

注意 問題は 3 問あります。問題中の Java プログラムの行頭の数値は、説明の都合上付けた行番号であり、プログラムの一部ではありません。また、この科目のクラスライブラリを利用している場合があります。解答はすべて別紙の解答用紙に記入しなさい。

問題 I 次は Swing の GUI 部品として働く 2 つのクラス Q1A と Q1B のクラス宣言を含む Java プログラムである。次ページの問いに答えなさい。

Q1.java

```
1 import java.awt.*;
2 import javax.swing.*;
3
4 class Q1 extends JPanel {
5     static final Image DEFAULT_IMAGE = new ImageIcon("default.png").getImage();
6     Image img;
7
8     protected void paintComponent(Graphics g) {
9         (a)
10    }
11 }
12
13 class Q1A extends Q1 {
14     Q1A(String image) {
15         try {
16             img = javax.imageio.ImageIO.read(new java.io.File(image));
17         } catch (java.io.IOException e) {
18             (b)
19         }
20         setPreferredSize(new Dimension(300, 200));
21     }
22
23     protected void paintComponent(Graphics g) {
24         super.paintComponent(g);
25         g.setColor(new Color(0, 255, 0));
26         g.fillRect(50, 50, 100, 80);
27         g.setColor(new Color(0.0f, 0.0f, 1.0f, 0.3f));
28         g.fillRect(50, 50, 100, 80);
29     }
30 }
31
32 class Q1B extends Q1 {
33     Q1B(String image) {
34         img = DEFAULT_IMAGE;
35         ImageIcon icon = (c);
36         if (icon.getImageLoadStatus() == MediaTracker.COMPLETE) {
37             (d)
38         }
39         setPreferredSize(new Dimension(300, 200));
40     }
41
42     protected void paintComponent(Graphics g) {
43         g.drawImage(img, 0, 0, getWidth(), getHeight(), this);
44         g.setColor(Color.BLUE);
45         g.fillRect(50, 50, 100, 80);
46         g.setColor(new Color((e), 1.0f, 0.0f, (f)));
47         g.fillRect(50, 50, 100, 80);
48     }
49 }
50 }
```

- (1) プログラム中の空欄を補って、Q1A と Q1B が GUI の部品として同じ表示となるようにしなさい。(25 点)
- (2) このプログラムに現れている下表中の語が、次のいずれであるかを、解答用紙の解答欄に 1 ～ 9 の数字で示しなさい。(30 点)

- | | | |
|----------------|-------------|------------------|
| 1. Java のキーワード | 2. パッケージ名 | 3. クラス名 |
| 4. インスタンスメソッド名 | 5. クラスメソッド名 | 6. クラス変数名 |
| 7. インスタンス変数名 | 8. 仮引数名 | 9. 局所 (ローカル) 変数名 |

行番号	語
8	Graphics
15	try
16	javax
16	read
17	IOException

行番号	語
28	g
37	icon
37	COMPLETE
44	img
44	getWidth

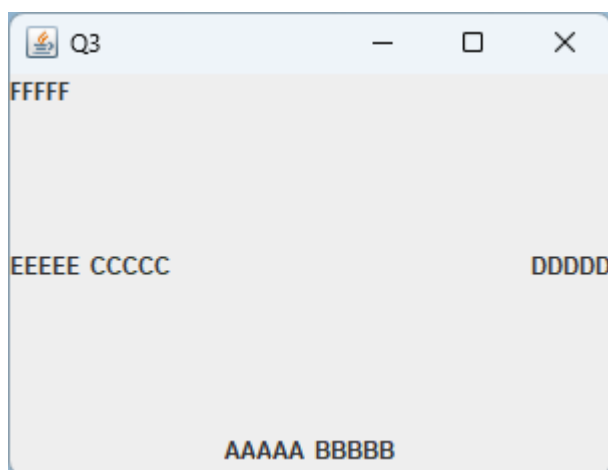
問題 II 次の Java プログラム Q2.java をコンパイルして、実行すると、問題末尾のようなウィンドウが表示された。プログラムの空欄を補いなさい。(20 点)

Q2.java

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

public class Q2 implements Runnable {
    (1) {
        JFrame f = new JFrame("Q2");
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container p = f.getContentPane();
        JPanel q = new JPanel();
        p.setPreferredSize(new Dimension(300, 200));
        p.setLayout(new BorderLayout(5, 5));
        q.setLayout(new FlowLayout());
        q.add(new JLabel("AAAAA"));
        q.add(new JLabel("BBBBB"));
        p.add(q, BorderLayout.(2));
        p.add(new JLabel("CCCCC"), BorderLayout.(3));
        p.add(new JLabel("DDDDD"), BorderLayout.(4));
        p.add(new JLabel("EEEEEE"), BorderLayout.(5));
        p.add(new JLabel("FFFFFF"), BorderLayout.NORTH);
        (6);
        f.setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater((7));
    }
}
```



問題 III 次の Java プログラム Q3.java は、「送る」というボタンが押される度に、デッキをシャッフルしてから一番上のカードを取り出し、表向きにして、別の山に移動していくようなプログラムである。

Q3.java

```
import javax.swing.*;
import java.awt.event.*;
import jp.ac.ryukoku.math.cards.*;

class Q3 {
    GameFrame frame = new GameFrame("Q3");
    JButton button = new JButton("送る");
    Pile deck = new Deck();
    Pile pile = new Pile();

    Q3() {
        frame.add(deck, 200, 200);
        frame.add(pile, 520, 200);
        frame.add(button, 210, 350);
        button.addActionListener(new Sender());
    }

    void send() {
        deck.shuffle();
        Card card = deck.pickUp();
        card.flip();
        card.moveTo(pile);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new Q3());
    }
}
```

- (1) プログラム中の空欄に、Q3 のメンバクラス **Sender** の宣言を補い、ボタンが押された時に、**send** メソッドが呼び出されるようにしなさい。ただし、スレッドを活用して、デッキのシャッフルやカードの反転、カードの移動のアニメーションが正しく表示されるようにすること。(15 点)
- (2) このプログラムでは、デッキが空になってしまうと、**deck.pickUp()** の戻り値が **null** となるため、**send** メソッド中で **NullPointerException** が発生してしまう。try 文を使ってこの例外をキャッチし、例外の発生が無視されるように **send** メソッドの宣言を書き換えなさい。(10 点)

問題 I

(1) (a) `g.drawImage(img, 0, 0, getWidth(), getHeight(), this);`

(b) `img = DEFAULT_IMAGE;`

(c) `new ImageIcon(image)`

(d) `img = icon.getImage();`

(e) `0.0f`

(f) `0.7f`

(2)

行番号	語	解答欄
8	Graphics	3
15	try	1
16	javax	2
16	read	5
17	IOException	3

行番号	語	解答欄
28	g	8
37	icon	9
37	COMPLETE	6
44	img	7
44	getWidth	4

問題 II

(1) `public void run ()`

(2) `SOUTH`

(3) `CENTER`

(4) `EAST`

(5) `WEST`

(6) `f.pack()`

(7) `new Q2()`

問題 III

(1)

```
class Sender implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        new Thread(() -> send()).start();
    }
}
```

(2) void send () {

```
    deck.shuffle();
    Card card = deck.pickUp();
    try {
        card.flip();
        card.moveTo(pile);
    }
    catch (NullPointerException e) {
    }
}
```

}