Môn học lý thuyết ngôn ngữ hướng đối tượng

Bài thực hành số 8: Tổng hợp về các kỹ thuật xây dựng lớp và tái sử dụng mã nguồn

Bài 1) Cho biết kết quả khi thực hiện các đoạn mã sau:

a) Chương trình 1

```
public class ValuesTest {
        int[] ia = new int[1];
        boolean b;
        int i;
        Object o;
        public static void main(String[] args) {
                ValuesTest instance = new ValuesTest();
                instance.print();
        }
        public void print() {
                System.out.println(ia[0]+" "+b+" "+i+" "+o);
        }
}
```

Trả lời: ……

……..

……..

b) Chương trình 2

```
public class MyClass {
        long var;
        public void MyClass(long param) { var = param; }       // (1)
        public static void main(String[] args) {
                MyClass a, b;
                a = new MyClass();                    // (2)
                b = new MyClass(5);                   // (3)
        }
}
```

Trả lời: ……

……..

……..

c) Chương trình 3

```
import java.util.*;

package com.cnpm.hust;
```

```
public class AClass {
   public Other anInstance;
}
class Other {
   int value;
}
```

Trả lời: ……

……..

……..

d) Chương trình 4

```
public class Assignment {
   public static void main(String[] args) {
      int a, b, c;
      b = 10;
      a = b = c = 20;
      System.out.println(a);
   }
}
```

Trả lời: ……

……..

……..

e) Chương trình 5

```
public class MyClass {
   public static void main(String[] args) {
      String a, b, c;
      c = new String("mouse");
      a = new String("cat");
      b = a;
      a = new String("dog");
      c = b;

      System.out.println(c);
   }
}
```

Trả lời: ……

……..

……..

f) Chương trình 6

```
public class Prog1 {
   public static void main(String[] args) {
      int k = 1;
      int i = ++k  +  k++  +  +k;
      System.out.println(i);
```

```
    }
}
```
Trả lời: ……
……..
……..

g) Chương trình 7

```
public class EvaluationOrder {
    public static void main(String[] args) {
        int[] array = { 4, 8, 16 };
        int i=1;
        array[++i] = --i;
        System.out.println(array[0] + array[1] + array[2]);
    }
}
```
Trả lời: ……
……..
……..

h) Chương trình 8

```
public class ParameterPass {
    public static void main(String[] args) {
        int i = 0;
        addTwo(i++);
        System.out.println(i);
    }

    static void addTwo(int i) {
        i += 2;
    }
}
```
Trả lời: ……
……..
……..

i) Chương trình 9

```
class MyClass {
    public static void main(String[] args) {
        String str1 = "str1";
        String str2 = "str2";
        String str3 = "str3";

        str1.concat(str2);
        System.out.println(str3.concat(str1));
    }
}
```

Trả lời: ……
……..
……..

j) Chương trình 10

```
public class RefEq {
   public static void main(String[] args) {
      String s = "ab" + "12";
      String t = "ab" + 12;
      String u = new String("ab12");
      System.out.println((s==t) + " " + (s==u));
   }
}
```

Trả lời: ……
……..
……..

k) Chương trình 11

```
public class MyClass {
   public static void main(String[] args) {
      String s = "hello";
      StringBuffer sb = new StringBuffer(s);
      sb.reverse();
      if (s == sb) System.out.println("a");
      if (s.equals(sb)) System.out.println("b");
      if (sb.equals(s)) System.out.println("c");
   }
}
```

Trả lời: ……
……..
……..

l) Chương trình 12

```
public class StringMethods {
   public static void main(String[] args) {
      String str = new String("eenny");
      str.concat(" meeny");
      StringBuffer strBuf = new StringBuffer(" miny");
      strBuf.append(" mo");
      System.out.println(str + strBuf);
   }
}
```

Trả lời: ……
……..
……..

m) Chương trình 13

```
public class MyClass {
   public static void main(String[] args) {
      jack();
      jill();
   }
   public static void jack(){
      String s1 = "hill5";
      String s2 = "hill" + "5";
      System.out.println(s1==s2);
   }
   public static void jill(){
      String s1 = "hill5";
      String s2 = "hill" + s1.length();
      System.out.println(s1==s2);
   }
}
```

Trả lời: ……

……..

……..

n) Chương trình 14

```
public class Puzzle {
    public static void main(String... args) {
            System.out.println("Hi Guys!");
            Character myChar = new Character('\u000d');
  }
}
```

Trả lời: ……

……..

……..

o) Chương trình 15 (soạn thảo nguyên dạng đoạn mã nguồn dưới đây, bao gồm cả phần comment):

```
/**
 * Version 1.0
 * Path D:\Test\units\Bai04\
 * July 17, 2015
 */

public class Test {
   public static void main(String[] args) {
      System.out.print("Hell");
      System.out.println("o world");
   }
}
```

Trả lời: ……
……..
……..

Bài 2)

Viết chương trình mô phỏng quá trình xử lý cửa soát vé tự động tại mỗi ga của một tuyến đường sắt bao gồm 4 ga: A, B, C và D; với A là điểm đầu và D là điểm cuối. Trong tuyến này, có 2 loại vé có thể sử dụng: Vé một chiều (OneWayTicket, sau đây gọi là vé) và thẻ trả trước (PrepaidCard, sau đây gọi là thẻ).

Vé trên tuyến đường sắt được quyết định dựa trên khoảng cách giữa các ga. Giá vé cho khoảng cách <= 6 km là 9.000 đồng (giá vé này được gọi là giá sàn). Cứ thêm <= 2 km thì giá vé tăng thêm một lượng là 2000 đồng. Ví dụ nếu khoảng cách là 9 km thì giá vé là 9.000 + 2*2000 = 13.000 (đồng).

Việc lên tàu tại ga xuất phát được ghi lại trên vé khi hành khách đi qua cửa soát vé tự động của ga đó để vào khu vực sân ga. Khi hành khách đi qua cửa soát vé tự động của ga đến để rời khu vực sân ga để vào ga đến, giá vé được tính toán dựa trên ga xuất phát và ga đến. Nếu lượng tiền đã trả cho vé không đủ thì cửa được đóng lại ngăn hành khách đó rời khỏi khu vực sân ga. Trong tuyến đường sắt này, hành khách có thể vào khu vực sân ga qua cửa soát vé tự động tại bất kỳ ga nào, tức là có thể lên tàu dù vé được xuất ở ga nào. Ví dụ, vé được xuất ở ga A, hành khách có thể vào khu vực sân ga thông qua cửa soát vé tự động trong ga B. Khi vé đã sử dụng, nó sẽ không được sử dụng nữa, dù lượng tiền có trong vé lớn hơn giá vé tính toán.

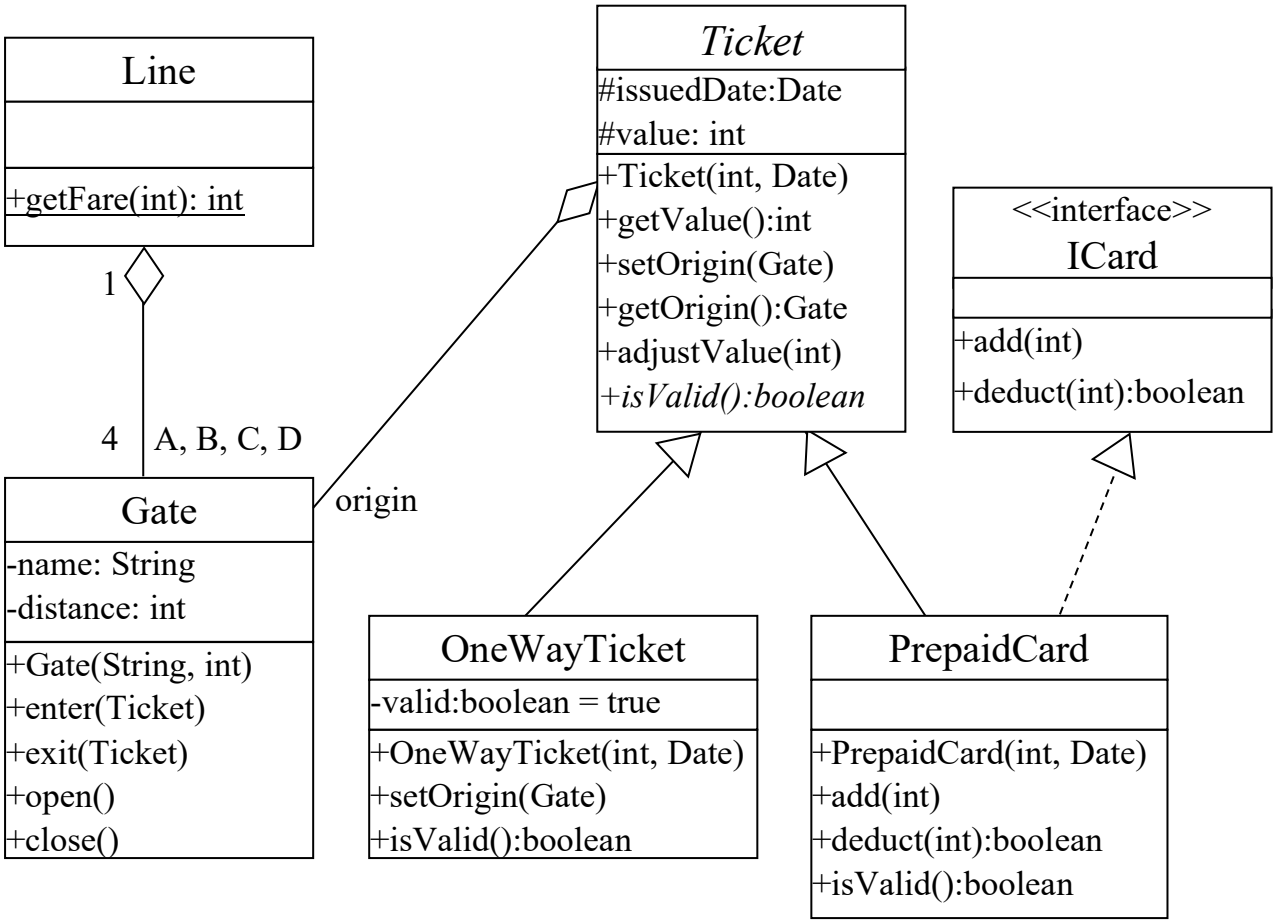
Việc lên tàu tại ga xuất phát được ghi lại trên thẻ khi hành khách vào khu vực sân ga thông qua cửa soát vé tự động của ga đó. Lúc này, nếu số tiền có trong thẻ < 9000 đồng thì cửa sẽ đóng để không cho hành khách đó vào khu vực sân ga. Khi hành khách rời khu vực sân ga qua cửa soát vé tự động của ga đến, giá vé được tính toán và trừ vào số dư tài khoản có trong thẻ khi hành khách đi qua. Lúc này, nếu số dư tài khoản nhỏ hơn giá vé thì cửa đóng lại không cho hành khách đi qua để rời khỏi khu vực sân ga.

Viết các lớp sau, theo gợi ý trên hình vẽ dưới đây:

a. Lớp `Gate` biểu thị cửa soát vé tự động.
    i. Phương thức khởi tạo của lớp `Gate` khởi tạo cho tên Ga (A, B, C hay D) và khoảng cách của ga đó tính từ ga đầu A.
    ii. Phương thức `enter` thực hiện khi hành khách bước vào khu vực sân ga thông qua cửa soát vé tự động của ga xuất phát. Nếu vé hoặc thẻ không hợp lệ thì cửa đóng; ngược lại nếu hành khách vào bình thường, thông tin về việc lên tàu được ghi lại trong vé hoặc thẻ.

iii. Phương thức `exit` thực hiện khi hành khách rời khỏi khu vực sân ga thông qua cửa soát vé tự động của ga đến. Nếu giá trị của vé hoặc thẻ (`value`) không đủ thì cửa đóng.

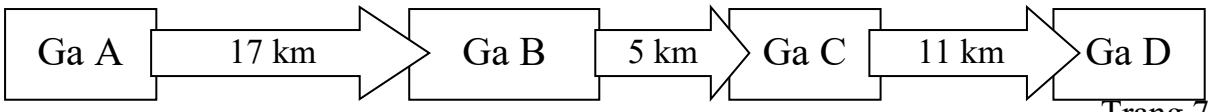
iv. Phương thức `open` và `close` hiển thị các thông báo về việc mở hoặc đóng cửa tương ứng.



b. Lớp `Line` thể hiện tuyến đường sắt.

i. Phương thức `getFare` tính toán giá vé dựa trên khoảng cách được đưa vào, trả về giá vé sau khi tính toán.

ii. Mối quan hệ giữa `Line` và `Gate` được cài đặt bằng các thành viên hằng thuộc kiểu `Gate` (tương ứng với các cửa soát vé tự động của Ga A, B, C và D) trong lớp `Line` được khởi tạo với khoảng cách như sau (chú ý ga A là điểm đầu được khởi tạo khoảng cách là 0 km; một ga có thể đi đến các ga theo cả 2 chiều, ví dụ Ga A có thể đến Ga B và Ga B có thể đi đến Ga A):



c. Lớp trừu tượng `Ticket` biểu diễn vé/thẻ.
    i. Phương thức `setOrigin` ghi lại ga xuất phát khi hành khách đi qua cửa soát vé tự động của ga để vào khu vực sân ga.
    ii. Phương thức `getOrigin` trả về ga xuất phát, nếu ga chưa được ghi lại (hành khách chưa vào khu vực sân ga) thì trả về `null`.
d. Giao diện `ICard` với 2 phương thức `add` (cộng thêm tiền vào thẻ) và `deduct` (trừ một khoản tiền khỏi thẻ).
e. Lớp `OneWayTicket` biểu diễn vé lên tàu, lớp `PrepaidCard` biểu diễn thẻ lên tàu. Hai lớp đều cần ghi đè các phương thức kế thừa từ lớp `Ticket`, lớp `PrepaidCard` định nghĩa chi tiết cho 2 phương thức của giao diện `ICard`.
f. Lớp `Test` trong đó:
    i. Tạo ra các thể hiện của lớp `OneWayTicket` (1 trường hợp vé không hợp lệ - cửa đóng tại ga xuất phát, 1 trường hợp vé không đủ tiền - cửa đóng tại cửa soát vé ga đến, 1 trường hợp vé đủ tiền và lên được tàu) và các thể hiện của các lớp khác để minh họa việc lên tàu từ cửa soát vé của ga này đến cửa soát vé của ga kia.
    ii. Tạo ra các thể hiện của lớp `Prepaid` (1 trường hợp thẻ có số tiền không đủ mức sàn - cửa đóng tại ga xuất phát, 1 trường hợp thẻ không đủ tiền để lên tàu lên ga đến - cửa đóng tại cửa soát vé ga đến, 1 trường hợp thẻ đủ tiền và lên được tàu) và các thể hiện của các lớp khác để minh họa việc lên tàu từ cửa soát vé của ga này đến cửa soát vé của ga kia.