

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 1 枚を含む 53 枚の 1 揃えのカードの山) と空の山が現われ、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、引いたカードを手札 (PairingHand クラスのインスタンス) に追加していく。ただし、引いたカードと同じランクのカードが手札にあれば、そのカードと引いたカードを山に捨てる。Hand クラスは付録 (4 ページ) のように宣言されているものとして、問いに答えなさい。

Q01.java

```
1 import jp.ac.ryukoku.math.cards.*;
2
3 class PairingHand extends Hand {
4     (a) ;
5
6     PairingHand(int x, int y, Pile discardPile) {
7         (b) (x, y, 40);
8         pile = discardPile;
9     }
10
11     void add(Card c) {
12         c.faceUp();
13         for (int i = 0; i < numCards; i++) {
14             if (cards[i].rank == c.rank) {
15                 discard(i, pile);
16                 c.moveTo(pile);
17                 (c) ;
18             }
19         }
20         if (numCards == cards.length) {
21             Card[] newCards = new Card[cards.length * 2];
22             for (int i = 0; i < cards.length; i++) {
23                 newCards[i] = cards[i];
24             }
25             cards = newCards;
26         }
27         (d) ;
28     }
29 }
30
31 class Q01 {
32     public static void main(String[] args) {
33         GameFrame frame = new GameFrame();
34         Deck deck = new Deck(1);
35         Pile pile = new Pile();
36         frame.add(deck, 100, 100);
37         frame.add(pile, 200, 100);
38         deck.shuffle();
39         Hand hand = new PairingHand(100, 300, pile);
40         while (!deck.isEmpty()) {
41             Card card = deck.pickUp();
42             (e) (card);
43         }
44     }
45 }
```

(次ページに続く)

(1) 空欄 (a) ~ (e) に補うべきものを解答用紙の解答欄に書きなさい。(20 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. インスタンス変数名 |
| 5. クラス変数名 | 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
3	extends
6	Pile
12	faceUp
15	discard

行番号	語
16	pile
20	numCards
25	newCards
38	deck

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 29 行目から 33 行目の各行が実行されることで標準出力に出力される文字列をそれぞれ解答用紙の解答欄に書きなさい。(25 点)

Q02.java

```
1 class P {
2     String s = "A";
3     void foo() { System.out.println("B"); }
4     void bar() { System.out.println(s); }
5     static void baz() { System.out.println("C"); }
6 }
7
8 class Q extends P {
9     void foo() {
10         System.out.print("D");
11         bar();
12     }
13     void bar() {
14         System.out.print("E");
15         super.foo();
16     }
17 }
18
19 class R extends Q {
20     void bar() { System.out.println("F"); }
21     static void baz() { System.out.println("G"); }
22 }
23
24 class Q02 {
25     public static void main(String[] args) {
26         P a = new P();
27         P b = new Q();
28         P c = new R();
29         a.foo();
30         b.foo();
31         b.bar();
32         c.foo();
33         Q.baz();
34     }
35 }
```

(次ページに問題 III)

問題 III Point クラスのインスタンスは、 xy 平面上を移動する点を表すオブジェクトとして働くことができる。下表は、その Point クラスのコンストラクタとインスタンスメソッド、クラスメソッドについてまとめたものである。

Point クラス — xy 平面上を移動する点を表すオブジェクトのクラス

コンストラクタ Point() Point(double x, double y)	初期座標が (0, 0) であるような動点として働くインスタンス 初期座標が (x, y) であるような動点として働くインスタンス
インスタンスメソッド void move(double x, double y) void shift(double dx, double dy) double getX() double getY()	(x, y) という座標に移動する。 x 軸方向に dx, y 軸方向に dy だけ平行移動する。 現在の x 座標を戻り値として返す。 現在の y 座標を戻り値として返す。
クラスメソッド Point midpoint(Point p, Point q)	点 p と点 q の中点を生成し、戻り値として返す。

例えば、次のプログラム Q03.java を実行すると、標準出力に

```
m = (-2.0, 1.5)
```

と出力される。

Q03.java

```
class Q03 {
    public static void main(String[] args) {
        Point p = new Point();
        Point q = new Point(-1.0, 3.0);
        p.move(-0.5, -2.0);
        q.shift(-2.5, 2.0);
        Point m = Point.midpoint(p, q);
        System.out.println("m = (" + m.getX() + ", " + m.getY() + ")");
    }
}
```

以上のような Point クラスのクラス宣言を解答用紙の解答欄に書きなさい。(31点)

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

class Hand {
    static final int CAPACITY = 5;

    int x, y;
    int deltaX = 100;
    int numCards;
    Card[] cards = new Card[CAPACITY];

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

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

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

    void discard(int i, Pile p) {
        if (0 <= i && i < numCards) {
            cards[i].moveTo(p);
            numCards--;
            while (i < numCards) {
                cards[i] = cards[i + 1];
                cards[i].moveTo(x + i * deltaX, y);
                i++;
            }
        }
    }
}
```

問題 I (1) (a)

Pile pile

(b)

super

(c)

return

(d)

super.add(c)

(e)

hand.add

(2)

行番号	語	解答欄
3	extends	8
6	Pile	1
12	faceUp	2
15	discard	2

行番号	語	解答欄
16	pile	4
20	numCards	4
25	newCards	6
38	deck	6

問題 II 29 行目 a.foo();

B

30 行目 b.foo();

DEB

31 行目 b.bar();

EB

32 行目 c.foo();

DF

33 行目 Q.baz();

C

学籍番号

氏名

(裏面に続く)

問題 III

```
class Point {
    private double x, y;

    Point() {
    }

    Point(double x, double y) {
        this.x = x;
        this.y = y;
    }

    void move(double x, double y) {
        this.x = x;
        this.y = y;
    }

    void shift(double dx, double dy) {
        x += dx;
        y += dy;
    }

    double getX() {
        return x;
    }

    double getY() {
        return y;
    }

    static Point midpoint(Point p, Point q) {
        return new Point((p.x + q.x) / 2.0, (p.y + q.y) / 2.0);
    }
}
```