public class JsoupTest {
    public static void main(String[] args) throws IOException {
        File f = new File(args[0]);
        Document doc = (ⅰ);
        for (Element elem: doc.select("h1")) {
            (ⅱ);
        }
        for (Element elem: doc.select("ul")) {
            (ⅲ);
        }
        for (Element elem: doc.select("address")) {
            (ⅳ);
        }

        String tmpdir = System.getProperty("java.io.tmpdir");
        String outf   = "tmp.html";
        // 変換したものをファイルに保存
        File g = new File(new File(tmpdir), outf);
        try (FileWriter out = new FileWriter(g)) {
            out.write(doc.outerHtml());
            System.err.printf("output written to %s.%n",
                              g.getCanonicalPath());
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

以下の問に答えよ。

（ⅰ）には、ファイル f の内容を構文解析して Document を返す式が入る。この式を答えよ。

（ⅱ）には、要素 elem の前に <hr> という HTML を挿入する式が入る。この式を答えよ。

（ⅲ）には、要素 elem の最初の子要素の後に ???? という HTML を挿入する式が入る。この式を答えよ。

（ⅳ）には、要素 elem の子ノードをすべて除去する式が入る。この式を答えよ。

III. 次の ThreePoles クラスは 要素数 3 の配列 data のグラフを表示し、以下のように動作する GUI アプリケーションである。

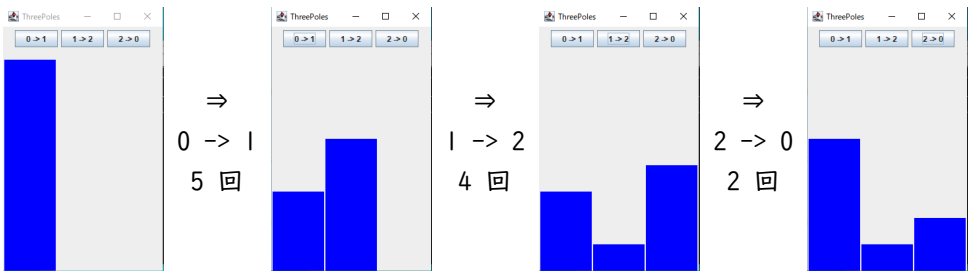
- 「0 -> 1」ボタンをクリックすれば、 data[0] が正の数ならばデクリメントして、data[1] をインクリメントする。
- 「1 -> 2」ボタンをクリックすれば、 data[1] が正の数ならばデクリメントして、data[2] をインクリメントする。
- 「2 -> 0」ボタンをクリックすれば、 data[2] が正の数ならばデクリメントして、data[0] をインクリメントする。

ファイル名 ThreePoles.java

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

public class ThreePoles extends JPanel implements ActionListener {
    private final int wid = 78, hei = 40, gap = 2;
```

```
    private final int max = 8, pad = 50;
    private int data[] = { max, 0, 0 };
    private JButton button1, button2, button3;

    public ThreePoles() {
        setPreferredSize(new Dimension(gap + 3 * (wid + gap),
                                       pad + max * hei));
        button1 = new JButton("0 -> 1");
                          ( i )                          ;
        button2 = new JButton("1 -> 2");
                          ( ii )                         ;
        button3 = new JButton("2 -> 0");
                          ( iii )                        ;
        add(button1); add(button2); add(button3);
    }

    public void actionPerformed(ActionEvent e) {
        Object target = e.getSource();
        int src, dst;
        if (target == button1) {
            src = 0; dst = 1;
        } else if (target == button2) {
            src = 1; dst = 2;
        } else {
            src = 2; dst = 0;
        }
        if (data[src] > 0) {
            data[src]--; data[dst]++;
        }
        repaint();
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        for (int i = 0; i < 3; i++) {
            int d = data[i];
            g.setColor(Color.BLUE);
            g.fillRect(gap + i * (gap + wid),
                       pad + (max - d) * hei, wid, d * hei);
        }
    }

    /* main は省略する */
}
```

実行例:



上の（ i ），（ ii ），（ iii ）の空欄を埋めよ。

このプログラムを内部クラスを用いて次のように書き換えるとき、（iv）～（viii）の空欄を埋めよ。

```
/* import は変わらないので省略する  */
public class ThreePoles2                ( iv )                 {
```

```
    private final int wid = 78, hei = 40, gap = 2;
    private final int max = 8, pad = 50;
    private int data[] = { max, 0, 0 };

    class ButtonListener implements ActionListener {
        private int src, dst;
        ButtonListener(int s, int d) {
                          (v)
        }

        public void actionPerformed(ActionEvent e) {
            if (data[src] > 0) {
                data[src]--; data[dst]++;
            }
            repaint();
        }
    }

    public ThreePoles2() {
        setPreferredSize(new Dimension(gap + 3 * (wid + gap),
                                       pad + max * hei));
        JButton button1 = new JButton("0 -> 1");
                                  (vi)                              ;
        JButton button2 = new JButton("1 -> 2");
                                  (vii)                             ;
        JButton button3 = new JButton("2 -> 0");
                                  (viii)                            ;
        add(button1); add(button2); add(button3);
    }

    /* paintComponent は変わらないので省略する */
    /* main は省略する */
}
```

IV. 次の（ｉ）～（ii）の Kotlin プログラムはどのように出力するか？答えよ。なお、`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し、 `toList()` メソッドは、出力するためにシーケンスをリストに変換する。

（ｉ）

```
fun foo(n: Int): Pair<Int, Int> {
    if (n <= 3) return Pair(n, 2);
    val (m, k) = foo(n - 3)
    return Pair(m * k, k + 1)
}

fun main() {
    val (a, b) = foo(11)
    println(a)
}
```

（ii）

```
fun bar(m: Int) = sequence {
    var n = m
    while (true) {
        yield(n)
        val k = n % 3
        if (k == 0) {
            n /= 3
        } else if (k == 1) {
            n = 2 * n + 1
        } else {
            n *= 2
```

```
        }
    }
}

fun main() {
    println(bar(11).take(6).toList())
}
```

V. 次の（ｉ）～（ii）の多肢選択問題に答えよ。

（ｉ）次の選択肢の中で論理型に分類される言語はどれか？もっともふさわしいものを一つ選べ。

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

（ii）次の選択肢の中でスクリプト言語と呼ぶのに、もっともふさわしくない言語はどれか？一つ選べ。

(A). PHP (B). Python (C). C++ (D). Perl (E). Ruby

VI. グラフィックスライブラリー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 `Turtle` を継承して、

ファイル名 Turtle.java

```java
public class Turtle {
    protected double direction;
    protected double x, y;

    public Turtle() {
        direction = 0;
        x = y = 0;
    }

    public void show() {
        // %.2f は小数点の後の桁数が 2 という意味
        System.out.printf(
          "position: (%.2f, %.2f), direction: %.2f%n",
          x, y, direction);
    }

    public void move(double distance) {
        x += distance * Math.cos(Math.toRadians(direction));
        y += distance * Math.sin(Math.toRadians(direction));
        show();
    }

    public void turn(double deg) {
        direction += deg;
    }
}
```

2 つのサブクラス `LimitedTurtle`, `PerverseTurtle` を定義する。

ファイル名 LimitedTurtle.java

```java
public class LimitedTurtle extends Turtle {
    protected int limit;
    public LimitedTurtle(int life) {
        super();
        limit = life;
    }
```

```
    @Override
    public void move(double distance) {
        if (limit > 0) {
            limit--;
            super.move(distance);
        } else {
            super.move(0);
        }
    }
}
```

ファイル名 PerverseTurtle.java

```
// perverse - へそ曲がりな
public class PerverseTurtle extends Turtle {
    public PerverseTurtle() {
        super();
    }

    @Override
    public void turn(double deg) {
        direction -= deg;
    }
}
```

ファイル名 TurtleTest.java

```
public class TurtleTest {
    public static void main(String[] args) {
        Turtle[] turtles = new Turtle[] {
             new Turtle(),
             new LimitedTurtle(2),
             new PerverseTurtle()
        };

        for (Turtle t: turtles) {
            t.move(10);
            t.turn(90);
            t.move(20);
            t.turn(-90);
            t.move(10);
        }
    }
}
```

(ⅰ). Turtle の 3 つのフィールド direction, x, y と LimitedTurtle のフィールド limit は protected と宣言されているが、これを private に変更しても（ここに掲載された 4 つのクラスをコンパイル・実行するとき）エラーにならないものを次からすべて選べ。

(A). direction (B). x (C). y (D). limit

(ⅱ). TurtleTest 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。

(ア) Math.cos(Math.toRadians(*d*))、(イ) Math.cos(Math.toRadians(*d*)) は次の表の値を使用せよ。（N は任意の整数とする。）

式＼*d*	0 + 360N	90 + 360N	180 + 360N	270 + 360N
(ア)	1.00	0.00	-1.00	0.00
(イ)	0.00	1.00	0.00	-1.00

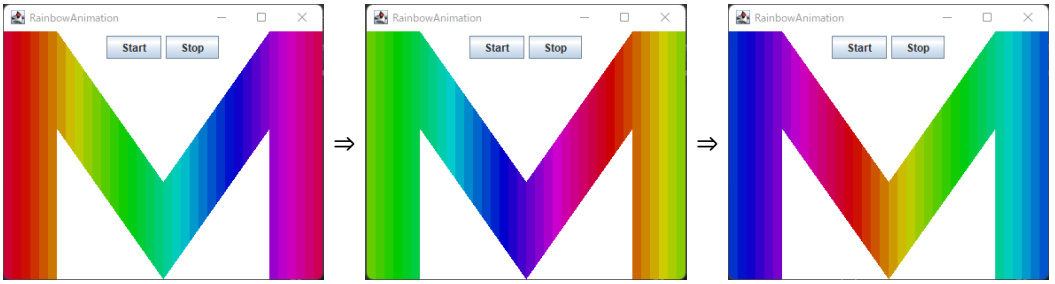
VII. 次の RainbowAnimation はグラデーションの色相のアニメーションを表示する GUI アプリケーションである。“Stop” ボタンをクリックするとアニメーションを停止し、“Start” ボタンをクリックするとアニメーションを再開する。

ファイル名 RainbowAnimation.java

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

public class RainbowAnimation extends JPanel implements Runnable {
    private final int width = 360, height = 280;
    private Shape shape;
    private volatile int init = 30;
    private volatile Thread thread = null;

    public RainbowAnimation() {
        setBackground(Color.WHITE);
        setPreferredSize(new Dimension(width, height));

        int[] xpoints = {  0,   0,  60,  60, 180, 300, 300,
                         360, 360, 300, 180,  60,   0};
        int[] ypoints = {  0, 280, 280, 110, 280, 110, 280,
                         280,   0,   0, 170,   0,   0};
        shape = new Polygon(xpoints, ypoints, xpoints.length);

        JButton startButton = new JButton("Start");
        startButton.addActionListener(ev -> startThread());
        JButton stopButton  = new JButton("Stop");
        stopButton.addActionListener(ev -> stopThread());
        add(startButton); add(stopButton);

        startThread();
    }

    private void startThread() {
        if (thread == null) {
            thread = new Thread(this);
            thread.start();
        }
    }

    private void stopThread() {
        (ⅰ);
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setClip(shape);
        for (int deg = 0; deg < 360; deg += 10) {
            int t = (init + deg) % 360;
            Color c = Color.getHSBColor(t / 360f, 1f, 0.8f);
            g.setColor(c);
            g.fillRect(deg, 0, 10, height);
        }
        g.setClip(null);
    }

    public void run() {
        Thread current = Thread.currentThread();
        while ((ⅱ)) {
            init += 15;
            (ⅲ);
            try {
                Thread.sleep(100);
```

```
60                } catch (InterruptedException ex) {}
61            }
62        }
63        /* main は省略する */
64    }
```

実行例:



空欄（ｉ）～（iii）を埋めてプログラムを完成せよ。（空欄はすべて文法上、式である。）念のため参考に `setClip` メソッドと `Polygon` クラスの API Documentation を以下に示す。

Package java.awt

Class **Graphics**

Method Details

setClip

```
public abstract void setClip(Shape clip)
```

Sets the current clipping area to an arbitrary clip shape.

Parameters:
clip - the Shape to use to set the clip. Passing null clears the current clip.

Package java.awt

Class **Polygon**

```
public class Polygon extends Object
                     implements Shape, Serializable
```

Constructor Details

Polygon

```
public Polygon(int[] xpoints,
               int[] ypoints,
               int npoints)
```

Constructs and initializes a Polygon from the specified parameters.

Parameters:
xpoints - an array of X coordinates
ypoints - an array of Y coordinates
npoints - the total number of points in the Polygon

以下に参考のために授業配布プリントの LeftRightButton.java, LeftRightButton2.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) {      // 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 LeftRightButton2.java

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

public class LeftRightButton2 extends JPanel {
    private int x = 20;

    public LeftRightButton2() {
        setPreferredSize(new Dimension(200, 70));
        JButton lBtn = new JButton("Left");
        JButton rBtn = new JButton("Right");
        lBtn.addActionListener(new LeftListener());
        rBtn.addActionListener(new RightListener());
        setLayout(new FlowLayout());
        add(lBtn); add(rBtn);
    }

    private class LeftListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            x -= 10;
            repaint();
        }
    }

    private class RightListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            x += 10;
```

```
            repaint();
        }
    }

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

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

ファイル Guruguru.java

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

public class Guruguru extends JPanel implements Runnable {
    private int r = 50;
    private volatile int x = 110, y = 70 ;
    private double theta = 0; // 角度
    private volatile Thread thread = null;

    public Guruguru() {
        setPreferredSize(new Dimension(200, 180));
        JButton startBtn = new JButton("start");
        startBtn.addActionListener(e -> startThread());
        JButton stopBtn = new JButton("stop");
        stopBtn.addActionListener(e -> stopThread());
        setLayout(new FlowLayout());
        add(startBtn); add(stopBtn);
        startThread();
    }

    private void startThread() {
        if (thread == null) {
            thread = new Thread(this);
            thread.start();
        }
    }

    private void stopThread() {
        thread = null;
    }

    @Override
    public void paintComponent(Graphics g) {
        super.paintComponent(g); // スーパークラスの paintComponent を呼び出す
        // 全体を背景色で塗りつぶす。
        g.drawString("Hello, World!", x, y);
    }

    public void run() {
        Thread thisThread = Thread.currentThread();
        for (; thread == thisThread; theta += 0.02) {
            x =  60 + (int)(r * Math.cos(theta));
            y = 100 - (int)(r * Math.sin(theta));
            repaint();  // paintComponent を間接的に呼出す
            try {
                Thread.sleep(30); // 30 ミリ秒お休み
            } catch (InterruptedException e) {}
        }
    }

    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;
    }
}
```

ファイル Quadratic.kt

```
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

```
fun main() {
    val nums = generateSequence(1) { x -> x + 3 }
    println(nums.take(10).toList())
    // 出力 ⇒ [1, 4, 7, 10, 13, 16, 19, 22, 25, 28]
    val fibs = sequence {
        var a = 1; var b = 1
        while (true) {
            yield(a)
            val c = a
            a = b; b += c
        }
    }
    println(fibs.take(10).toList())
    // 出力 ⇒ [1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
}
```

（計算用紙）

オブジェクト指向言語・期末テスト解答用紙（2022 年 07 月 28 日）

学籍番号		氏名	

I. (3 × 2)

(i)		(ii)	

II. (3 × 4)

(i)	
(ii)	
(iii)	
(iv)	

III. (1, 1, 1, 3, 3, 1, 1, 1)

(i)	
(ii)	
(iii)	
(iv)	
(v)	
(vi)	
(vii)	
(viii)	

IV. (4 × 2)

(i)	
(ii)	

V. (4 × 2)

(i)		(ii)	

VI. (4, 8)

(i)	
(ii)	position:(,), direction: 〃 :(,), 〃 :

VII. (4 × 3)

(i)	
(ii)	
(iii)	

授業・テストの感想