Chuyên đề 1

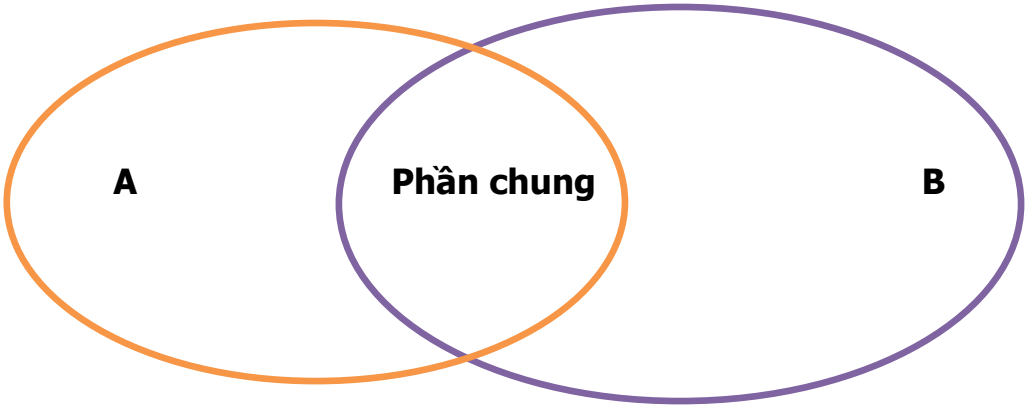
Tập hợp dữ liệu với Collection

Mục tiêu

- ✓ Vận dụng được kiến thức về Collection để tạo đối tượng lưu trữ tập hợp dữ liệu đối tượng
- ✓ Sử dụng được biểu thức quy tắc để xác thực dữ liệu
- ✓ Ứng dụng các lớp trong package java.lang để hỗ trợ nghiệp vụ lập trình.

Bài thực hành 1:

Yêu cầu: Viết chương trình Java core sinh ra 2 tập hợp số nguyên ngẫu nhiên (số lượng phần tử của mỗi tập hợp tùy ý nhưng không ít hơn 10 và quá 100):



Vận dụng kiến thức về mảng, collections để in ra:

- Các phần tử chung của A và B
- Các phần tử chỉ có ở tập hợp A
- Các phần tử chỉ có ở tập hợp B

```
Arr 1:[14, 11, 7, 13, 7, 16, 10, 19, 8]
Arr 2:[10, 1, 10, 16, 5, 6, 2, 1, 6, 17, 8, 7]
Phần tử chỉ có ở tập hợp A
[14, 11, 13, 19]
Phần tử chỉ có ở tập hợp B
[1, 5, 6, 2, 1, 6, 17]
Phần chung A: [7, 7, 16, 10, 8]
Phần chung B: [10, 10, 16, 8, 7]
Phần chung A + B: [7, 8, 10, 16]
BUILD SUCCESSFUL (total time: 0 seconds)
```

Code tham khảo:

TapHopNgauNhien.java

```
package javacoreii.chuyende1;

import java.util.ArrayList;
import java.util.Random;
import java.util.Set;
import java.util.TreeSet;

/**
 *
 * @author minhvufc
 */
public class TapHopNgauNhien {

    ArrayList<Integer> lstNumA = new ArrayList<>();
    ArrayList<Integer> lstNumB = new ArrayList<>();

    private void generateNumber() {
        int min = 0;
        int max = 20;
        // Sinh số ngẫu nhiên cho mảng
        for (int i = 0; i < 9; i++) {
            Random ran = new Random();
            int number = ran.nextInt(max - min + 1) + min;
            lstNumA.add(number);
        }
        System.out.println("Arr 1:" + lstNumA);
        for (int i = 0; i < 12; i++) {
            int number = (int) (Math.random() * ((max - min) + 1)) + min;
            lstNumB.add(number);
        }
```

```
        System.out.println("Arr 2:" + lstNumB);
    }

    private void intersect() {
        ArrayList<Integer> lstNumATemp = new ArrayList<>(lstNumA);
        ArrayList<Integer> lstNumBTemp = new ArrayList<>(lstNumB);

        // Phần tử có ở tập A mà không có ở tập B
        if (lstNumATemp.removeAll(lstNumB)) {
            System.out.println("Phần tử chỉ có ở tập hợp A");
            System.out.println(lstNumATemp);
        } else {
            System.out.println("2 tập hợp hoàn toàn khác biệt");
        }

        // Phần tử có ở tập B mà không có ở tập A
        if (lstNumBTemp.removeAll(lstNumA)) {
            System.out.println("Phần tử chỉ có ở tập hợp B");
            System.out.println(lstNumBTemp);
        } else {
            System.out.println("2 tập hợp hoàn toàn khác biệt");
        }

        ArrayList<Integer> intersectionA = new ArrayList<>(lstNumA);
        ArrayList<Integer> intersectionB = new ArrayList<>(lstNumB);
        // Phần chung A
        intersectionA.removeAll(lstNumATemp);
        System.out.println("Phần chung A: " + intersectionA);
        // Phần chung B
        intersectionB.removeAll(lstNumBTemp);
        System.out.println("Phần chung B: " + intersectionB);

        // Tổng hợp phần chung A + B
        Set<Integer> totalIntersect = new TreeSet<>(intersectionA);
        totalIntersect.addAll(intersectionB);
        System.out.println("Phần chung A + B: " + totalIntersect);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        TapHopNgauNhien demo = new TapHopNgauNhien();
        demo.generateNumber();
```

```
        demo.intersect();
    }

}
```

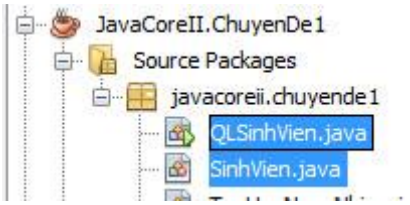
Bài thực hành 2:

Yêu cầu: Tạo class SinhVien cho đối tượng sinh viên gồm:

- Name – kiểu String: giới hạn 10 ký tự => sai nhập lại
- Email – kiểu String: đúng định dạng email => sai nhập lại
- Phone – kiểu String: định dạng số di động 10 số, chỉ chấp nhận đầu số 090, 091 và 098. VD: 0984.111.222 là hợp lệ
- Year – kiểu Integer: giới hạn so với năm hệ thống phải đủ từ 16 tuổi trở lên

Sử dụng collection để lưu trữ tập hợp đối tượng trên, thực hiện các yêu cầu sau:

- In danh sách sinh viên trên theo tên thứ tự alphabel
- In danh sách sinh viên trên theo độ tuổi tăng dần
- In danh sách sinh viên sau khi xóa bỏ sinh viên thứ 2.



Code tham khảo:

SinhVien.java

```
package javacoreii.chuyende1;

import java.time.Year;
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

/**
 *
 * @author minhvufc
 */
public class SinhVien implements Comparable<SinhVien> {

    protected String name;
    protected String email;
```

```
protected String phone;
protected int year;

public SinhVien() {
}

public SinhVien(String name, String email, String phone, int year) {
    this.name = name;
    this.email = email;
    this.phone = phone;
    this.year = year;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public String getPhone() {
    return phone;
}

public void setPhone(String phone) {
    this.phone = phone;
}

public int getYear() {
    return year;
}

public void setYear(int year) {
    this.year = year;
}
```

```
public void input() {
    Scanner s = new Scanner(System.in);
    // Nhập tên giới hạn 10 ký tự
    boolean isDone = false;
    do {
        System.out.println("Nhập họ và tên");
        this.name = s.nextLine();
        if (this.name.length() <= 10) {
            isDone = true;
        } else {
            System.err.println("Tên không quá 10 ký tự");
        }
    } while (!isDone);

    // Nhập đúng email
    isDone = false;
    do {
        System.out.println("Nhập email");
        this.email = s.nextLine();
        // minh.vt@yahoo.com.vn
        Pattern p = Pattern.compile(
                "^[a-z0-9]+(.[a-z0-9]+)*@[a-z0-9]+.{1}[a-z]{2,3}(.[a-z]{2})*$");
        Matcher matcher = p.matcher(this.email);
        if (matcher.matches()) {
            isDone = true;
        } else {
            System.err.println("Sai định dạng email");
        }
    } while (!isDone);

    // Nhập đúng số đt
    isDone = false;
    do {
        System.out.println("Nhập số điện thoại");
        this.phone = s.nextLine();
        // 0984.111.222
        Pattern p = Pattern.compile(
                "^(090|091|098){1}[0-9].[0-9]{3}.[0-9]{3}$");
        Matcher matcher = p.matcher(this.phone);
        if (matcher.matches()) {
            isDone = true;
        } else {
            System.err.println("Sai định dạng điện thoại");
```

```
            }
        } while (!isDone);

        // Nhập năm sinh tính đến thời điểm hiện tại phải đủ 16 tuổi
        isDone = false;
        do {
            System.out.println("Nhập năm sinh");
            this.year = s.nextInt();
            Year born = Year.of(this.year);
            Year atNow = Year.now();
            int age = atNow.compareTo(born);
            if (age >= 16) {
                isDone = true;
            } else {
                System.err.println("Chưa đủ tuổi quy định");
            }
        } while (!isDone);
        System.out.println("Cảm ơn đã nhập");
    }

    public void output() {
        System.out.println("SinhVien{" + "name=" + name + ", email=" + email + ",
phone=" + phone + ", year=" + year + '}');
    }

    @Override
    public int compareTo(SinhVien o) {
        return this.name.compareTo(o.getName());
    }

}
```

QLSinhVien.java

```
package javacoreii.chuyende1;

import java.util.ArrayList;
import java.util.Collections;

/**
 *
 * @author minhvufc
 */
public class QLSinhVien {
```

```
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        ArrayList<SinhVien> lstSVBkap = new ArrayList<>();

        // Nhập thông tin 3 sinh viên
        for (int i = 0; i < 3; i++) {
            SinhVien sv = new SinhVien();
            sv.input();
            lstSVBkap.add(sv);
        }

        // Test
//        lstSVBkap.add(new SinhVien("B", "xxx", "123", 1984));
//        lstSVBkap.add(new SinhVien("A", "xxx", "123", 2000));
//        lstSVBkap.add(new SinhVien("D", "xxx", "123", 1999));
//        lstSVBkap.add(new SinhVien("H", "xxx", "123", 1995));
//        lstSVBkap.add(new SinhVien("E", "xxx", "123", 1996));
        // In danh sách theo thứ tự
        System.out.println("\nThứ tự alphabel");
        Collections.sort(lstSVBkap);
        for (SinhVien sv : lstSVBkap) {
            sv.output();
        }

        // Sắp xếp theo độ tuổi tăng dần
        System.out.println("\nThứ tự tuổi tăng dần");
        lstSVBkap.sort((o1, o2) -> {
            return -(o1.getYear() - o2.getYear());
        });
        for (SinhVien sv : lstSVBkap) {
            sv.output();
        }

        System.out.println("\nXóa phần tử 2");
        // Xóa bỏ sinh viên thứ 2
        lstSVBkap.remove(1);
        for (SinhVien sv : lstSVBkap) {
            sv.output();
        }
    }

}
```