

# Class và Object

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

1/ (3 điểm) Xây dựng một lớp Point dùng để biểu diễn các đối tượng điểm trên mặt phẳng tọa độ Oxy. Lớp Point gồm có 2 thuộc tính dùng riêng x, y và các phương thức:

- Phương thức khởi tạo gán giá trị cho 2 thuộc tính của Point
- Phương thức equals(Point p) cho phép so sánh giá trị của đối tượng Point hiện tại với một đối tượng Point p khác (so sánh bằng)
- Phương thức translate(double dx, double dy) cho phép dịch chuyển tọa độ của point(x, y) đến tọa độ mới point(x + dx, y + dy)
- Phương thức rotate(double theta) cho phép xoay đối tượng điểm đi một góc theta (radian) ngược chiều kim đồng hồ. Ta sử dụng công thức lượng giác sau:

$$x_2 = x_1 \cdot \cos\theta - y_1 \cdot \sin\theta$$

$$y_2 = x_1 \cdot \sin\theta + y_1 \cdot \cos\theta$$

(tham khảo: <https://www.khanacademy.org/partner-content/pixar/sets/rotation/v/set-7>)

- Phương thức distance(Point p) tính khoảng cách từ đối tượng Point hiện tại đến một đối tượng Point p khác
- Các phương thức getter và setter cho các thuộc tính

Bài làm:

```
package lab_assignment3;

public class Point {
    // Thành phần dữ liệu
    private double x, y;

    // Thành phần xử lý
    public Point(double x, double y) {
        this.x = x;
        this.y = y;
    }

    public boolean equals(Point p) {
        return (this.x == p.x && this.y == p.y);
    }

    public void translate(double dx, double dy) {
        this.x = this.x + dx;
        this.y = this.y + dy;
    }
}
```

```
public void rotate(double theta) {  
    double x1 = x;  
    double y1 = y;  
    this.x = x1 * Math.cos(theta) - y1 * Math.sin(theta);  
    this.y = x1 * Math.sin(theta) - y1 * Math.cos(theta);  
}  
  
public double distance(Point p) {  
    return Math.sqrt(Math.pow(p.y - this.y, 2) + Math.pow(p.x - this.x, 2));  
}  
  
public double getX() {  
    return this.x;  
}  
  
public void setX(double x) {  
    this.x = x;  
}  
  
public double getY() {  
    return this.y;  
}  
  
public void setY(double y) {  
    this.y = y;  
}  
}
```

2/ (3 điểm) Xây dựng một lớp IntegerSet dùng để biểu diễn một tập hợp các số nguyên dương từ 0 – 100. Tập hợp này được biểu diễn bằng một mảng **boolean**. Mỗi phần tử  $a[i]$  của mảng có giá trị là true nếu số nguyên  $i$  tồn tại trong tập hợp và có giá trị là false nếu  $i$  không tồn tại trong tập hợp. Một phương thức khởi tạo không tham số ban đầu sẽ tạo ra một tập hợp rỗng. Ví dụ: giả sử ta có tập hợp số nguyên  $A = \{1, 5, 9, 8\}$

Thì ta có biểu diễn của mảng  $a$  trong IntegerSet như sau:

	0	1	2	3	4	5	6	7	8	9	
$a =$	false	true	false	false	false	true	false	false	true	true	...

Hãy xây dựng các phương thức:

- Phương thức `union(IntegerSet set)` cho phép thay đổi tập hợp của đối tượng IntegerSet hiện tại bằng cách hội (hợp) với một tập hợp số nguyên IntegerSet set khác (Nghĩa là, mỗi phần tử trong tập hợp mới là tồn tại nếu phần tử đó tồn tại trong tập hợp này hoặc tập hợp kia)
- Phương thức `intersection(IntegerSet set)` cho phép thay đổi tập hợp của đối tượng IntegerSet hiện tại bằng cách giao với một tập hợp số nguyên IntegerSet set khác

(Nghĩa là, mỗi phần tử trong tập hợp mới là tồn tại nếu phần tử đó tồn tại đồng thời trong cả hai tập hợp)

- Phương thức `insertElement(int e)` cho phép thêm một phần tử số nguyên  $e$  ( $0 \leq e \leq 100$ ) vào tập hợp
- Phương thức `deleteElement(int e)` cho phép xóa một phần tử số nguyên  $e$  ( $0 \leq e \leq 100$ ) ra khỏi tập hợp
- Phương thức `isEqual(IntegerSet set)` cho phép kiểm tra xem tập hợp số nguyên của đối tượng hiện tại có bằng với tập số nguyên của một đối tượng `IntegerSet set` khác hay không
- Phương thức `display()` cho phép xuất các phần tử số nguyên trong tập hợp ra màn hình, mỗi phần tử cách nhau bởi một khoảng trắng

Bài làm:

```
package lab_assignment3;

public class IntegerSet {
    private boolean[] a;

    public IntegerSet() {
        a = new boolean[101];

        for (boolean element : a) { // foreach
            element = false;
        }

        /*for (int i=0; i<a.length; i++) {
            a[i] = false;
        }*/
    }

    public void union(IntegerSet set) {
        for (int i=0; i<101; i++) {
            if (this.a[i] == true || set.a[i] == true)
                this.a[i] = true;
        }
    }

    public void intersection(IntegerSet set) {
        for (int i=0; i<101; i++) {
            if (this.a[i] == true && set.a[i] == true)
                this.a[i] = true;
            else this.a[i] = false;
        }
    }

    public void insertElement(int e) {
        if (e >= 0 && e <= 100) {
            this.a[e] = true;
        }
    }
}
```

```
void deleteElement(int e) {
    if(e >= 0 && e <= 100) {
        this.a[e] = false;
    }
}

boolean isEqual(IntegerSet set) {
    for(int i = 0; i <= 100; i++) {
        if(this.a[i] != set.a[i])
            return false;
    }
    return true;
}

void display() {
    for(int i = 0; i <= 100; i++) {
        if(a[i] == true)
            System.out.print(i + " ");
    }
}
```

3/ (4 điểm) Xây dựng một lớp Date cho phép biểu diễn ngày tháng năm của một thời điểm. Lớp gồm có 3 thuộc tính cơ bản là ngày (int), tháng (int) và năm (int). Hãy xây dựng các phương thức của lớp:

- Cho phép khởi tạo các giá trị mặc định của thuộc tính là 01/01/2020 nếu đối tượng được tạo ra không có tham số truyền vào
- Cho phép khởi tạo các giá trị mặc định của thuộc tính tháng và năm là 01/2020 nếu đối tượng được tạo ra có tham số về ngày hợp lệ (nếu không hợp lệ thì quay về phương thức khởi tạo không tham số)
- Cho phép khởi tạo các giá trị mặc định của thuộc tính năm là 2020 nếu đối tượng được tạo ra có tham số về ngày và tháng hợp lệ (nếu không hợp lệ thì quay về phương thức khởi tạo không tham số)
- Cho phép khởi tạo các giá trị mặc định của các thuộc tính ngày, tháng và năm nếu đối tượng được tạo ra có tham số về ngày, tháng và năm hợp lệ (nếu không hợp lệ thì quay về phương thức khởi tạo không tham số)
- Xây dựng phương thức để xác định năm được biểu diễn bởi đối tượng có phải là năm nhuận hay không
- Xây dựng các phương thức getter và setter cho phép lấy giá trị thuộc tính và thay đổi giá trị thuộc tính một cách hợp lệ
- Phương thức display() cho phép xuất ra màn hình thông tin của đối tượng Date theo định dạng “ngày/tháng/năm”, mỗi thành phần có ít nhất 2 chữ số.

Ví dụ: date[ngày = 12, tháng = 8, năm = 2032]

date.display(): Xuất “12/08/2032”

Gợi ý: tham khảo bài 6 – bài tập Tuần 3

Bài làm:

```
package lab_assignment3;

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

    public Date() {
        this.date = 1;
        this.month = 1;
        this.year = 2020;
    }

    public Date(int date) {
        this();
        if (date >= 1 && date <= 31) {
            this.date = date;
        }
    }

    public Date(int date, int month) {
        this();
        int datesOfMonth; // Số ngày trong tháng
        if (month >= 1 && month <= 12) {
            datesOfMonth = numberOfDate(month, this.year);
            if (date >= 1 && date <= datesOfMonth) {
                this.date = date;
                this.month = month;
            }
        }
    }

    public Date(int date, int month, int year) {
        this();
        int datesOfMonth; // Số ngày trong tháng
        if (year > 0) {
            if (month >= 1 && month <= 12) {
                datesOfMonth = numberOfDate(month, year);
                if (date >= 1 && date <= datesOfMonth) {
                    this.date = date;
                    this.month = month;
                    this.year = year;
                }
            }
        }
    }

    public int numberOfDate(int month, int year) { // Tính số ngày trong tháng
        int[] datesOfMonth = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
        int[] datesOfMonthLeapYear = {31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
        // Năm nhuận tháng 2 có 29 ngày
```

```
        if (isLeapYear(year)) return datesOfMonthLeapYear[month];
        else return datesOfMonth[month];
    }

    public boolean isLeapYear(int year) {
        return (year % 4 == 0 && year % 100 != 0 || year % 400 == 0);
    }

    public int getDate() {
        return this.date;
    }

    public int getMonth() {
        return this.month;
    }

    public int getYear() {
        return this.year;
    }

    public void setDate(int date) {
        int datesOfMonth = numberOfDate(this.month, this.year);
        if (date >= 1 && date <= datesOfMonth)
            this.date = date;
    }

    public void setMonth(int month) {
        if (month >= 1 && month <= 12)
            this.month = month;
    }

    public void setYear(int year) {
        if (year > 0)
            this.year = year;
    }

    public void display() {
        System.out.printf("%02d/%02d/%02d\n", this.date, this.month, this.year);
    }
}
```