

オブジェクト指向言語・中間テスト問題用紙 (2026年 06月 11日・ 09:35 ～ 10:20)

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

参考のために問題用紙の末尾に授業配布プリントの `MouseTest.java`, `LeftRightButton3.java`, `LeftRightButton4.java`, と `MouseAdapter` クラスのソース (の抜粋) を掲載する。

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 `xs` を `double` 型を保存するための、空の `ArrayList` に初期化するための宣言として正しいものはどれか? 一つ選べ。

- (A). `Double<ArrayList> xs = new <ArrayList>()`
- (B). `ArrayList<Double> xs = new ArrayList<>()`
- (C). `double<ArrayList> xs = new <ArrayList>()`
- (D). `ArrayList<double> xs = new ArrayList<>()`
- (E). `double[ArrayList] xs = new [ArrayList]()`
- (F). `ArrayList[double] xs = new ArrayList[]()`

(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、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である。
- (B). Java からクラスなどの機構を除き初心者向けに設計しなおした言語が C である。
- (C). Java は、型を実行時にはじめてチェックする動的な型付け言語である。
- (D). HTML に埋め込まれ、ブラウザで動く Java のサブセットを JavaScript という。

II. 次の枠内の文章は `java.awt.Point` クラス、`java.awt.Shape` インタフェースの `contains` メソッド、`java.awt.geom.Ellipse2D.Double` クラスのコンストラクターといくつかのフィールド、`java.awt.event.MouseEvent` クラスのいくつかのフィールドとメソッド、`java.awt.Graphics` クラスの `setClip` メソッドの Java™ API 仕様からの抜粋である。(ただし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。また、問題を解くのに関係ない継承 (`extends`) やインタフェースの実装 (`implements`) も省略している。)

パッケージ `java.awt`

クラス `Point`

```
public class Point
```

整数精度で指定された、 (x,y) 座標空間内の位置を表す点を表します。

パッケージ `java.awt`

インタフェース `Shape`

```
public interface Shape
```

`Shape`インタフェースは、何らかの形式の幾何学的な形状を表すオブジェクトの定義を提供します。

メソッドの詳細

contains

```
public boolean contains(Point p)
```

指定されたPointが、内側の定義によって記述されたShapeの境界の内側にあるかどうかをテストします。

パラメータ:

p - テストされる指定されたPoint

戻り値:

指定されたPointがShapeの境界の内側にある場合はtrue、それ以外の場合はfalse。

パッケージ java.awt.geom

クラス **Ellipse2D.Double**

```
public static class Ellipse2D.Double implements Shape
```

Ellipse2D.Double クラスは、double精度で指定された楕円を定義します。

注: Ellipse2D.Double は正確には Ellipse2D クラスの Double という名前の static な内部クラスだが、この間では Ellipse2D.Double という名前のクラスと考えても差し支えない。

フィールドの詳細

x

```
public double x
```

このEllipse2D.Doubleの表示枠矩形の左上隅のX座標です。

y

```
public double y
```

このEllipse2D.Doubleの表示枠矩形の左上隅のY座標です。

w

```
public double w
```

このEllipse2D.Doubleの全体の幅です。

h

```
public double h
```

このEllipse2D.Doubleの全体の高さです。

コンストラクターの詳細

Ellipse2D.Double

```
public Ellipse2D.Double(double x, double y, double w, double h)
```

指定された座標からEllipse2D.Doubleを構築および初期化します。

パラメータ:

- x - 表示枠矩形の左上隅のX座標
- y - 表示枠矩形の左上隅のY座標
- w - 表示枠矩形の幅
- h - 表示枠矩形の高さ

パッケージ java.awt.event

クラス MouseEvent

```
public class MouseEvent
```

コンポーネント内でマウス・アクションが発生したことを示すイベントです。

フィールドの詳細

BUTTON1

```
public static final int BUTTON1
```

マウス・ボタン1を示します。getButton() によって使用されます。

BUTTON2

```
public static final int BUTTON2
```

マウス・ボタン2を示します。getButton() によって使用されます。

BUTTON3

```
public static final int BUTTON3
```

マウス・ボタン3を示します。getButton() によって使用されます。

メソッドの詳細

getPoint

```
public Point getPoint()
```

イベントの座標を表す Point オブジェクトを返します。

戻り値:

x座標およびy座標を含むPointオブジェクト。

getClickCount

```
public int getClickCount()
```

このイベントのクリック回数を返します。

戻り値:

クリック数を表す整数値。

getButton

```
public int getButton()
```

状態が変更されたマウス・ボタンがある場合、そのイベントのマウス・ボタンを返します。戻り値は、次の定数のいずれかの可能性があります。

- BUTTON1 – 最初のマウス・ボタン（通常は左ボタン）
- BUTTON2 – 2番目のマウス・ボタン（通常はホイール・ボタン）
- BUTTON3 – 3番目のマウス・ボタン（通常は右ボタン）

これらの定数を使用して、返されたボタン番号と比較できます。

getX

```
public int getX()
```

イベントのX座標を返します。

戻り値:

X座標を表す整数値。

getY

```
public int getY()
```

イベントのY座標を返します。

戻り値:

Y座標を表す整数値。

パッケージ java.awt

クラス Graphics

```
public class Graphics
```

Graphicsクラスは、アプリケーションがさまざまなデバイス上に実現されたコンポーネントやオンスクリーン・イメージ上に描画できるようにする、すべてのグラフィックス・コンテキストのための抽象基底クラスです。

メソッドの詳細

setClip

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

与えられた Shape を現在のクリッピング領域として設定します。

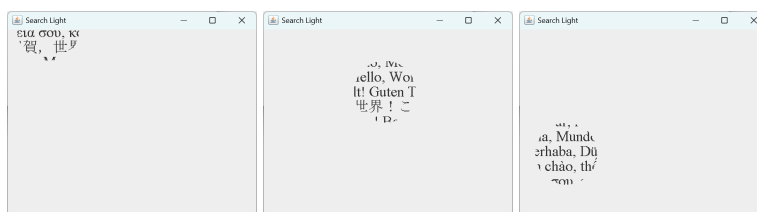
パラメータ:

clip - クリップ設定に使用するShape。 nullを渡すと、現在のclipがクリアされます。

SearchLight クラスは、タイマーを使用してクリップする円が画面上をサーチライトのように動くアニメーションを表示する。さらに、マウス・イベントを処理して、円の大きさを変えたり、円の位置を変えたりする。

- クリックが円内で発生した場合、円の大きさを変更する。
 - 左クリックなら円の大きさを大きくする。
 - 右クリックなら円の大きさを小さくする。
 - ただしダブルクリックなら（ボタンに関わらず）円の大きさを初期値に戻す。
- クリックが円外で発生した場合、円の位置を変更する。
 - ダブルクリックなら、クリックした位置に円の中心が来るように円を移動する。
 - そうでないなら、クリックした位置が円の中心より左なら円を左に、そうでないなら右に移動する。

次のスクリーンショットは、マウスをクリックしないときのアニメーションの様子である。



開始すぐ

しばらく後

さらに後

SearchLight 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 SearchLight.java

```
1 import java.awt.*;
2 import javax.swing.*;
3 import java.awt.event.*;
4 import java.awt.geom.*;
5
6 class SearchLight extends JPanel
7     implements ActionListener, MouseListener {
8     private double dx = 5;
9     private int width = 400, height = 300;
10    private double r0 = 50, r = r0, dr = 5;
11    private Ellipse2D.Double circ =
12        new Ellipse2D.Double(-r, -r, 2 * r, 2 * r);
13
14    public SearchLight() {
15        setPreferredSize(new Dimension(width, height));
16        addMouseListener(this);
17        Timer timer = new Timer(50, this);
18        timer.start();
19    }
20
21    public void actionPerformed(ActionEvent e) {
22        circ.x += dx;
23        if (circ.x < -r && dx < 0
24            || circ.x + r > width && dx > 0) {
25            dx = -dx; circ.y += r;
26            if (circ.y > height - r) {
27                circ.y = -r;
28            }
29        }
30        repaint();
31    }
```

```

32
33 public void mouseClicked(MouseEvent e) {
34     if ( (1) ) {
35         if ( (2) ) {
36             circ.width = circ.height = 2 * r0;
37         } else if ( (3) ) {
38             circ.width -= 2 * dr;
39             circ.height -= 2 * dr;
40         } else {
41             circ.width += 2 * dr;
42             circ.height += 2 * dr;
43         }
44     } else if ( (2) ) {
45         circ.x = (4-1) - r;
46         circ.y = (4-2) - r;
47     } else {
48         if ( (4-1) < circ.x + r ) {
49             circ.x -= 2 * r;
50         } else {
51             circ.x += 2 * r;
52         }
53     }
54     repaint();
55 }
56 public void mousePressed(MouseEvent e) {}
57 public void mouseReleased(MouseEvent e) {}
58 public void mouseEntered(MouseEvent e) {}
59 public void mouseExited(MouseEvent e) {}
60
61 @Override
62 protected void paintComponent(Graphics g) {
63     super.paintComponent(g);
64     (5);
65     g.setFont(new Font("Serif", Font.PLAIN, 24));
66     int x = 0, y = -12;
67     g.drawString("Γεια σου, κόσμε! Γεια σου, κόσμε!",
68                 x, y += 24);
69     // 以下の drawString は掲載省略
70 }
71 }
72 /* main は省略する */

```

(1) にはマウスをクリックした点が円 `circ` の内部かどうかを判定する式が入る。この式を答えよ。

(2) には、マウスクリックがダブルクリックか（クリック回数が2以上か）を判定する式が入る（2箇所共通）。この式を答えよ。

(3) には、マウスクリックが右クリックかどうかを判定する式が入る。この式を答えよ。

(4-1), (4-2) には、それぞれマウスクリックの X 座標、Y 座標を取得する式が入る。(4-1) のほうを答えよ。

(5) には、`circ` を現在の描画のクリッピング領域として設定する式が入る。この式を答えよ。

さらに、この `SearchLight.java` を匿名クラスとラムダ式を用いて次のように同等のプログラム `SearchLight7.java` に書き換える。やはり `SearchLight7` 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: SearchLight7.java

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4 import java.awt.geom.*;
5
6 class SearchLight7 (6) {
7     /* フィールドは同じなので省略する */
8
9     public SearchLight7() {
10         setPreferredSize(new Dimension(width, height));
11         addMouseListener((7));
12         Timer timer = new Timer(50, (8));
13         timer.start();
14     }
15     /* paintComponent は同じなので省略する */
16 }
17 /* main は省略する */
```

(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。（解答用紙末尾に掲載の）

java.awt.event.MouseAdapter クラスを用いても良い。ただし、SearchLight.java と同じ部分は、`/* 元の AB 行目から YZ 行目と同じ */` のように省略せよ。

(8) の空欄に入る式を、ラムダ式を用いて書け。ただし、SearchLight.java と同じ部分は、

`/* 元の CD 行目から WX 行目と同じ */` のように省略せよ。

以下に参考のために授業配布プリントの MouseTest.java, LeftRightButton3.java, LeftRightButton4.java と MouseAdapter クラスのソース（の抜粋）を掲載する。

ファイル MouseTest.java

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.*; /* 1 */
4
5 class MouseTest extends JPanel
6     implements MouseListener /* 2 */ {
7     private int x = 50, y = 20;
8
9     public MouseTest() {
10         setPreferredSize(new Dimension(150, 150));
11         addMouseListener(this); /* 3 */
12     }
13
14     /* 4 */
15     public void mouseClicked(MouseEvent e) {
16         x = e.getX(); y = e.getY();
17         repaint();
18         return;
19     }
20
21     public void mousePressed(MouseEvent e) {} /* 5 */
22     public void mouseReleased(MouseEvent e) {} /* 5 */
23     public void mouseEntered(MouseEvent e) {} /* 5 */
24     public void mouseExited(MouseEvent e) {} /* 5 */
25
26     @Override
27     public void paintComponent(Graphics g) {
28         super.paintComponent(g);
29         g.drawString("HELLO WORLD!", x, y);
30     }
31 }
32
33 void main() { /* 省略 */ }
```

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; repaint();
15            }
16        });
17        rBtn.addActionListener(new ActionListener() {
18            public void actionPerformed(ActionEvent e) {
19                x += 10; repaint();
20            }
21        });
22        add(lBtn); add(rBtn);
23    }
24
25    @Override
26    public void paintComponent(Graphics g) {
```

```

27         super.paintComponent(g);
28         g.drawString("HELLO WORLD!", x, 55);
29     }
30 }
31
32 void main() { /* 省略 */ }

```

ファイル LeftRightButton4.java

```

1  import javax.swing.*;
2  import java.awt.*;
3
4  class LeftRightButton4 extends JPanel {
5      private int x = 20;
6
7      public LeftRightButton4() {
8          setPreferredSize(new Dimension(200, 70));
9          JButton lBtn = new JButton("Left");
10         JButton rBtn = new JButton("Right");
11         lBtn.addActionListener(e -> {
12             x -= 10; repaint();
13         });
14         rBtn.addActionListener(e -> {
15             x += 10; repaint();
16         });
17         setLayout(new FlowLayout());
18         add(lBtn); add(rBtn);
19     }
20
21     @Override
22     public void paintComponent(Graphics g) {
23         super.paintComponent(g);
24         g.drawString("HELLO WORLD!", x, 55);
25     }
26 }
27
28 void main() { /* 省略 */ }

```

java.awt.event.MouseAdapter クラス

```

1  // import 文などは掲載を省略する
2
3  public class MouseAdapter implements MouseListener {
4      public void mouseClicked(MouseEvent e) {}
5      public void mousePressed(MouseEvent e) {}
6      public void mouseReleased(MouseEvent e) {}
7      public void mouseEntered(MouseEvent e) {}
8      public void mouseExited(MouseEvent e) {}
9  }

```

オブジェクト指向言語・中間テスト解答用紙（2026 年 06 月 11 日）

学籍番号		氏名	
------	--	----	--

I. (2 × 3)

(1)		(2)	
-----	--	-----	--

II. (8 × 3)

(1)	
(2)	
(3)	
(4-1)	
(5)	
(6)	
(7)	
(8)	

授業・テストの感想は裏面へ

