



Chương 5

Inheritance

Mục tiêu:

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

- ✓ Xây dựng được ứng dụng hoặc chương trình Java có cấu trúc gồm nhiều lớp được tổ chức theo sự phân cấp kế thừa
- ✓ Biết cách tái sử dụng mã nguồn
- ✓ Biết cách vận dụng linh hoạt mối quan hệ kế thừa hay hợp thành cho việc thiết kế lớp
- ✓ Biết cách vận dụng tầm vực truy xuất vào việc thiết kế lớp và chia sẻ dữ liệu giữa các lớp thông qua hình thức kế thừa

Bài tập:

Sử dụng phương pháp lập trình hướng đối tượng, các em hãy giải quyết những bài tập sau đây:

Bài 1 (5 điểm)

Hãy viết chương trình xây dựng một lớp Time để biểu diễn giờ phút giây của một thời điểm. Lớp này gồm có các phương thức như mô tả dưới đây:

- Cho phép **thiết lập giá trị tự động** là 00:00:00 **mỗi khi người dùng tạo ra một đối tượng Time mới** mà không chỉ định rõ giờ phút giây của đối tượng này.
- Cho phép thiết lập giá trị là hh:mm:00 mỗi khi người dùng tạo ra một đối tượng Time mới với các tham số về giờ (hh) và phút (mm) được truyền vào
- Cho phép thiết lập giá trị là hh:mm:ss mỗi khi người dùng tạo ra một đối tượng Time mới với các tham số về giờ (hh), phút (mm) và giây (ss) được truyền vào
- Cho phép lấy giá trị (getter) và thay đổi giá trị (setter) của từng thuộc tính thành viên giờ phút giây trong lớp

Yêu cầu: hãy viết thêm các phương thức:

- incrementHour(int value): cho phép tăng số giờ lên value đơn vị và điều chỉnh lại giờ phút giây cho phù hợp
- incrementMinute(int value): cho phép tăng số phút lên value đơn vị và điều chỉnh lại giờ phút giây cho phù hợp
- incrementSecond(int value): cho phép tăng số giây lên value đơn vị và điều chỉnh lại giờ phút giây cho phù hợp

- `outputTime()`: in ra màn hình giờ phút giây của đối tượng theo format 'hh:mm:ss'

Bài làm:

```
package assignment4;

// Tham khảo cách tính: https://www.mathsisfun.com/time-add-subtract.html
public class Time {
    private int hh; // Giờ
    private int mm; // Phút
    private int ss; // Giây

    public Time() {
        this.hh = 0;
        this.mm = 0;
        this.ss = 0;
    }

    public Time(int hh, int mm) {
        this.hh = hh;
        this.mm = mm;
        this.ss = 0;
    }

    public Time(int hh, int mm, int ss) {
        this.hh = hh;
        this.mm = mm;
        this.ss = ss;
    }

    public int getHour() {
        return this.hh;
    }

    public void setHour(int hh) {
        this.hh = hh;
    }

    public int getMinute() {
        return this.mm;
    }

    public void setMinute(int mm) {
        this.mm = mm;
    }

    public int getSecond() {
        return this.ss;
    }

    public void setSecond(int ss) {
        this.ss = ss;
    }

    public void outputTime() {
        String strHour = (hh < 10 ? "0" + hh : "" + hh);
        String strMinute = (mm < 10 ? "0" + mm : "" + mm);
        String strSecond = (ss < 10 ? "0" + ss : "" + ss);
    }
}
```

```
        System.out.println(strHour + ":" + strMinute + ":" + strSecond);
    }

    // Phương thức tăng số giờ thêm 1 giá trị nguyên (có thể là giá trị dương hoặc
âm)
    public void incrementHour(int value) {
        if (value == 0) return;
        this.hh = (this.hh + value) % 24;

        // Nếu số giờ bé hơn 0 thì lấy số giờ cộng thêm 24
        if (this.hh < 0) this.hh = this.hh + 24;

        // Số phút và giây vẫn giữ nguyên
    }

    // Phương thức tăng số phút thêm 1 giá trị nguyên (có thể là giá trị dương hoặc
âm)
    public void incrementMinute(int value) {
        if (value == 0) return;

        // Cách 1:
        this.mm = this.mm + value;
        while (this.mm < 0) {
            this.mm += 60;
            incrementHour(-1);
        }
        while (this.mm > 59) {
            this.mm -= 60;
            incrementHour(1);
        }

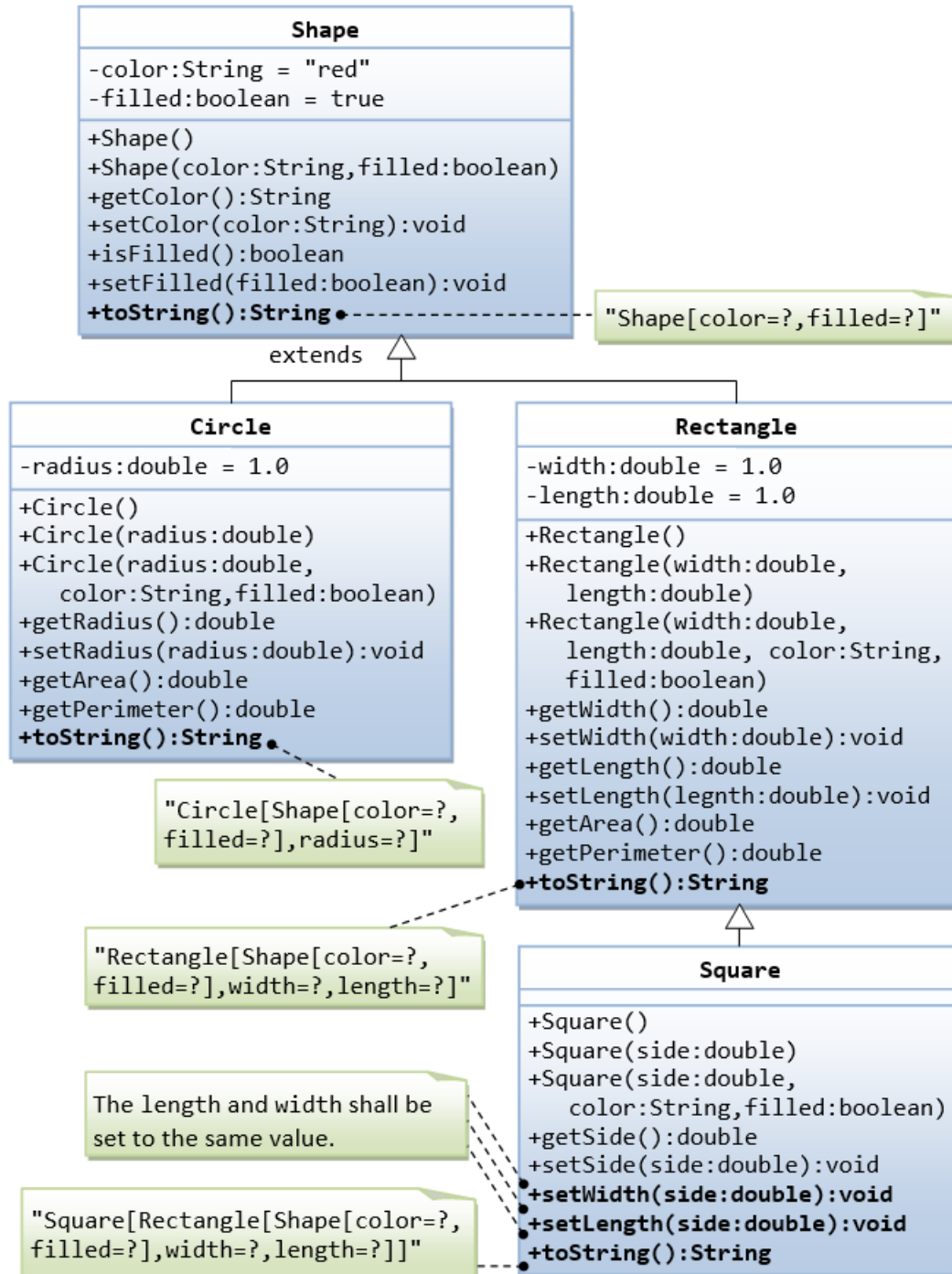
        // Cách 2:
        /*this.mm = this.mm + value;
        if (this.mm > 0) {
            incrementHour(this.mm / 60);
            this.mm = this.mm % 60;
        } else if (this.mm < 0){
            incrementHour(this.mm / 60 - 1);
            this.mm = (this.mm % 60 + 60) % 60;
        }*/

        // Số giây vẫn giữ nguyên
    }

    // Phương thức tăng số giây thêm 1 giá trị nguyên (có thể là giá trị dương hoặc
âm)
    public void incrementSecond(int value) {
        if (value == 0) return;
        this.ss = this.ss + value;
        while (this.ss < 0) {
            this.ss += 60;
            incrementMinute(-1);
        }
        while (this.ss > 59) {
            this.ss -= 60;
            incrementMinute(1);
        }
    }
}
```

Bài 2 (5 điểm)

Hãy xây dựng chương trình với các lớp được mô tả trong sơ đồ cây thừa kế dưới đây:



Bài làm:

```
package assignment5;

public class Bai4 {
    public static void main(String[] args) {
        Shape shape = new Shape();
        shape.setColor("blue");
        shape.setFilled(false);
        System.out.println(shape.toString());

        Circle circle = new Circle(3.5, "green", true);
        System.out.println(circle.toString());

        Rectangle rectangle = new Rectangle(12, 18, "gray", false);
        System.out.println(rectangle.toString());

        Square square = new Square(5, "cyan", true);
        square.setWidth(12);
        square.setLength(15);
        System.out.println(square.toString());
    }
}

class Shape {
    private String color;
    private boolean filled;

    public Shape() {
        this.color = "red";
        this.filled = true;
    }

    public Shape(String color, boolean filled) {
        this.color = color;
        this.filled = filled;
    }

    public String getColor() { // getter của thuộc tính color
        return this.color;
    }

    public void setColor(String color) { // setter của thuộc tính color
        this.color = color;
    }

    public boolean isFilled() { // getter của thuộc tính filled
        return this.filled;
    }

    public void setFilled(boolean filled) { // setter của thuộc tính filled
        this.filled = filled;
    }

    public String toString() {
        return "Shape[color=" + this.color + ",filled=" + this.filled + "];"
    }
}
```

```
class Circle extends Shape {
    private double radius;
    public Circle() {
        super();
        this.radius = 1.0;
    }

    public Circle(double radius) {
        super();
        this.radius = radius;
    }

    public Circle(double radius, String color, boolean filled) {
        super(color, filled);
        this.radius = radius;
    }

    public double getRadius() {
        return this.radius;
    }

    public void setRadius(double radius) {
        this.radius = radius;
    }

    public double getArea() {
        return this.radius * this.radius * 3.14;
    }

    public double getPerimeter() {
        return this.radius * 2 * 3.14;
    }

    public String toString() {
        return
        "Circle[Shape[color="+super.getColor()+",filled="+super.isFilled()+"],radius="+this.r
        adius+"]";
    }
}

class Rectangle extends Shape{
    private double width;
    private double length;

    public Rectangle() {
        super();
        this.width = 1.0;
        this.length = 1.0;
    }

    public Rectangle(double width, double length) {
        super();
        this.width = width;
        this.length = length;
    }

    public Rectangle(double width, double length, String color, boolean filled) {
        super(color, filled);
        this.width = width;
    }
}
```

```
        this.length = length;
    }

    public double getWidth() {
        return this.width;
    }

    public void setWidth(double width) {
        this.width = width;
    }

    public double getLength() {
        return this.length;
    }

    public void setLength(double length) {
        this.length = length;
    }

    public double getArea() {
        return this.width * this.length;
    }

    public double getPerimeter() {
        return (this.width + this.length) * 2;
    }

    public String toString() {
        return
"Rectangle[Shape[color="+super.getColor()+",filled="+super.isFilled()+"],width="+this
.width+",length="+this.length+"]";
    }
}

class Square extends Rectangle {
    public Square() {
        super();
    }

    public Square(double side) {
        super(side, side);
    }

    public Square(double side, String color, boolean filled) {
        super(side, side, color, filled);
    }

    public double getSide() {
        return this.getWidth();
    }

    public void setSide(double side) {
        this.setLength(side);
        this.setWidth(side);
    }

    public void setWidth(double side) {
        super.setWidth(side);
    }
}
```



```
public void setLength(double side) {  
    super.setLength(side);  
}  
  
public String toString() {  
    return  
"Square[Rectangle[Shape[color="+super.getColor()+",filled="+super.isFilled()+"],width  
="+this.getWidth()+",length="+this.getLength()+"]]";  
}  
}
```