

Bài tập chương 4

Phần 1: Generics

1. Cho lớp Group được định nghĩa như sau:

```
public class Group<T> {  
    T items[]; // mảng lưu các phần tử trong nhóm  
    int num; // số phần tử hiện tại trong nhóm  
    public Group(Class<T> dataType, int length) {  
        this.items = (T[])  
java.lang.reflect.Array.newInstance(dataType, length);  
    }  
    public void addItem(T t) {  
        //TODO: thêm phần tử t vào nhóm  
    }  
    public T getItem(int index) {  
        //TODO: lấy phần tử thứ index từ nhóm  
    }  
    public int getNumberItems() {  
        //TODO: trả về số phần tử trong nhóm  
    }  
}
```

a) Viết phần định nghĩa các hàm theo chỉ dẫn

b) Trong hàm main của lớp mặc định, sử dụng lớp Group để biểu diễn một nhóm đối tượng Integer và một nhóm đối tượng String. Biết để khởi tạo đối tượng Group cần truyền kiểu lớp cho T như sau:

```
Group<Integer> group_int=new Group<>(Integer.class,100);  
Group<String> group_str=new Group<>(String.class,100);
```

2. Viết lớp generic Pair<K,V> trong đó K và V là hai tham số kiểu.

Các thuộc tính: k và v thuộc hai kiểu K và V

Các hàm cần định nghĩa:

- Hàm khởi tạo có tham số : gán giá trị cho k và v từ tham số khởi tạo

- Hàm lấy giá trị của k

- Hàm lấy giá trị của v

a) Định nghĩa lớp Pair và các hàm của nó

b) Sử dụng lớp Pair trong hàm main để minh họa các hàm

c) Định nghĩa lớp IntPair thừa kế lớp Pair, trong đó thay tham số kiểu K bằng kiểu Integer, sử dụng lớp này trong hàm main để minh họa các hàm.

3. Cho các lớp sau:

```
public abstract class Document {
    String title;
    int pages;
    String size;
    public abstract void print();
}

public class Book extends Document {
    String author;
    @Override
    public void print() {
        System.out.println("print the book");
    }
}

public class Magazine extends Document {
    String publisher;
    @Override
    public void print() {
        System.out.println("print the magazine");
    }
}

public class Bookshelf<T> {
    List<T> items;
```

```

    int num;

    public Bookshelf() {
        items=new ArrayList<T>();
    }

    public void putItems(T t) {
        //TODO: thêm đối tượng t và danh sách items
    }
}

```

a) Định nghĩa hàm putItems trong lớp Bookshelf

b) Nếu muốn lớp Bookshelf chỉ chấp nhận các tham số kiểu là dẫn xuất của lớp Document thì làm như thế nào? Viết thêm hàm getItemByTitle(String title) trả về một item theo title truyền vào, nếu không tìm thấy item phù hợp trong danh sách item thì trả về null.

c) Viết hàm main minh họa hoạt động lớp Bookshelf.

Phần 2: Xử lý ngoại lệ

4. Cho các lớp sau:

```

class NegativeAgeException extends Exception{
    public String getMessage(){
        return "Current year can not be less than birthyear";
    }
}

class Student{
    String name;
    int birthyear;//năm sinh
    public Student() {
    }
    public Student(String name, int birthyear) {
        this.name = name;
        this.birthyear = birthyear;
    }
}

```

```
        public int getAge(int year){  
            return year-birthyear;  
        }  
    }
```

- a) Sửa lại hàm `getAge()` của lớp `Student` để nó ném ra ngoại lệ `NegativeAgeException` nếu năm tham số nhỏ hơn năm sinh.
- b) Thêm hàm `setName(String name)` để thiết lập thuộc tính `name`, nếu tham số truyền vào là chuỗi rỗng thì ném ra ngoại lệ kiểu `IllegalArgumentException`
- c) Viết hàm `main` để minh họa việc bắt lỗi khi sử dụng `Student`