

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 次の Java プログラム Q01.java をコンパイルして起動すると、ウィンドウの中にトランプのデッキ (ジョーカーを含まない52枚の1揃えのカードの山) が現われ、デッキがシャッフルされた後、デッキから1枚ずつカードを引き、表向きにして、スート順、ランク順になるように4行13列の格子状に並べていく。その際、同じスートの隣り合うカードは重なってしまうので、カードを1枚並べるたびに、それより右側に置かれている同じスートのカードを順に raise (重なり順を最上位に) するようにしている。

Q01.java

```
1 import jp.ac.ryukoku.math.cards.*;
2
3 class Q01 {
4     (a) field = new Card (b) ;
5
6     public static void main(String[] args) {
7         GameFrame frame = new GameFrame();
8         Pile pile = new Deck();
9         frame.add(pile, 50, 60);
10        pile.shuffle();
11        while ( (c) ) {
12            Card card = pile.pickUp();
13            int r = card.suit.getNumber() - 1;
14            int c = card.rank.getNumber() - 1;
15            (d) ;
16            card.moveTo(c * 40 + 180, r * 120 + 60);
17            field[r][c] = card;
18            while (++c < field[r] (e) ) {
19                if ( (f) ) {
20                    field[r][c].raise();
21                }
22            }
23        }
24    }
25 }
```

- (1) 空欄 (a) ~ (f) に補うべきものを解答用紙の解答欄に書きなさい。(24 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 ~ 8 の数字で示しなさい。(24 点)
1. クラス名

2. インスタンスメソッド名

3. クラスメソッド名

4. クラス変数名

5. インスタンス変数名

6. ローカル (局所) 変数名

7. パッケージ名

8. Java のキーワード

行番号	語
1	import
6	String
8	Pile
9	frame

行番号	語
10	shuffle
13	suit
13	getNumber
17	field

(次ページに問題Ⅱ)

問題Ⅱ 次の Java プログラム Q02.java をコンパイルして実行したとする。このプログラムの 37 行目、42 行目、47 行目、52 行目、57 行目で表向きになるカードを、例えば「クラブの A」のように、それぞれ解答用紙の解答欄に書きなさい。(25 点)

Q02.java

```
1 import jp.ac.ryukoku.math.cards.*;
2
3 class P {
4     static Rank r = Rank.ACE;
5     Rank s = Rank.DEUCE;
6
7     P(Rank r) { s = r; }
8
9     Rank get() { return s; }
10
11     void set(Rank r) { s = r; }
12 }
13
14 class Q extends P {
15     Rank t = Rank.THREE;
16
17     Q() { super(Rank.FOUR); }
18
19     Rank get() { return t; }
20
21     void set(Rank r) {
22         super.set(Rank.FIVE);
23         t = r;
24     }
25 }
26
27 class R extends Q {
28     Rank get() { return s; }
29 }
30
31 class Q02 {
32     public static void main(String[] args) {
33         GameFrame f = new GameFrame();
34         P p = new P(Rank.SIX);
35         Card c = new Card(Suit.CLUBS, p.get());
36         f.add(c, 100, 100);
37         c.faceUp();
38
39         p = new Q();
40         c = new Card(Suit.CLUBS, p.get());
41         f.add(c, 200, 100);
42         c.faceUp();
43
44         p.set(Rank.SEVEN);
45         c = new Card(Suit.CLUBS, p.get());
46         f.add(c, 300, 100);
47         c.faceUp();
48
49         Q q = new R();
50         c = new Card(Suit.CLUBS, q.get());
51         f.add(c, 400, 100);
52         c.faceUp();
53
54         q.set(Rank.EIGHT);
55         c = new Card(Suit.CLUBS, q.get());
56         f.add(c, 500, 100);
57         c.faceUp();
58     }
59 }
```

(次ページに問題Ⅲ)

問題 III 次は第5回配布資料で紹介した Hand クラスの宣言である。

Hand.java

```
import jp.ac.ryukoku.math.cards.*;

class Hand {
    int x, y;                // 左端の手札の位置
    int deltaX = 100;        // 隣の手札との x 座標の差
    int numCards;            // 手札の枚数
    Card[] cards = new Card[5]; // 手札の配列

    Hand(int x, int y) {
        this.x = x;
        this.y = y;
    }

    void add(Card c) {
        if (numCards < cards.length) {
            c.moveTo(x + numCards * deltaX, y);
            cards[numCards++] = c;
        }
    }
}
```

この Hand クラスでは、配列 `cards` に空きがない (`numCards >= cards.length` の) 状態で `add` をメソッドを起動してもカードを追加することはできない。この Hand クラスのサブクラスとして、次のような ExtHand クラスの宣言を解答用紙の解答欄にかきなさい。(27 点)

- ExtHand クラスは Hand クラスの直接のサブクラスである。
- ExtHand クラスは、Hand クラスと同等のコンストラクタを持つ。
- ExtHand クラスでは、Hand のインスタンスメソッド `add` がオーバーライドされている。
- ExtHand クラスの `add` メソッドは、配列に空きがあれば、単に スーパクラスの `add` を起動するだけであるが、配列に空きがなければ、現在の配列の 2 倍の長さを持つ配列を新たに生成し、新しい配列に古い配列の内容をコピーし、インスタンス変数 `cards` の値を新しい配列で置き換えた上でスーパクラスの `add` を起動する。

例えば、次のプログラム Q03.java を実行すると、変数 `h` に記憶されている ExtHand クラスのインスタンスは、生成時には長さ 5 の配列を保持しているが、その後、2 回、新たな配列が生成されて、最終的には、長さ 20 の配列を保持することになる。

Q03.java

```
import jp.ac.ryukoku.math.cards.*;

class Q03 {
    public static void main(String[] args) {
        GameFrame f = new GameFrame();
        Deck d = new Deck();
        f.add(d);
        d.shuffle();
        Hand h = new ExtHand(160, 400);
        for (int i = 0; i < 15; i++) {
            Card c = d.pickUp();
            h.add(c);
            c.faceUp();
        }
    }
}
```

問題Ⅰ (1) (a)

static Card[] []

(b)

[4][13]

(c)

!pile.isEmpty()

(d)

card.flip()

(e)

.length

(f)

field[r][c] != null

(2)

行番号	語	解答欄
1	import	8
6	String	1
8	Pile	1
9	frame	6

行番号	語	解答欄
10	shuffle	2
13	suit	5
13	getNumber	2
17	field	4

問題Ⅱ 37行目

クラブの6

42行目

クラブの3

47行目

クラブの7

52行目

クラブの4

57行目

クラブの5

学籍番号

氏名

(裏面に続く)

問題 III

```
import jp.ac.ryukoku.math.cards.*;

class ExtHand extends Hand {
    ExtHand(int x, int y) {
        super(x, y);
    }

    void add(Card c) {
        if (numCards == cards.length) {
            Card[] newCards = new Card[cards.length * 2];
            for (int i = 0; i < cards.length; i++) {
                newCards[i] = cards[i];
            }
            cards = newCards;
        }
        super.add(c);
    }
}
```