HD LAB 8: GENERIC

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Sử dụng Generic

PHẦN I

Bài 1 (2 điểm)

Tạo file **Lab8Bai1.java** sử dụng ArrayList và thực hiện các công việc sau:

- ✓ Thêm vào ArrayList 1 số nguyên
- ✓ Thêm vào ArrayList 1 số thực
- ✓ Thêm vào ArrayList 1 giá trị boolean
- ✓ Thêm vào ArrayList 1 xâu ký tự
- ✓ In ra màn hình 4 giá trị trên từ ArrayList

```
public class Lab8Bai1 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        ArrayList arr = new ArrayList();
        arr.add(50);
        arr.add(5.5);
        arr.add(true);
        arr.add("Java");
//      Xuat noi dung ra man hinh
        System.out.println(">> Integer: " + arr.get(0));
        System.out.println(">> Double: " + arr.get(1));
        System.out.println(">> Boolaen: " + arr.get(2));
        System.out.println(">> String: " + arr.get(3));
    }

}
```

Bài 2 (2 điểm)

Tạo file **Lab8Bai2.java** sử dụng ArrayList<> và thực hiện các công việc sau:

- ✓ Generic ArrayList là kiểu Integer (ArrayList<Integer> myarrr = new ArrayList<Integer>();)
- ✓ Sử dụng vòng lặp để nhập các số từ 1 đến 10 vào mảng myarrr trên
- ✓ Sử dụng vòng lặp để hiển thị các số từ 1 đến 10 từ mảng myarrr.

```
public class Lab8Bai2 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        ArrayList<Integer> myArr = new ArrayList<Integer>();
//      Nhap 10 so nguyen
        for (int i = 0; i < 10; i++) {
            myArr.add(i);
        }
//      Xuat ra man hinh 10 so co trong danh sach
        for (int i = 0; i < 10; i++) {
            System.out.println(">> " + myArr.get(i));
        }
    }

}
```

PHẦN II

Bài 3 (2 điểm)

1. Tạo lớp mô tả thông tin sản phẩm gồm tên và giá như sau

```
public class Product implements Serializable {

    String name;
    double price;

    public Product(String name, double price) {
        this.name = name;
        this.price = price;
    }

    public String getName() {
        return name;
    }

    public double getPrice() {
        return price;
    }

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

    public void setPrice(double price) {
        this.price = price;
    }
}
```

2. Tạo lớp DAO và khai báo các phương thức thao tác CSDL như sau

```
abstract public class DAO<Entity> {

    protected List<Entity> list = new ArrayList<>();

//  Phuong thuc them moi phan tu
    public void add(Entity entity) {
        list.add(entity);
    }

//  Phuong thuc xoa phan tu
    public void remove(Entity entity) {
        list.remove(entity);
    }

//  Phuong thuc truu tuong cap nhat thong tin phan tu
    abstract public void update(Entity entity);

//  Phuong thuc truu tuong tim kiem phan tu
    abstract public Entity find(String id);

//  Phuong lay lay danh sach
    public List<Entity> getList() {
        return list;
    }
```

```
//  Phuong thuc load du lieu tu file vao chuong trinh
    public void load(String path) {
        try {
            FileInputStream fis = new FileInputStream(path);
            ObjectInputStream ois = new ObjectInputStream(fis);
            list = (List<Entity>) ois.readObject();
            ois.close();
            fis.close();
        } catch (Exception e) {
            throw new RuntimeException(e.getMessage());
        }
    }
```

```
//  Phuong thuc ghi du lieu tu chuong trinh vao file
    public void store(String path) {
        try {
            FileOutputStream fos = new FileOutputStream(path);
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(list);
            oos.close();
            fos.close();
        } catch (Exception e) {
            throw new RuntimeException(e.getMessage());
        }
    }
```

Bài 4 (2 điểm)

1. Tạo lớp ProductDAO kế thừa từ lớp DAO (ở bài 3) và viết mã thực hiện các phương thức abstract

```
public class ProductDAO extends DAO<Product> {

//  Viet lai phuong thuc update cua lop DAO
    @Override
    public void update(Product entity) {
        for (int i = 0; i < list.size(); i++) {
            if (list.get(i).name.equalsIgnoreCase(entity.name)) {
                list.set(i, entity);
            }
        }
    }

//  Viet lai phuong thuc find cua lop DAO
    @Override
    public Product find(String id) {
        for (Product p : list) {
            if (p.name.equals(id)) {
                return p;
            }
        }
        return null;
    }

}
```

2. Tạo lớp ProductManager chứa main() thực hiện việc quản lý 2 sản phẩm như sau:

```
public class ProductManager {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        Product p1 = new Product("iPhone9", 1000.0);
        Product p2 = new Product("Samsung Start", 3000.0);

        ProductDAO dao = new ProductDAO();
        dao.add(p1);
        dao.add(p2);
        dao.store("prod.txt");

        ProductDAO dao2 = new ProductDAO();
        dao2.load("prod.txt");
        Product p = dao2.find("iPhone9");
        System.out.println(">> Name: " + p.name);
        System.out.println(">> Price: " + p.price);

    }

}
```

Hết