

Chương 3

Class và Object

Mục tiêu:

Kết thúc bài tập này, sinh viên có khả năng:

- ✓ Xây dựng một lớp hoàn chỉnh
- ✓ Biết cách tạo đối tượng từ lớp đã xây dựng
- ✓ Biết cách truy xuất các thuộc tính và phương thức bên trong lớp
- ✓ Biết cách tư duy theo hướng đối tượng để giải quyết vấn đề

Bài tập:

Sử dụng phương pháp hướng đối tượng hãy giải quyết các bài toán sau đây:

(Ghi chú: Sinh viên lập trình trên IDE rồi đưa source code vào bài làm)

Bài 1 (1 điểm)

Tạo lớp SanPham gồm 3 thuộc tính là tên, giá và tỉ lệ giảm giá. Lớp cũng bao gồm 2 phương thức là tính thuế nhập khẩu (thuế nhập khẩu = 10% giá trị sản phẩm) và xuất thông tin ra màn hình. Phương thức xuất ra màn hình cần thể hiện các thông tin sau:

- Tên sản phẩm
- Đơn giá
- Tỉ lệ giảm giá
- Thuế nhập khẩu

Chương trình:

```
package assignment3;

import java.util.Scanner;

class Product {
    String name;
    double price;
    double saleOffRate;
    double tax;

    void inputProduct() {
        Scanner input = new Scanner(System.in);
        System.out.print("Nhập tên sản phẩm: ");
        name = input.nextLine();

        System.out.print("Nhập giá sản phẩm: ");
        price = input.nextDouble();

        System.out.print("Nhập tỉ lệ giảm giá: ");
        saleOffRate = input.nextDouble();
    }
}
```



```
        input.close();
    }

    void outputProduct() {
        System.out.println("Tên sản phẩm: " + name);
        System.out.println("Giá trị sản phẩm: " + price);
        System.out.println("Tỉ lệ giảm giá: " + saleOffRate);
        System.out.println("Thuế nhập khẩu: " + tax);
    }

    void calculateTax() {
        tax = price * 10.0 / 100.0;
    }

    public static void main() {
        Product product = new Product();
        product.inputProduct();
        product.calculateTax();
        product.outputProduct();
    }
}
```

Bài 2 (5 điểm)

Tạo các lớp biểu diễn các đối tượng hình học tương ứng: hình vuông, hình tròn, hình chữ nhật, hình tam giác. Xuất thông tin và tính chu vi, diện tích của các hình đó

Hướng dẫn: sinh viên có thể sử dụng công thức Heron để tính diện tích dựa vào độ dài 3 cạnh của tam giác:

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

Trong đó: $p = \frac{1}{2}(a+b+c)$

Chương trình:

2.1 Lớp hình vuông

```
package assignment3;

import java.util.Scanner;

class Square {
    int edge; // Độ dài cạnh
    int perimeter; // Chu vi
    int area; // Diện tích

    void inputSquare() {
        Scanner input = new Scanner(System.in);
        do {
            System.out.print("Nhập độ dài cạnh hình vuông: ");
            edge = input.nextInt();
        } while (edge < 0);
        input.close();
    }

    // Phương thức tính chu vi
    void calcPerimeter() {
        perimeter = edge * 2;
    }

    // Phương thức tính diện tích
    void calcArea() {
        area = edge * edge;
    }

    void outputSquare() {
        System.out.println("Chu vi hình vuông: " + perimeter);
        System.out.println("Diện tích hình vuông: " + area);
    }

    public static void main() {
        Square square = new Square();
        square.inputSquare();
        square.calcPerimeter();
        square.calcArea();
        square.outputSquare();
    }
}
```

2.2 Lớp hình tròn

```
package assignment3;

import java.util.Scanner;

class Circle {
    int radius; // Bán kính
    final double pi = 3.14; // Số PI
    double perimeter; // Chu vi
    double area; // Diện tích

    void inputCircle() {
        Scanner input = new Scanner(System.in);
        do {
            System.out.print("Nhập bán kính hình tròn: ");
            radius = input.nextInt();
        } while (radius < 0);
        input.close();
    }

    // Phương thức tính chu vi
    void calcPerimeter() {
        perimeter = radius * 2 * pi;
    }

    // Phương thức tính diện tích
    void calcArea() {
        area = radius * radius * pi;
    }

    void outputCircle() {
        System.out.println("Chu vi hình tròn: " + perimeter);
        System.out.println("Diện tích hình tròn: " + area);
    }

    public static void main() {
        Circle circle = new Circle();
        circle.inputCircle();
        circle.calcPerimeter();
        circle.calcArea();
        circle.outputCircle();
    }
}
```

2.3 Lớp hình chữ nhật

```
package assignment3;

import java.util.Scanner;

class Rectangle {
    int length; // chiều dài
    int width; // chiều rộng
    double perimeter; // chu vi
    double area; // diện tích

    void inputRectangle() {
        Scanner input = new Scanner(System.in);
        do {
            System.out.print("Nhập chiều dài: ");
            length = input.nextInt();
        } while (length < 0);

        do {
            System.out.print("Nhập chiều rộng: ");
            width = input.nextInt();
        } while (width < 0);

        input.close();
    }

    // Phương thức tính chu vi
    void calcPerimeter() {
        perimeter = (length + width) * 2;
    }

    // Phương thức tính diện tích
    void calcArea() {
        area = length * width;
    }

    void outputRectangle() {
        System.out.println("Chu vi hình chữ nhật: " + perimeter);
        System.out.println("Diện tích hình chữ nhật: " + area);
    }

    public static void main() {
        Rectangle rectangle = new Rectangle();
        rectangle.inputRectangle();
        rectangle.calcPerimeter();
        rectangle.calcArea();
        rectangle.outputRectangle();
    }
}
```

2.4 Lớp hình tam giác

```
package assignment3;

import java.util.Scanner;

class Triangle {
    int a; // Độ dài cạnh thứ 1
    int b; // Độ dài cạnh thứ 2
    int c; // Độ dài cạnh thứ 3
    double perimeter; // Chu vi
    double area; // Diện tích

    void inputTriangle() {
        Scanner input = new Scanner(System.in);
        System.out.println("Nhập độ dài các cạnh: ");
        do {
            System.out.print("a = ");
            a = input.nextInt();
        } while (a < 0);

        do {
            System.out.print("b = ");
            b = input.nextInt();
        } while (b < 0);

        do {
            System.out.print("c = ");
            c = input.nextInt();
        } while (c < 0);

        input.close();
    }

    // Phương thức tính chu vi
    void calcPerimeter() {
        perimeter = a + b + c;
    }

    // Phương thức tính diện tích
    void calcArea() {
        calcPerimeter();

        // Công thức Heron tính diện tích tam giác dựa vào độ dài 3 cạnh
        double p = perimeter / 2;
        area = Math.sqrt(p * (p - a) * (p - b) * (p - c));
    }

    void outputTriangle() {
        System.out.println("Chu vi hình tam giác: " + perimeter);
        System.out.println("Diện tích hình tam giác: " + area);
    }

    public static void main() {
        Triangle triangle = new Triangle();
        triangle.inputTriangle();
        triangle.calcPerimeter();
        triangle.calcArea();
        triangle.outputTriangle();
    }
}
```

```
}  
}
```

Bài 3 (1 điểm)

Tạo lớp Point biểu diễn các điểm trong mặt phẳng tọa độ Oxy. Nhập vào 2 điểm (2 đối tượng điểm). Tính khoảng cách giữa chúng và xuất ra kết quả.

Hướng dẫn: Giả sử có 2 điểm P1 (x1, y1) và P2 (x2, y2) thì khoảng cách từ P1 đến P2 được tính theo công thức

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Chương trình:

```
package assignment3;  
  
import java.util.Scanner;  
  
class Point {  
    int x;  
    int y;  
  
    void inputPoint() {  
        Scanner input = new Scanner(System.in);  
        System.out.println("Nhập tọa độ điểm: ");  
  
        System.out.print("x = ");  
        x = input.nextInt();  
  
        System.out.print("y = ");  
        y = input.nextInt();  
    }  
  
    // Phương thức tính khoảng cách 2 điểm với tham số là một đối tượng điểm  
    double calcDistance(Point p2) {  
        double distance = Math.sqrt(Math.pow(p2.x - x, 2) + Math.pow(p2.y - y,  
2));  
        return distance;  
    }  
  
    void outputPoint() {  
        System.out.println("(" + x + ", " + y + ")");  
    }  
  
    public static void main(String[] args) {  
        Point point1 = new Point();  
        point1.inputPoint();  
        point1.outputPoint();  
  
        Point point2 = new Point();  
        point2.inputPoint();  
        point2.outputPoint();  
  
        System.out.println("Khoảng cách giữa 2 điểm trên: " +  
point1.calcDistance(point2));  
    }  
}
```

```
}  
}
```

Bài 4 (1 điểm)

Viết chương trình nhập vào 1 phân số.

a/ Hãy cho biết phân số đó là phân số âm hay phân số dương hay bằng 0

b/ Hãy rút gọn phân số đó về phân số tối giản

Chương trình:

```
package assignment3;  
  
import java.util.*;  
  
public class PhanSo {  
    // Thành phần dữ liệu  
    int tuSo;  
    int mauSo;  
    int reducedTuSo; // Tử số sau khi rút gọn  
    int reducedMauSo; // Mẫu số sau khi rút gọn  
    String type; // Loại phân số  
    int ucln; // ước chung lớn nhất  
  
    // Thành phần xử lý  
    public PhanSo() {  
        tuSo = 1;  
        mauSo = 1;  
        reducedTuSo = 1;  
        reducedMauSo = 1;  
        type = "";  
        ucln = 1;  
    }  
  
    void inputPhanSo() {  
        Scanner input = new Scanner(System.in);  
        System.out.print("Nhập vào tử số: ");  
        tuSo = input.nextInt();  
  
        do {  
            System.out.print("Nhập vào mẫu số: ");  
            mauSo = input.nextInt();  
  
        } while (mauSo == 0);  
    }  
  
    // Phương thức xác định loại phân số  
    void getType() {  
        if (tuSo == 0) type = "không!";  
        else if (tuSo / mauSo >= 0) type = "dương!";  
        else type = "âm!";  
    }  
  
    // Phương thức tìm ước số chung lớn nhất  
    void timUCLN() {  
        int tmpTuSo, tmpMauSo;  
        tmpTuSo = Math.abs(tuSo);
```



```
        tmpMauSo = Math.abs(mauSo);

        if (tmpTuSo == 0) ucln = tmpMauSo;
        else
        {
            while (tmpTuSo != tmpMauSo) {
                if (tmpTuSo > tmpMauSo) tmpTuSo -= tmpMauSo;
                else tmpMauSo -= tmpTuSo;
            }
            ucln = tmpTuSo;
        }
    }

    // Phương thức rút gọn phân số
    void rutGon() {
        reducedTuSo = tuSo / ucln;
        reducedMauSo = mauSo / ucln;
    }

    void outputPhanSo() {
        if (mauSo > 0) {
            System.out.println("Phân số: " + tuSo + "/" + mauSo + " là phân
số" + type);
        } else {
            System.out.println("Phân số: " + tuSo + "/" + mauSo + ") là phân
số" + type);
        }

        if (reducedTuSo != 0) {
            if (reducedMauSo > 0) {
                System.out.println("Phân số tối giản: " + reducedTuSo + "/"
+ reducedMauSo);
            } else {
                System.out.println("Phân số tối giản: " + reducedTuSo +
"/(" + reducedMauSo + ")");
            }
        } else {
            System.out.println("Phân số tối giản: " + reducedTuSo /
reducedMauSo);
        }
    }

}

public static void main(String[] args) {

    // Đối tượng PhanSo bài tập 4
    PhanSo phanSo = new PhanSo();
    phanSo.inputPhanSo();

    // Xác định loại phân số;
    phanSo.getType();

    // Rút gọn phân số
    phanSo.timUCLN();
    phanSo.rutGon();

    // Xuất kết quả ra màn hình
    phanSo.outputPhanSo();
}
```

}

Bài 5 (1 điểm)

Viết chương trình nhập vào 2 phân số. Tìm phân số lớn nhất và xuất ra kết quả

Chương trình:

```
package assignment3;

import java.util.*;

public class PhanSoBai5 {
    // Thành phần dữ liệu
    int tuSo; // Tử số
    int mauSo; // Mẫu số
    int ucln; // Ước số chung lớn nhất

    // Thành phần xử lý
    public PhanSoBai5() {
        tuSo = 1;
        mauSo = 1;
    }

    void inputPhanSo() {
        Scanner input = new Scanner(System.in);
        System.out.print("Nhập vào tử số: ");
        tuSo = input.nextInt();

        do {
            System.out.print("Nhập vào mẫu số: ");
            mauSo = input.nextInt();
        } while (mauSo == 0);
    }

    // Phương thức tìm ước chung lớn nhất
    void timUCLN() {
        int tmpTuSo, tmpMauSo;
        tmpTuSo = Math.abs(tuSo);
        tmpMauSo = Math.abs(mauSo);

        if (tmpTuSo == 0) ucln = tmpMauSo;
        else
        {
            while (tmpTuSo != tmpMauSo) {
                if (tmpTuSo > tmpMauSo) tmpTuSo -= tmpMauSo;
                else tmpMauSo -= tmpTuSo;
            }
            ucln = tmpTuSo;
        }
    }

    // Phương thức rút gọn phân số
    void rutGon() {
        timUCLN();
        tuSo = tuSo / ucln;
        mauSo = mauSo / ucln;
    }
}
```

```
// Phương thức kiểm tra phân số này có lớn hơn phân số kia
boolean isGreaterThan(PhanSoBai5 phanSo2) {
    //Nếu mẫu số của phân số 1 == mẫu số của phân số 2
    if (mauSo == phanSo2.mauSo) {
        return (tuSo >= phanSo2.tuSo);
    } else {
        // Nếu 2 mẫu số khác nhau => đồng bộ hóa mẫu số
        int tmpTuSo1 = tuSo * phanSo2.mauSo;
        int tmpTuSo2 = phanSo2.tuSo * mauSo;
        return (tmpTuSo1 >= tmpTuSo2);
    }
}

public static void main(String[] args) {
    // Nhập vào đối tượng phân số thứ 1
    PhanSoBai5 phanSo1 = new PhanSoBai5();
    phanSo1.inputPhanSo();

    // Nhập vào đối tượng phân số thứ 2
    PhanSoBai5 phanSo2 = new PhanSoBai5();
    phanSo2.inputPhanSo();

    // Rút gọn hai phân số
    phanSo1.rutGon();
    phanSo2.rutGon();

    if (phanSo1.isGreaterThan(phanSo2)) {
        System.out.println("Phân số lớn nhất là: " + phanSo1.tuSo + "/" +
phanSo1.mauSo);
    } else {
        System.out.println("Phân số lớn nhất là: " + phanSo2.tuSo + "/" +
phanSo2.mauSo);
    }
}
```

Bài 6 (4 điểm)

Tạo lớp Date biểu diễn các ngày trong năm. Thuộc tính của lớp Date gồm có ngày, tháng và năm. Cho phép người dùng nhập vào 1 ngày, hãy in ra ngày đứng trước và đứng sau của ngày đó 1 ngày.

Ví dụ: Người dùng nhập vào ngày 28/02/2019

Ngày trước: 27/02/2019

Ngày sau: 01/03/2019

Chương trình:

```
package assignment3;

import java.util.*;

public class Date {
    int date;
    int month;
    int year;

    void inputDate() {
        String stringDate;
        Scanner input = new Scanner(System.in);
        System.out.print("Nhập vào ngày tháng năm: ");
        stringDate = input.nextLine();
        String[] array = stringDate.split("/");
        date = Integer.parseInt(array[0]);
        month = Integer.parseInt(array[1]);
        year = Integer.parseInt(array[2]);
    }

    String getDate() {
        String strDate = (date < 10 ? "0" + date : "" + date);
        String strMonth = (month < 10 ? "0" + month : "" + month);
        return (strDate + "/" + strMonth + "/" + year);
    }

    // Phương thức kiểm tra xem ngày nhập vào có phải là ngày cuối cùng của tháng
    // (Dựa trên bài tập 13 - Tuần 2 (Assignment 2))
    boolean isLastDateInMonth(int date, int month, int year) {
        switch (month) {
            case 1:
            case 3:
            case 5:
            case 7:
            case 8:
            case 10:
            case 12:
                return (date == 31);

            case 2:
                // Tính ra ngày cuối cùng của tháng 2 (xét theo năm Nhuận
                // và năm không Nhuận)
        }
    }
}
```

```
        int finalDateOfFeb = (((year % 4 == 0 && year % 100 != 0)
|| year % 400 == 0) ? 29 : 28);
        return (date == finalDateOfFeb);

        case 4:
        case 6:
        case 9:
        case 11:
            return (date == 30);
    }
    return false;
}

// Phương thức kiểm tra xem tháng nhập vào có phải là tháng cuối cùng của năm
boolean isLastMonthInYear(int month) {
    return (month == 12);
}

// Phương thức kiểm tra xem năm nhập vào là năm Nhuận hay năm thường
boolean isLeapYear(int year) {
    return ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0);
}

// Phương thức tính ngày cuối cùng của một tháng
int finalDateInMonth(int month, int year) {
    int[] datesPerMonth = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31
};
    int[] datesPerMonthLeapYear = { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31,
30, 31 };

    // Nếu năm là năm nhuận
    if (isLeapYear(year)) {
        return datesPerMonthLeapYear[month];
    } else {
        return datesPerMonth[month];
    }
}

// Phương thức trả về ngày đứng sau
String getNextDate() {
    int nextDate, nextMonth, nextYear;

    // Thuật toán tìm ngày đứng sau
    if (isLastDateInMonth(date, month, year)) { // Kiểm tra xem ngày hiện
tại có phải là ngày cuối cùng của tháng và năm
        if (isLastMonthInYear(month)) { // Kiểm tra xem tháng hiện tại có
phải là tháng cuối cùng của năm
            nextDate = 1;
            nextMonth = 1;
            nextYear = year + 1;
        } else {
            nextDate = 1;
            nextMonth = month + 1;
            nextYear = year;
        }
    }
}
```

```
        }
    } else {
        nextDate = date + 1;
        nextMonth = month;
        nextYear = year;
    }

    // Ghép chữ số 0 vào đầu nếu ngày hoặc tháng là số chỉ có 1 chữ số
    String strDate = (nextDate < 10 ? "0" + nextDate : "" + nextDate);
    String strMonth = (nextMonth < 10 ? "0" + nextMonth : "" + nextMonth);

    return (strDate + "/" + strMonth + "/" + nextYear);
}

// Phương thức trả về ngày đứng trước
String getPreviousDate() {
    int preDate, preMonth, preYear;

    // Thuật toán tìm ngày đứng trước
    if (date == 1) { // Kiểm tra xem ngày hiện tại có phải là ngày đầu tiên
của tháng
        if (month == 1) { // Kiểm tra xem tháng hiện tại có phải là tháng
đầu tiên của năm
            preDate = 31;
            preMonth = 12;
            preYear = year - 1;
        } else {
            preDate = finalDateInMonth(month - 1, year);
            preMonth = month - 1;
            preYear = year;
        }
    } else {
        preDate = date - 1;
        preMonth = month;
        preYear = year;
    }

    // Ghép chữ số 0 vào đầu nếu ngày hoặc tháng là số chỉ có 1 chữ số
    String strDate = (preDate < 10 ? "0" + preDate : "" + preDate);
    String strMonth = (preMonth < 10 ? "0" + preMonth : "" + preMonth);

    return (strDate + "/" + strMonth + "/" + preYear);
}

public static void main(String[] args) {
    Date date = new Date();
    date.inputDate();

    System.out.println("Ngày hiện tại: " + date.getDate());
    System.out.println("Ngày trước đó: " + date.getPreviousDate());
    System.out.println("Ngày tiếp theo: " + date.getNextDate());
}
}
```