情報理工学実験 I
オブジェクト指向プログラミング
ソースコードとアウトプット

Klak Bartosz
クラク バルトッシュ
学生番号 02166102

2018 年 4 月 18 日

1　課題 1.1

Listing 1　TestArray2.java

```java
public class TestArray2 {
    public static double getMean(double[] array){
        double ret=0.0;
        for(double value : array){ ret += value; }
        return ret/array.length;
    }
    public static double[] incArray(double[] array){
        double[] ret = new double[array.length];
        for(int i=0; i < array.length; i++){
            ret[i] = array[i] + 1;
        }
        return ret;
    }
    public static double getSum(double[] array){
        double ret=0.0;
        for(int i=0; i < array.length; i++){
                ret = ret + array[i];
        }
        return ret;
    }

    public static double getNonZeroProduct(double[] array){
        double ret=1.0;
        for(int i=0; i < array.length; i++){
                if(array[i]!=0){
                        ret = ret * array[i];
                }
        }
        return ret;
    }

    public static void main(String[] args) {
        double[] a1 = {1.012, -2.599, 3.421};
        System.out.printf("mean: %1.3f\n",getMean(a1));
        double[] a2 = incArray(a1);
        for (double value : a2) {
            System.out.printf("%1.3f ", value);
        }
        System.out.println();
        System.out.printf("mean: %1.3f\n",getMean(a2));
        double[] b1 = {1,2,3,4,5,6,7,8,9,10};
        double[] b2 = {-5,-4,-3,-2,-1,0,1,2,3,4,5};

        System.out.printf("sum of b1: %1.0f\n",getSum(b1));
        System.out.printf("nonzeroproduct of b1: %1.0f\n",getNonZeroProduct(b1));
        System.out.printf("sum of b2: %1.0f\n",getSum(b2));
        System.out.printf("nonzeroproduct of b2: %1.0f\n",getNonZeroProduct(b2));
    }
}
```

TestArray2.java の console output:

mean: 0.611

2.012 -1.599 4.421

mean: 1.611

sum of b1: 55

nonzeroproduct of b1: 3628800

sum of b2: 0

nonzeroproduct of b2: -14400

2　課題 1.2

Listing 2　Kadai2.java

```
1  import java.util.ArrayList;
2
3  public class Kadai2 {
4      public static void main(String[] args) {
5          ArrayList<Integer> array = new ArrayList<>();
6          for(int i=0; i<args.length; i++){
7                  array.add(Integer.parseInt(args[i]));
8          }
9
10         for(int i=0; i<args.length/2; i++){
11                 int temp = array.get(i);
12                 array.set(i, array.get(args.length -i -1));
13                 array.set(args.length -i -1, temp);
14         }
15         for(Integer c : array){
16             System.out.println(c);
17         }
18
19     }
20 }
```

Kadai2.java の console output:

0

8

9

3

4

3

1

3　課題 1.3

Listing 3　Birthday.java

```
import java.util.Scanner;
import java.text.SimpleDateFormat;
import java.time.LocalDate;
import java.time.Period;
import java.time.temporal.ChronoUnit;
import java.util.Calendar;


public class Birthday{
        public static void main(String args[]){

                System.out.println("Input your birthday (year month day): ");
                Scanner scanner = new Scanner(System.in);
                int y = scanner.nextInt();
                int m = scanner.nextInt();
                int d = scanner.nextInt();
                Calendar c = Calendar.getInstance();
                c.set(y, m-1, d);
       SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd (E)");
                System.out.println("Your birthday was on " + sdf.format(c.getTime()));
                scanner.close();

                LocalDate kanreki = LocalDate.of(y+60, m, d);
       LocalDate today = LocalDate.now();
       Period p = Period.between(today, kanreki);
       long p2 = ChronoUnit.DAYS.between(today, kanreki);
       System.out.println("There are " + p.getYears() + " years, "
                          + p.getMonths() + " months, and "
                          + p.getDays() +  " days until your 60th birthday. ("
                          + p2 + " days left)\n"+
                          + p2 * 24 + " hours left.");

        }
}
```

Birthday.java の console output:

Input your birthday (year month day):

1994 03 17

Your birthday was on 1994/03/17 (木)

There are 35 years, 10 months, and 27 days until your 60th birthday. (13117 days left)

314808 hours left.

4　課題 1.4

Listing 4　HowMuchTime.java

```
1  import java.util.Scanner;
2  import java.text.SimpleDateFormat;
3  import java.time.LocalDate;
4  import java.time.Period;
5  import java.time.temporal.ChronoUnit;
6  import java.util.Calendar;
7
8  public class HowMuchTime{
9          public static void main(String args[]){
10                 System.out.println("Input your birthday (year month day): ");
11                 Scanner scanner = new Scanner(System.in);
12                 int y = scanner.nextInt();
13                 int m = scanner.nextInt();
14                 int d = scanner.nextInt();
15                 Calendar c = Calendar.getInstance();
16                 c.set(y, m-1, d);
17         SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd (E)");
18                 System.out.println("Your birthday was on " + sdf.format(c.getTime()) + "\n");
19
20
21                 System.out.println("Input the date to check how old you will be then (year month day): ");
22                 Scanner scanner2 = new Scanner(System.in);
23                 int y2 = scanner.nextInt();
24                 int m2 = scanner.nextInt();
25                 int d2 = scanner.nextInt();
26                 Calendar c2 = Calendar.getInstance();
27                 c2.set(y2, m2-1, d2);
28         SimpleDateFormat sdf2 = new SimpleDateFormat("yyyy/MM/dd (E)");
29                 System.out.println("You want to check for " + sdf2.format(c2.getTime()) + "\n");
30
31
32                 LocalDate birthday = LocalDate.of(y, m, d);
33         LocalDate then = LocalDate.of(y2, m2, d2);
34         Period p = Period.between(birthday, then);
35         long p2 = ChronoUnit.DAYS.between(birthday, then);
36         System.out.println("On " + sdf2.format(c2.getTime()) + " you will be "
37                                  + p.getYears() + " years, "
38                            + p.getMonths() + " months, and "
39                            + p.getDays() +  " days old\nwhich is "
40                            + p2 + " days in total\nwhich is "+
41                            + p2 * 24 + " hours in total.");
42         scanner.close();
43         scanner2.close();
44          }
45 }
```

HowMuchTime.java の console output:

Input your birthday (year month day):

1916 4 30

Your birthday was on 1916/04/30 (日)

Input the date to check how old you will be then (year month day):

1995 08 24

You want to check for 1995/08/24 (木)

On 1995/08/24 (木) you will be 79 years, 3 months, and 25 days old

which is 28970 days in total

which is 695280 hours in total.

5　課題 1.5

Listing 5　Hand.java

```
enum Hand {ROCK, PAPER, SCISSORS}
```

Listing 6　RandomJankenPlayer.java

```
public class RandomJankenPlayer {
    private String name;
    public RandomJankenPlayer(String name){
        this.name = name;
    }
    public Hand showHand(){
        Hand play;
        double rnd = Math.random();
        if(rnd < 1.0/3.0){
            play = Hand.ROCK;
        }else if (rnd < 2.0/3.0){
            play = Hand.PAPER;
        }else{
            play = Hand.SCISSORS;
        }
        return play;
    }
    public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }
        // main
    public static void main(String[] args){
        RandomJankenPlayer player1 = new RandomJankenPlayer("Yamada");
        System.out.println(player1.getName());
        RandomJankenPlayer player2 = new RandomJankenPlayer("Suzuki");
        System.out.println(player2.getName());
        Hand hand1, hand2;
        for(int i=0; i<10; i++){
            hand1 = player1.showHand();
            hand2 = player2.showHand();
            System.out.println(i+")"+
                               "["+player1.name+"]"+hand1+" vs "+
                               "["+player2.name+"]"+hand2);
        }
    }
}
```

Listing 7　Judge.java

```
1  public class Judge {
2      private String name;
3      private RandomJankenPlayer player1, player2;
4      public Judge(String _name){
5          this.name = _name;
6      }
7
8      public String getName(){
9          return this.name;
10     }
11
12     public void setPlayers(RandomJankenPlayer _player1,
13                            RandomJankenPlayer _player2){
14         player1 = _player1;
15         player2 = _player2;
16     }
17
18     public void play(int n){
19         Hand hand1, hand2;
20
21         int win1=0, win2=0;
22         int lose1=0, lose2=0;
23         int draw1=0, draw2=0;
24         int i;
25
26         for(i=0;i<n;i++){
27                 hand1 = player1.showHand();
28                 hand2 = player2.showHand();
29
30                 if (hand1 == hand2){
31                         draw1=draw1+1;
32                         draw2=draw2+1;
33                 }else if((hand1 == Hand.ROCK && hand2 == Hand.SCISSORS)||(hand1 == Hand.PAPER && hand2 ==
                       Hand.ROCK)||(hand1 == Hand.SCISSORS && hand2 == Hand.PAPER) ){
34                         win1=win1+1;
35                         lose2=lose2+1;
36                 }else{
37                         win2=win2+1;
38                         lose1=lose1+1;
39                 }
40         }
41
42         System.out.println("Player1 : "+player1.getName());
43         System.out.println("Player2 : "+player2.getName());
44         System.out.println("Judge   : "+this.getName());
45         System.out.println();
46         System.out.println("Results: "+n+" games");
47         System.out.println(player1.getName()+" "+win1+" win, "+lose1+" lose, "+draw1+" draw");
48         System.out.println(player2.getName()+" "+win2+" win, "+lose2+" lose, "+draw2+" draw");
49     }
50     public static void main(String[] args) {
51         try{
52             int num = Integer.parseInt(args[0]);
53             RandomJankenPlayer player1 = new RandomJankenPlayer("Yamada");
54             RandomJankenPlayer player2 = new RandomJankenPlayer("Suzuki");
55             Judge judge = new Judge("Sato");
56             judge.setPlayers(player1,player2);
57             judge.play(num);
58         }catch(Exception e){
59             System.out.println("this requires an integer argument.");
60         }
61     }
62 }
```

Judge.java の console output (n=100 の場合):

Player1 : Yamada
Player2 : Suzuki
Judge : Sato

Results: 100 games
Yamada 24 win, 37 lose, 39 draw
Suzuki 37 win, 24 lose, 39 draw

6　課題 1.6

Listing 8　Judge.java

```
1  public class Judge {
2      private String name;
3      private RandomJankenPlayer player1, player2;
4      public Judge(String _name){
5          this.name = _name;
6      }
7
8      public String getName(){
9          return this.name;
10     }
11
12     public void setPlayers(RandomJankenPlayer _player1,
13                            RandomJankenPlayer _player2){
14         player1 = _player1;
15         player2 = _player2;
16     }
17     public void play(int m){
18         Hand hand1, hand2;
19
20         double win1=0, win2=0;
21         double lose1=0, lose2=0;
22         double draw1=0, draw2=0;
23         int i,j;
24
25         for(j=0;j<m;j++){
26         for(i=0;i<10;i++){
27                 hand1 = player1.showHand();
28                 hand2 = player2.showHand();
29
30                 if (hand1 == hand2){
31                         draw1=draw1+1;
32                         draw2=draw2+1;
33                 }else if((hand1 == Hand.ROCK && hand2 == Hand.SCISSORS)||(hand1 == Hand.PAPER && hand2 ==
                       Hand.ROCK)||(hand1 == Hand.SCISSORS && hand2 == Hand.PAPER) ){
34                         win1=win1+1;
35                         lose2=lose2+1;
36                 }else{
37                         win2=win2+1;
38                         lose1=lose1+1;
39                 }
40         }}
41
42         System.out.println("Player1 : "+player1.getName());
43         System.out.println("Player2 : "+player2.getName());
44         System.out.println("Judge   : "+this.getName());
45         System.out.println();
46
47         System.out.println("Average results for "+m+" sets of 10 games");
48         double w1, w2, l1, l2, d1, d2;
49         w1=win1/m;
50         w2=win2/m;
51         l1=lose1/m;
52         l2=lose2/m;
53         d1=draw1/m;
54         d2=draw2/m;
55         System.out.println(player1.getName()+" "+w1+" win, "+l1+" lose, "+d1+" draw");
56         System.out.println(player2.getName()+" "+w2+" win, "+l2+" lose, "+d2+" draw");
57     }
58     public static void main(String[] args) {
59         try{
60             int num = Integer.parseInt(args[0]);
61             RandomJankenPlayer player1 = new RandomJankenPlayer("Yamada");
62             RandomJankenPlayer player2 = new RandomJankenPlayer("Suzuki");
63             Judge judge = new Judge("Sato");
64             judge.setPlayers(player1,player2);
65             judge.play(num);
66         }catch(Exception e){
67             System.out.println("this requires an integer argument.");
68         }
69     }
70 }
```

console output (m=1000 の場合):

Player1 : Yamada
Player2 : Suzuki
Judge : Sato

Average results for 1000 sets of 10 games
Yamada 3.325 win, 3.342 lose, 3.333 draw
Suzuki 3.342 win, 3.325 lose, 3.333 draw

7　課題 1.7

Listing 9　Result.java

```
enum Result {WIN, LOSE, DRAW}
```

Listing 10　RandomJankenPlayer.java

```
public class RandomJankenPlayer {
    private String name;
    public RandomJankenPlayer(String name){
        this.name = name;
    }

    int n_win=0, n_lose=0, n_draw=0;

    public void notify(Result q){


        if(q==Result.WIN){
                n_win=n_win+1;
        }else if(q==Result.LOSE){
                n_lose=n_lose+1;
        }else if(q==Result.DRAW){
                n_draw=n_draw+1;
        }
    }

    public void report(){
        System.out.println(this.getName()+" "+n_win+" win, "+n_lose+" lose, "+n_draw+" draw");
    }

    public Hand showHand(){
        Hand play;
        double rnd = Math.random();
        if(rnd < 1.0/3.0){
            play = Hand.ROCK;
        }else if (rnd < 2.0/3.0){
            play = Hand.PAPER;
        }else{
            play = Hand.SCISSORS;
        }
        return play;
    }
    public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }

          public Integer getNWin() {
                        return n_win;
                }
          public Integer getNLose() {
                        return n_lose;
                }
          public Integer getNDraw() {
                        return n_draw;
                }
}
```

Listing 11　Judge.java

```
1  public class Judge {
2      private String name;
3      private RandomJankenPlayer player1, player2, player3;
4      public Judge(String _name){
5          this.name = _name;
6      }
7
8      public String getName(){
9          return this.name;
10     }
11
12     public void setPlayers(RandomJankenPlayer _player1,
13                            RandomJankenPlayer _player2,
14                            RandomJankenPlayer _player3){
15
16         player1=_player1;
17         player2=_player2;
18         player3=_player3;
19
20     }
21
22     public void play(int n){
23         Hand hand1, hand2;
24         RandomJankenPlayer p1,p2;
25         int c=0,d=0,e=0;
26         int j;
27
28
29         for(j=0; j<n; j++){
30                 double rnd = Math.random();
31                 if(rnd < 1.0/3.0){
32                         p1 = player1;
33                         p2 = player2;
34                         c=c+1;
35
36                 }else if (rnd < 2.0/3.0){
37                         p1 = player2;
38                         p2 = player3;
39                         d=d+1;
40
41                 }else{
42                         p1 = player3;
43                         p2 = player1;
44                         e=e+1;
45                 }
46
47                 boolean flag = true;
48                 while(flag){
49                         hand1 = p1.showHand();
50                 hand2 = p2.showHand();
51                 if (hand1 == hand2){
52                         p1.notify(Result.DRAW);
53                         p2.notify(Result.DRAW);
54                 }else if((hand1 == Hand.ROCK && hand2 == Hand.SCISSORS)||(hand1 == Hand.PAPER && hand2 ==
                       Hand.ROCK)||(hand1 == Hand.SCISSORS && hand2 == Hand.PAPER) ){
55                         p1.notify(Result.WIN);
56                         p2.notify(Result.LOSE);
57                         flag = false;
58                 }else{
59                         p1.notify(Result.LOSE);
60                         p2.notify(Result.WIN);
61                         flag =false;
62                                 }
63                         }
64         }
65
66         System.out.println("Judge   : "+this.getName());
67         System.out.println();
68         System.out.println("Results: "+ n +" games");
69         System.out.println("1 vs 2: "+ c +" games");
70         System.out.println("1 vs 3: "+ e +" games");
71         System.out.println("2 vs 3: "+ d +" games");
72         System.out.println();
73
74     }
75     public static void main(String[] args) {
76         try{
```

```
77              int num = Integer.parseInt(args[0]);
78              RandomJankenPlayer player1 = new RandomJankenPlayer("Yamada");
79              RandomJankenPlayer player2 = new RandomJankenPlayer("Suzuki");
80              RandomJankenPlayer player3 = new RandomJankenPlayer("Tanaka");
81              System.out.println("Player1 : "+player1.getName());
82              System.out.println("Player2 : "+player2.getName());
83              System.out.println("Player3 : "+player3.getName());
84
85              Judge judge = new Judge("Sato");
86              judge.setPlayers(player1, player2, player3);
87              judge.play(num);
88              player1.report();
89              player2.report();
90              player3.report();
91          }catch(Exception e){
92              System.out.println("this requires an integer argument.");
93          }
94      }
95  }
```

Judge.java の console output (n=20 の場合):

Player1 : Yamada

Player2 : Suzuki

Player3 : Tanaka

Judge : Sato

Results: 20 games

1 vs 2: 7 games

1 vs 3: 10 games

2 vs 3: 3 games

Yamada 10 win, 7 lose, 20 draw

Suzuki 5 win, 5 lose, 11 draw

Tanaka 5 win, 8 lose, 11 draw

8　課題 1.9

Listing 12　RandomJankenPlayer.java

```java
public class RandomJankenPlayer {
    private String name;

    public String getName() {
                return this.name;
        }
        public void setName(String name) {
                this.name = name;
        }

    public RandomJankenPlayer(){}
    public RandomJankenPlayer(String name){
        this.name = name;
    }


    public Hand showHand(){
        Hand play;
        double rnd = Math.random();
        if(rnd < 1.0/3.0){
            play = Hand.ROCK;
        }else if (rnd < 2.0/3.0){
            play = Hand.PAPER;
        }else{
            play = Hand.SCISSORS;
        }
        return play;
    }
}
```

Listing 13　InteractiveJankenPlayer.java

```java
import java.util.Scanner;
public class InteractiveJankenPlayer extends RandomJankenPlayer{

        public InteractiveJankenPlayer(){
                System.out.print("Your name? ");
        Scanner scanner = new Scanner(System.in);
        String name = scanner.nextLine();
        this.setName(name);
        }

        public Hand showHand(){
        Hand play = Hand.PAPER;
        Scanner choice = new Scanner(System.in);
        int t=1;
        while(t==1){
                System.out.print("Your hand? (R/S/P)");
                String letter = choice.nextLine();
        if (letter.equals("R")){
                play = Hand.ROCK;
                break;
        }else if (letter.equals("S")){
                play = Hand.SCISSORS;
                break;
        }else if (letter.equals("P")){
                play = Hand.PAPER;
                break;
        }else{
                System.out.println("Error: Insert R, S or P");
        }}
        return play;
    }
}
```

Listing 14　Judge.java

```
public class Judge {
    private String name;
    private RandomJankenPlayer player1, player2;
    public Judge(String _name){
        this.name = _name;
    }

    public String getName(){
        return this.name;
    }

    public void setPlayers(RandomJankenPlayer _player1,
                           RandomJankenPlayer _player2){
        player1 = _player1;
        player2 = _player2;
    }

    public void play(int n){
        System.out.println("Player1 : "+player1.getName());
        System.out.println("Player2 : "+player2.getName());
        System.out.println("Judge   : "+this.getName());
        System.out.println();
        System.out.println("Results: "+n+" games");



        Hand hand1, hand2;

        int win1=0, win2=0;
        int lose1=0, lose2=0;
        int draw1=0, draw2=0;
        int i;

        for(i=0;i<n;i++){


                hand1 = player1.showHand();
                hand2 = player2.showHand();
            System.out.println(player1.getName()+":"+hand1 +" vs "+player2.getName()+":" +hand2);


                if (hand1 == hand2){
                        draw1=draw1+1;
                        draw2=draw2+1;
              System.out.println("Draw...");
                }else if((hand1 == Hand.ROCK && hand2 == Hand.SCISSORS)||(hand1 == Hand.PAPER && hand2 ==
                     Hand.ROCK)||(hand1 == Hand.SCISSORS && hand2 == Hand.PAPER) ){
                        win1=win1+1;
                        lose2=lose2+1;
                        System.out.println(player1.getName()+" Win!");
                }else{
                        win2=win2+1;
                        lose1=lose1+1;
                        System.out.println(player2.getName()+" Win!");
                }
        }

        System.out.println();
        System.out.println(player1.getName()+" "+win1+" win, "+lose1+" lose, "+draw1+" draw");
        System.out.println(player2.getName()+" "+win2+" win, "+lose2+" lose, "+draw2+" draw");
    }
    public static void main(String[] args) {
        try{
            int num = Integer.parseInt(args[0]);
            RandomJankenPlayer player1 = new InteractiveJankenPlayer();
            RandomJankenPlayer player2 = new RandomJankenPlayer("Suzuki");
            Judge judge = new Judge("Sato");
            judge.setPlayers(player1,player2);
            judge.play(num);
        }catch(Exception e){
            System.out.println("this requires an integer argument.");
        }
    }
}
```

Judge.java の console output (n=5 の場合:)

```
Your name? Bartek
Player1 : Bartek
Player2 : Suzuki
Judge : Sato

Results: 5 games
Your hand? (R/S/P)R
Bartek:ROCK vs Suzuki:SCISSORS
Bartek Win!
Your hand? (R/S/P)P
Bartek:PAPER vs Suzuki:PAPER
Draw...
Your hand? (R/S/P)S
Bartek:SCISSORS vs Suzuki:PAPER
Bartek Win!
Your hand? (R/S/P)W
Error: Insert R, S or P
Your hand? (R/S/P)P
Bartek:PAPER vs Suzuki:PAPER
Draw...
Your hand? (R/S/P)R
Bartek:ROCK vs Suzuki:PAPER
Suzuki Win!

Bartek 2 win, 1 lose, 2 draw
Suzuki 1 win, 2 lose, 2 draw
```

9 課題 1.10

Listing 15 RandomJankenPlayer.java

```
1  public class RandomJankenPlayer {
2      private String name;
3
4      public String getName() {
5                  return this.name;
6          }
7          public void setName(String name) {
8                  this.name = name;
9          }
10
11     public RandomJankenPlayer(){}
12     public RandomJankenPlayer(String name){
13         this.name = name;
14     }
15
16     int n_win=0, n_lose=0, n_draw=0;
17
18     public void notify(Result q){
19
20         if(q==Result.WIN){
21                 n_win=n_win+1;
22         }else if(q==Result.LOSE){
23                 n_lose=n_lose+1;
24         }else if(q==Result.DRAW){
25                 n_draw=n_draw+1;
26         }
27     }
28
29     public void report(){
30         System.out.println(this.getName()+" "+n_win+" win, "+n_lose+" lose, "+n_draw+" draw");
31     }
32
33           public Integer getNWin() {
34                   return n_win;
35                 }
36           public Integer getNLose() {
37                   return n_lose;
38                 }
39           public Integer getNDraw() {
40                   return n_draw;
41                 }
42
43     public Hand showHand(){
44         Hand play;
45         double rnd = Math.random();
46         if(rnd < 1.0/3.0){
47             play = Hand.ROCK;
48         }else if (rnd < 2.0/3.0){
49             play = Hand.PAPER;
50         }else{
51             play = Hand.SCISSORS;
52         }
53         return play;
54     }
55 }
```

Listing 16　JankenPlayerTypeA.java

```
public class JankenPlayerTypeA extends RandomJankenPlayer {
    public JankenPlayerTypeA(String name){
        super(name);
    }

    JankenPlayerTypeA a;

    private Hand myHand;
    boolean check2 = false;


    public void notify(Result q){

        if(q==Result.WIN){
                n_win=n_win+1;
                check2 = false;
        }else if(q==Result.LOSE){
                n_lose=n_lose+1;
                check2 = true;
        }else if(q==Result.DRAW){
                n_draw=n_draw+1;
                check2 = false;
        }
    }

    public Hand showHand(){
        Hand play = Hand.PAPER;

        if(check2 == true){
                if(myHand == Hand.ROCK){
                        play = Hand.PAPER;
                        myHand = Hand.PAPER;
                }else if(myHand == Hand.PAPER){
                        play = Hand.SCISSORS;
                        myHand = Hand.SCISSORS;
                }else if(myHand == Hand.SCISSORS){
                        play = Hand.ROCK;
                        myHand = Hand.ROCK;
                }
                return play;
        }else{
                double rnd = Math.random();
            if(rnd < 1.0/3.0){
                play = Hand.ROCK;
                myHand = Hand.ROCK;
            }else if (rnd < 2.0/3.0){
                play = Hand.PAPER;
                myHand = Hand.PAPER;
            }else{
                play = Hand.SCISSORS;
                myHand = Hand.SCISSORS;
             }
            return play;
        }
    }
}
```

Listing 17　Judge.java

```
1
2
3  public class Judge {
4      private String name;
5      private RandomJankenPlayer player1, player2;
6      public Judge(String _name){
7          this.name = _name;
8      }
9
10     public String getName(){
11         return this.name;
12     }
13
14     public void setPlayers(RandomJankenPlayer _player1,
15                            RandomJankenPlayer _player2){
16         player1 = _player1;
17         player2 = _player2;
18     }
19
20     public void play(int n){
21         System.out.println("Player1 : "+player1.getName());
22         System.out.println("Player2 : "+player2.getName());
23         System.out.println("Judge   : "+this.getName());
24         System.out.println();
25         System.out.println("Results: "+n+" games");
26
27         Hand hand1, hand2;
28         RandomJankenPlayer p1,p2;
29        // int c=0,d=0,e=0;
30         int j;
31         for(j=0; j<n; j++){
32                         p1 = player1;
33                         p2 = player2;
34
35                         hand1 = p1.showHand();
36                 hand2 = p2.showHand();
37                 System.out.println(player1.getName()+":"+hand1 +" vs "+player2.getName()+":" +hand2);
38                 if (hand1 == hand2){
39                         p1.notify(Result.DRAW);
40                         p2.notify(Result.DRAW);
41                          System.out.println("Draw...");
42                 }else if((hand1 == Hand.ROCK && hand2 == Hand.SCISSORS)||(hand1 == Hand.PAPER && hand2 ==
                        Hand.ROCK)||(hand1 == Hand.SCISSORS && hand2 == Hand.PAPER) ){
43                         p1.notify(Result.WIN);
44                         p2.notify(Result.LOSE);
45                         System.out.println(player1.getName()+" Win!");
46                 }else{
47                         p1.notify(Result.LOSE);
48                         p2.notify(Result.WIN);
49                         System.out.println(player2.getName()+" Win!");
50                         }
51
52         }
53     }
54     public static void main(String[] args) {
55         try{
56             int num = Integer.parseInt(args[0]);
57             RandomJankenPlayer player1 = new JankenPlayerTypeA("Yamada");
58             RandomJankenPlayer player2 = new RandomJankenPlayer("Suzuki");
59             Judge judge = new Judge("Sato");
60             judge.setPlayers(player1,player2);
61             judge.play(num);
62             System.out.println();
63             player1.report();
64             player2.report();
65         }catch(Exception e){
66             System.out.println("this requires an integer argument.");
67         }
68     }
69 }
```

Judge.java の console output (n=10 の場合):

```
Player1 : Yamada
Player2 : Suzuki
Judge : Sato

Results: 10 games
Yamada:ROCK vs Suzuki:SCISSORS
Yamada Win!
Yamada:PAPER vs Suzuki:PAPER
Draw...
Yamada:PAPER vs Suzuki:SCISSORS
Suzuki Win!
Yamada:SCISSORS vs Suzuki:ROCK
Suzuki Win!
Yamada:ROCK vs Suzuki:PAPER
Suzuki Win!
Yamada:PAPER vs Suzuki:PAPER
Draw...
Yamada:ROCK vs Suzuki:ROCK
Draw...
Yamada:ROCK vs Suzuki:SCISSORS
Yamada Win!
Yamada:ROCK vs Suzuki:PAPER
Suzuki Win!
Yamada:PAPER vs Suzuki:ROCK
Yamada Win!

Yamada 3 win, 4 lose, 3 draw
Suzuki 4 win, 3 lose, 3 draw
```

10　課題 1.12

Listing 18　NameAvailable.java

```
1  public abstract interface NameAvailable {
2      public abstract String getName();
3
4  }
```

Listing 19　JankenPlayer.java

```
1  public abstract class JankenPlayer implements NameAvailable{
2
3          public String getName() {
4                          return this.name;
5                  }
6
7     private String name;
8
9          public JankenPlayer(){}
10         public JankenPlayer(String name){
11         this.name = name;
12     }
13
14         public void setName(String name) {
15                 this.name = name;
16         }
17
18     public void notify(Result q){
19
20         if(q==Result.WIN){
21                 n_win=n_win+1;
22         }else if(q==Result.LOSE){
23                 n_lose=n_lose+1;
24         }else if(q==Result.DRAW){
25                 n_draw=n_draw+1;
26         }
27     }
28
29     public void report(){
30         System.out.println(this.getName()+" "+n_win+" win, "+n_lose+" lose, "+n_draw+" draw");
31     }
32
33         public abstract Hand showHand();
34
35     int n_win=0, n_lose=0, n_draw=0;
36           public Integer getNWin() {
37                          return n_win;
38                  }
39           public Integer getNLose() {
40                          return n_lose;
41                  }
42           public Integer getNDraw() {
43                          return n_draw;
44                  }
45 }
```

Listing 20 InteractiveJankenPlayer.java

```
import java.util.Scanner;
public class InteractiveJankenPlayer extends JankenPlayer implements NameAvailable {
        public InteractiveJankenPlayer(){
                System.out.print("Your name? ");
        Scanner scanner = new Scanner(System.in);
        String name = scanner.nextLine();
        this.setName(name);
        }

    public Hand showHand(){
        Hand play = Hand.PAPER;
        Scanner choice = new Scanner(System.in);
        int t=1;
        while(t==1){
                System.out.print("Your hand? (R/S/P)");
                String letter = choice.nextLine();
                if (letter.equals("R")){
                        play = Hand.ROCK;
                        break;
                }else if (letter.equals("S")){
                        play = Hand.SCISSORS;
                        break;
        }else if (letter.equals("P")){
                play = Hand.PAPER;
                break;
        }else{
                System.out.println("Error: Insert R, S or P");
        }}
        return play;
    }
}
```

Listing 21 RandomJankenPlayer.java

```
public class RandomJankenPlayer extends JankenPlayer implements NameAvailable{
    private String name;

    public String getName() {
                return this.name;
        }
        public void setName(String name) {
                this.name = name;
        }

    public RandomJankenPlayer(){}
    public RandomJankenPlayer(String name){
        this.name = name;
    }

    int n_win=0, n_lose=0, n_draw=0;

    public void notify(Result q){

        if(q==Result.WIN){
                n_win=n_win+1;
        }else if(q==Result.LOSE){
                n_lose=n_lose+1;
        }else if(q==Result.DRAW){
                n_draw=n_draw+1;
        }
    }

    public void report(){
        System.out.println(this.getName()+" "+n_win+" win, "+n_lose+" lose, "+n_draw+" draw");
    }

          public Integer getNWin() {
                        return n_win;
                }
          public Integer getNLose() {
                        return n_lose;
                }
          public Integer getNDraw() {
                        return n_draw;
                }

```

```
43      public Hand showHand(){
44          Hand play;
45          double rnd = Math.random();
46          if(rnd < 1.0/3.0){
47              play = Hand.ROCK;
48          }else if (rnd < 2.0/3.0){
49              play = Hand.PAPER;
50          }else{
51              play = Hand.SCISSORS;
52          }
53          return play;
54      }
55  }
```

Listing 22　JankenPlayerTypeA.java

```
1   public class JankenPlayerTypeA extends JankenPlayer implements NameAvailable{
2       public JankenPlayerTypeA(String name){
3           super(name);
4       }
5
6       JankenPlayerTypeA a;
7
8       private Hand myHand;
9       boolean check2 = false;
10
11
12      public void notify(Result q){
13
14          if(q==Result.WIN){
15                  n_win=n_win+1;
16                  check2 = false;
17          }else if(q==Result.LOSE){
18                  n_lose=n_lose+1;
19                  check2 = true;
20          }else if(q==Result.DRAW){
21                  n_draw=n_draw+1;
22                  check2 = false;
23          }
24      }
25
26
27      public Hand showHand(){
28          Hand play = Hand.PAPER;
29
30          if(check2 == true){
31                  if(myHand == Hand.ROCK){
32                          play = Hand.PAPER;
33                          myHand = Hand.PAPER;
34                  }else if(myHand == Hand.PAPER){
35                          play = Hand.SCISSORS;
36                          myHand = Hand.SCISSORS;
37                  }else if(myHand == Hand.SCISSORS){
38                          play = Hand.ROCK;
39                          myHand = Hand.ROCK;
40                  }
41                  return play;
42          }else{
43                  double rnd = Math.random();
44              if(rnd < 1.0/3.0){
45                  play = Hand.ROCK;
46                  myHand = Hand.ROCK;
47              }else if (rnd < 2.0/3.0){
48                  play = Hand.PAPER;
49                  myHand = Hand.PAPER;
50              }else{
51                  play = Hand.SCISSORS;
52                  myHand = Hand.SCISSORS;
53               }
54              return play;
55          }
56      }
57  }
```

Listing 23 Judge.java

```
1  import java.util.Collections;
2  import java.util.ArrayList;
3  import java.util.Scanner;
4
5  public class Judge  implements NameAvailable {
6      private String name;
7      private JankenPlayer player1, player2;
8      public Judge(String _name){
9          this.name = _name;
10     }
11
12     public String getName(){
13         return this.name;
14     }
15
16     public void setPlayers(JankenPlayer _player1,
17                            JankenPlayer _player2){
18         player1 = _player1;
19         player2 = _player2;
20     }
21
22     public void play(int n){
23         System.out.println("Player1 : "+player1.getName());
24         System.out.println("Player2 : "+player2.getName());
25         System.out.println("Judge   : "+this.getName());
26         System.out.println();
27         System.out.println("Results: "+n+" games");
28
29         Hand hand1, hand2;
30         JankenPlayer p1,p2;
31        // int c=0,d=0,e=0;
32         int j;
33         for(j=0; j<n; j++){
34                         p1 = player1;
35                         p2 = player2;
36
37                         hand1 = p1.showHand();
38                 hand2 = p2.showHand();
39                 System.out.println(player1.getName()+":"+hand1 +" vs "+player2.getName()+":" +hand2);
40                 if (hand1 == hand2){
41                         p1.notify(Result.DRAW);
42                         p2.notify(Result.DRAW);
43                          System.out.println("Draw...");
44                 }else if((hand1 == Hand.ROCK && hand2 == Hand.SCISSORS)||(hand1 == Hand.PAPER && hand2 ==
                       Hand.ROCK)||(hand1 == Hand.SCISSORS && hand2 == Hand.PAPER) ){
45                         p1.notify(Result.WIN);
46                         p2.notify(Result.LOSE);
47                         System.out.println(player1.getName()+" Win!");
48                 }else{
49                         p1.notify(Result.LOSE);
50                         p2.notify(Result.WIN);
51                         System.out.println(player2.getName()+" Win!");
52                         }
53
54         }
55     }
56
57     public void members(Judge judge, JankenPlayer _player11, JankenPlayer _player22){
58         System.out.print("Member list? (Y/N) ");
59         Scanner scanner = new Scanner(System.in);
60         int t=1;
61                 while(t==1){
62
63                 String letter = scanner.nextLine();
64                         if (letter.equals("Y")){
65                                 ArrayList<String> name = new ArrayList<>();
66                         name.add(_player11.getName());
67                         name.add(_player22.getName());
68                         name.add(judge.getName());
69                         Collections.sort(name);
70                         for(String c : name){
71                             System.out.println(c);
72                         }
73                         System.out.println();
74                                 break;
75                         }else if (letter.equals("N")){
76                                 break;
```

```
77                          }else{
78                                  System.out.print("Error: Insert Y or N ");
79                          }
80                  }
81      }
82
83      public static void main(String[] args) {
84          try{
85              int num = Integer.parseInt(args[0]);
86              JankenPlayer player1 = new JankenPlayerTypeA("Yamada");
87              JankenPlayer player2 = new InteractiveJankenPlayer();
88              Judge judge = new Judge("Sato");
89              judge.setPlayers(player1,player2);
90              judge.members(judge, player1, player2);
91
92                  judge.play(num);
93                  System.out.println();
94                  player1.report();
95                  player2.report();
96          }catch(Exception e){
97              System.out.println("this requires an integer argument.");
98          }
99      }
100 }
```

Judge.java の console output (n=3 の場合):

Your name? Bartek

Member list? (Y/N) Y

Bartek

Sato

Yamada

Player1 : Yamada

Player2 : Bartek

Judge : Sato

Results: 3 games

Your hand? (R/S/P)R

Yamada:ROCK vs Bartek:ROCK

Draw...

Your hand? (R/S/P)R

Yamada:SCISSORS vs Bartek:ROCK

Bartek Win!

Your hand? (R/S/P)P

Yamada:ROCK vs Bartek:PAPER

Bartek Win!

Yamada 0 win, 2 lose, 1 draw

Bartek 2 win, 0 lose, 1 draw

11　課題 1.13

Listing 24　JankenPlayer.java

```
1  public abstract class JankenPlayer {
2
3          public String getName() {
4                          return this.name;
5                  }
6
7     private String name;
8
9          public JankenPlayer(){}
10         public JankenPlayer(String name){
11         this.name = name;
12     }
13
14         public void setName(String name) {
15                 this.name = name;
16         }
17
18     public void notify(Result q){
19
20         if(q==Result.WIN){
21                 n_win=n_win+1;
22         }else if(q==Result.LOSE){
23                 n_lose=n_lose+1;
24         }else if(q==Result.DRAW){
25                 n_draw=n_draw+1;
26         }
27     }
28
29    public void report(){
30         System.out.println("There were "+n_draw+" draws");
31    }
32
33         public abstract Hand showHand();
34
35     int n_win=0, n_lose=0, n_draw=0;
36           public Integer getNWin() {
37                          return n_win;
38                  }
39           public Integer getNLose() {
40                          return n_lose;
41                  }
42           public Integer getNDraw() {
43                          return n_draw;
44                  }
45
46
47 }
```

Listing 25　RandomJankenPlayer.java

```
public class RandomJankenPlayer extends JankenPlayer{
    private String name;

    public String getName() {
                return this.name;
        }
        public void setName(String name) {
                this.name = name;
        }

    public RandomJankenPlayer(){}
    public RandomJankenPlayer(String name){
        this.name = name;
    }

    int n_win=0, n_lose=0, n_draw=0;

          public Integer getNWin() {
                        return n_win;
                }
          public Integer getNLose() {
                        return n_lose;
                }
          public Integer getNDraw() {
                        return n_draw;
                }

    public Hand showHand(){
        Hand play;
        double rnd = Math.random();
        if(rnd < 1.0/3.0){
            play = Hand.ROCK;
        }else if (rnd < 2.0/3.0){
            play = Hand.PAPER;
        }else{
            play = Hand.SCISSORS;
        }
        return play;
    }
}
```

Listing 26　Judge2.java

```
import java.util.Scanner;
import java.util.HashMap;

public class Judge3 {
    private String name;
    private JankenPlayer[] players;
    private Hand[] hands;
    public Judge3(String name){
        this.name = name;
    }
    public String getName(){
        return this.name;
    }
    public void setPlayers(JankenPlayer[] players){
        this.players = players;
    }
    private void notifyAll(Hand handWin, Hand handLose){
        for(int i=0; i<players.length; i++){
            if(hands[i]==handWin){
                players[i].notify(Result.WIN);
            }else if(hands[i]==handLose){
                players[i].notify(Result.LOSE);
            }
        }
    }
    public void play(){
        hands = new Hand[players.length];
        int countRock=0, countScissors=0, countPaper=0;
        for(int i=0; i<players.length; i++){
            hands[i] = players[i].showHand();
            if(hands[i]==Hand.ROCK){
                countRock += 1;
            }else if(hands[i]==Hand.SCISSORS){
                countScissors += 1;
            }else if(hands[i]==Hand.PAPER){
                countPaper += 1;
            }
       //     System.out.println(players[i].getName()+" "+hands[i]);
        }
        if(countRock==players.length ||
           countScissors==players.length ||
           countPaper==players.length ||
           countRock*countScissors*countPaper!=0){ // Draw
            for(int i=0; i<players.length; i++){
                players[i].notify(Result.DRAW);
            }
        }else if (countRock==0){     // Scissors Win
            notifyAll(Hand.SCISSORS,Hand.PAPER);
        }else if(countScissors==0){  // Paper Win
            notifyAll(Hand.PAPER,Hand.ROCK);
        }else if(countPaper==0){     // Rock Win
            notifyAll(Hand.ROCK,Hand.SCISSORS);
        }else{
            System.err.println("Please email to takigawa if you see this message.");
        }
     //   System.out.println("R:"+countRock+" S:"+countScissors+" P:"+countPaper);
     //   System.out.println();
    }
    public static void main(String[] args) {
        try{
                boolean flag = true;
                while(flag==true){
                int num = Integer.parseInt(args[0]);
                System.out.print("Number of players? ");
                        Scanner scanner = new Scanner(System.in);
                int w = scanner.nextInt();
                JankenPlayer[] players = new JankenPlayer[w];
                for(int k=0;k<w;k++){
                        players[k] = new RandomJankenPlayer("Player"+(k+1));
                }
                Judge3 judge = new Judge3("Sato");
                judge.setPlayers(players);
                for(int i=0; i<num; i++){
                        judge.play();
                }

                players[0].report();
```

```
78                  }
79          }catch(Exception e){
80              System.out.println("this requires an integer argument.");
81              e.printStackTrace();
82          }
83      }
84  }
```

Judge2.java の console output (n=100 の場合):

Number of players? 3

There were 38 draws

Number of players? 5

There were 63 draws

Number of players? 7

There were 78 draws

Number of players? 9

There were 91 draws