

```

1 public class DeskPhone implements Callable {
2     private int myNumber;
3     private boolean isRinging;
4
5     public DeskPhone() {
6         this(0,false);
7     }
8     public DeskPhone(int myNumber, boolean isRinging) {
9         this.myNumber = myNumber;
10        this.isRinging = isRinging;
11    }
12
13    public int getMyNumber() {
14        return myNumber;
15    }
16
17    public void setMyNumber(int myNumber) {
18        this.myNumber = myNumber;
19    }
20
21    public boolean isRinging() {
22        return isRinging;
23    }
24
25    public void setRinging(boolean ringing) {
26        isRinging = ringing;
27    }
28
29    @Override
30    public void powerOn() {
31        System.out.println("power on Desk phone.");
32    }
33
34    @Override
35    public void dial(int phoneNumber) {
36        setRinging(true);
37        setMyNumber(phoneNumber);
38    }
39
40    @Override
41    public void answer() {
42        if(this.isRinging){
43            System.out.println("Hello, It's desk phone(" + getMyNumber() + ").");
44        }else{
45            System.out.println("Sorry, It's power off.");
46        }
47    }
48
49    @Override
50    public boolean callPhone(int phoneNumber) {
51        return this.myNumber == phoneNumber;
52    }
53
54    @Override
55    public String toString() {
56        return "DeskPhone{" +
57            "myNumber=" + myNumber +
58            ", isRinging=" + isRinging +
59            '}';
60    }
61 }
62

```

```
1 public class MobilePhone implements Callable{
2     private int myNumber;
3     private boolean isRinging;
4
5     public MobilePhone() {
6         this(0,false);
7     }
8
9     public MobilePhone(int myNumber, boolean isRinging) {
10         this.myNumber = myNumber;
11         this.isRinging = isRinging;
12     }
13
14     public int getMyNumber() {
15         return myNumber;
16     }
17
18     public void setMyNumber(int myNumber) {
19         this.myNumber = myNumber;
20     }
21
22     public boolean isRinging() {
23         return isRinging;
24     }
25
26     public void setRinging(boolean ringing) {
27         isRinging = ringing;
28     }
29
30     @Override
31     public void powerOn() {
32         System.out.println("power on Mobile phone.");
33     }
34
35     @Override
36     public void dial(int phoneNumber) {
37         setRinging(true);
38         setMyNumber(phoneNumber);
39     }
40
41     @Override
42     public void answer() {
43         if(this.isRinging){
44             System.out.println("Hello, It's mobile phone(" + getMyNumber() + ").");
45         }else{
46             System.out.println("Sorry, It's power off.");
47         }
48     }
49
50     @Override
51     public boolean callPhone(int phoneNumber) {
52         return this.myNumber == phoneNumber;
53     }
54
55     @Override
56     public String toString() {
57         return "MobilePhone{" +
58             "myNumber=" + myNumber +
59             ", isRinging=" + isRinging +
60             '}';
61     }
62 }
63
```

```
1 public interface Callable {  
2     void powerOn(); //утсаа асаах  
3     void dial(int phoneNumber); //дугаар луу залгах  
4     void answer(); //дүүдлагад хариу өгөх  
5     boolean callPhone(int phoneNumber); //өгөгдсөндугаар луу залгагдаж буй эсэх  
6     boolean isRinging(); //утас дуугарч байгаа эсэх  
7 }  
8
```

```
1 public abstract class Creature {
2     int x;
3     int y;
4     int age;
5
6     public Creature(int x, int y, int age) {
7         this.x = x;
8         this.y = y;
9         this.age = age;
10    }
11
12    public void age(){
13        age++;
14    }
15
16    public void move(int xDistance){
17        x += xDistance;
18    }
19
20    public int getX() {
21        return x;
22    }
23
24    public void setX(int x) {
25        this.x = x;
26    }
27
28    public int getY() {
29        return y;
30    }
31
32    public void setY(int y) {
33        this.y = y;
34    }
35
36    public int getAge() {
37        return age;
38    }
39
40    public void setAge(int age) {
41        this.age = age;
42    }
43 }
44
```

```
1 public abstract class Human extends Creature implements Talkable{
2     public Human(int x, int y, int age) {
3         super(x, y, age);
4     }
5
6     public abstract void attack();
7
8     @Override
9     public void talk() {
10         System.out.println("Hello, It's me.");
11     }
12 }
13
```

```
1 public abstract class Animal extends Creature implements Talkable{
2     public Animal(int x, int y, int age) {
3         super(x, y, age);
4     }
5
6     public abstract void attack();
7 }
8
```

```
1 public interface Talkable {  
2     void talk();  
3 }  
4
```

```

1 public class Kevin extends Human implements Swimbable, Programmer{
2
3     double swimming = 0;
4     boolean solveProblem;
5     boolean writingCode;
6
7     public Kevin(int x, int y, int age, boolean solveProblem, boolean writingCode) {
8         super(x, y, age);
9         this.solveProblem = solveProblem;
10        this.writingCode = writingCode;
11    }
12
13    public double getSwimming() {
14        return swimming;
15    }
16
17    public void setSwimming(double swimming) {
18        this.swimming = swimming;
19    }
20
21    public boolean isSolveProblem() {
22        return solveProblem;
23    }
24
25    public void setSolveProblem(boolean solveProblem) {
26        this.solveProblem = solveProblem;
27    }
28
29    public boolean isWritingCode() {
30        return writingCode;
31    }
32
33    public void setWritingCode(boolean writingCode) {
34        this.writingCode = writingCode;
35    }
36
37    @Override
38    public void attack() {
39        System.out.println("Fight!!!");
40    }
41
42    @Override
43    public void swim(int xDistance) {
44        swimming += xDistance;
45    }
46
47    @Override
48    public void swimMove(int xDistance, int yDistance) {
49        swimming += Math.sqrt(xDistance*xDistance + yDistance*yDistance);
50    }
51
52    @Override
53    public void solveProblem() {
54        if(solveProblem){
55            System.out.println("Solved problem.");
56        }
57    }
58
59    @Override
60    public void writingCode() {
61        if(writingCode){
62            System.out.println("finished.");
63        }
64    }
65
66    @Override
67    public String toString() {
68        return "Kevin{" +
69            "x=" + x +
70            ", y=" + y +

```



```
71         ", age=" + age +  
72         ", swimming=" + swimming +  
73         ", solveProblem=" + solveProblem +  
74         ", writingCode=" + writingCode +  
75         '}';  
76     }  
77 }  
78
```

```
1 public class Turtle extends Animal implements Swimable{
2
3     double swimming = 0;
4
5     public Turtle(int x, int y, int age) {
6         super(x, y, age);
7     }
8
9     public double getSwimming() {
10         return swimming;
11     }
12
13     public void setSwimming(double swimming) {
14         this.swimming = swimming;
15     }
16
17     @Override
18     public void attack() {
19         System.out.println("Bite!!!");
20     }
21
22     @Override
23     public void swim(int xDistance) {
24         swimming += xDistance;
25     }
26
27     @Override
28     public void swimMove(int xDistance, int yDistance) {
29         swimming += Math.sqrt(xDistance * xDistance + yDistance*yDistance);
30     }
31
32     @Override
33     public void talk() {
34         System.out.println("Aun aun");
35     }
36
37     @Override
38     public String toString() {
39         return "Turtle{" +
40             "x=" + x +
41             ", y=" + y +
42             ", age=" + age +
43             ", swimming=" + swimming +
44             '}';
45     }
46 }
47
```

```
1 public class Pigeon extends Animal implements Flyable{
2
3     double flying = 0;
4
5     public Pigeon(int x, int y, int age) {
6         super(x, y, age);
7     }
8
9     public double getFlying() {
10         return flying;
11     }
12
13     public void setFlying(double flying) {
14         this.flying = flying;
15     }
16
17     @Override
18     public void attack() {
19         System.out.println("Peck!!!");
20     }
21
22     @Override
23     public void fly(int yDistance) {
24         flying += yDistance;
25     }
26
27     @Override
28     public void flyMove(int xDistance, int yDistance) {
29         flying += Math.sqrt(xDistance*xDistance + yDistance*yDistance);
30     }
31
32     @Override
33     public void talk() {
34         System.out.println("Guyan guyan");
35     }
36
37     @Override
38     public String toString() {
39         return "Pigeon{" +
40             "x=" + x +
41             ", y=" + y +
42             ", age=" + age +
43             ", flying=" + flying +
44             '}';
45     }
46 }
47
```

```
1 public interface Programmer {  
2     void solveProblem();  
3     void writingCode();  
4 }  
5
```

```
1 public interface Swimbable {  
2     void swim(int xDistance);  
3     void swimMove(int xDistance, int yDistance);  
4 }  
5
```

```
1 public interface Flyable {  
2     void fly(int yDistance);  
3     void flyMove(int xDistance, int yDistance);  
4 }  
5
```

```

1 import java.util.Scanner;
2
3 public class MainApp {
4     MainApp(int probNum){
5         switch (probNum){
6             case 1: {
7                 System.out.println("Problem01: ");
8                 testCalllable();
9                 break;
10            }
11            case 2: {
12                System.out.println("Problem02: ");
13                testAbstractClass();
14                break;
15            }
16            default:{
17                System.out.println("Sorry, problem not found!");
18                break;
19            }
20        }
21    }
22
23    static Scanner input(String str){
24        System.out.print(str);
25        return new Scanner(System.in);
26    }
27
28    public static void main(String[] args) {
29        while(true){
30            int probNum = input("Enter problem number (if stop it, enter 0): ").nextInt();
31            if(probNum == 0){
32                break;
33            }
34            new MainApp(probNum);
35        }
36    }
37
38    void testCalllable(){
39        /*
40         * Ширээний утас объект үүсгэнэ
41         * Гар утас объект үүсгэнэ
42         * Утас тус бүрийг асаана.
43         * Утас тус бүрээс өгөгдсөн дугаар луу дуудлага хийнэ
44         * Утас бүр дуудлагад хариу өгнө
45         */
46        DeskPhone deskPhone = new DeskPhone();
47        MobilePhone mobilePhone = new MobilePhone();
48        deskPhone.powerOn();
49        mobilePhone.powerOn();
50        deskPhone.dial(input("Enter desk phone number: ").nextInt());
51        mobilePhone.dial(input("Enter mobile phone number: ").nextInt());
52        deskPhone.answer();
53        mobilePhone.answer();
54    }
55
56    void testAbstractClass(){
57        Kevin kevin = new Kevin(input("Enter x of Kevin: ").nextInt(), input("Enter y of Kevin: ").
58        nextInt(), input("Enter age of Kevin: ").nextInt(), true, true);
59        Turtle turtle = new Turtle(input("Enter x of Turtle: ").nextInt(), input("Enter y of Turtle: ").
60        nextInt(), input("Enter age of Turtle: ").nextInt());
61        Pigeon pigeon = new Pigeon(input("Enter x of Pigeon: ").nextInt(), input("Enter y of Pigeon: ").
62        nextInt(), input("Enter age of Pigeon: ").nextInt());
63        kevin.getSwimming();
64        kevin.swim(input("Enter swim distance on x of Kevin: ").nextInt());
65        kevin.talk();
66        kevin.attack();
67        kevin.writingCode();
68        kevin.toString();
69        turtle.getSwimming();
70        turtle.attack();

```

```
68         turtle.swimMove(input("Enter swim distance on x of Turtle: ").nextInt(), input("Enter swim
    distance on y of Turtle: ").nextInt());
69         turtle.toString();
70         pigeon.getFlying();
71         pigeon.flyMove(input("Enter fly distance on x of Pigeon: ").nextInt(), input("Enter fly
    distance on x of Pigeon: ").nextInt());
72         pigeon.attack();
73         pigeon.talk();
74         pigeon.toString();
75     }
76 }
77
```



Enter problem number (if stop it, enter 0): 1

Problem01:

power on Desk phone.

power on Mobile phone.

Enter desk phone number: 99206304

Enter mobile phone number: 99650592

Hello, It's desk phone(99206304).

Hello, It's mobile phone(99650592).

Enter problem number (if stop it, enter 0): 2

Problem02:

Enter x of Kevin: 12

Enter y of Kevin: 2

Enter age of Kevin: 23

Enter x of Turtle: 34

Enter y of Turtle: 100

Enter age of Turtle: 56

Enter x of Pigeon: 100

Enter y of Pigeon: 1000

Enter age of Pigeon: 3

Enter swim distance on x of Kevin: 5

Hello, It's me.

Fight!!!

finished.

Bite!!!

Enter swim distance on x of Turtle: 6

Enter swim distance on y of Turtle: 3

Enter fly distance on x of Pigeon: 6

Enter fly distance on x of Pigeon: 7

Peck!!!

Guyan guyan

Enter problem number (if stop it, enter 0): 0

Process finished with exit code 0

!