**~\OneDrive - VietNam National University - HCM INTERNATIONAL UNIVERSITY\Desktop\DSA\DSA LAB NEW\Lab 3 Stacks & Queues\ITITSB22029_DoMinhDuy_Lab3\ReverseApp\ReversePersonApp.java**

```java
// reverse.java
// stack used to reverse a string
// to run this program: C>java ReverseApp

// Create a stack of objects of class Person and use to reverse a list of persons.
import java.util.*; // for I/O and List

// Define a class Person to store the name and age
class Person {
   private String name;
   private int age;

   public Person(String name, int age) {
      this.name = name;
      this.age = age;
   }

   public String getName() {
      return name;
   }

   public int getAge() {
      return age;
   }

   @Override
   public String toString() {
      return name + " (" + age + " years)";
   }
}

// Stack class that stores Person objects
class PersonStack {
   private int maxSize;
   private Person[] stackArray;
   private int top;

   public PersonStack(int max) { // Constructor
      maxSize = max;
      stackArray = new Person[maxSize];
      top = -1;
   }

   public void push(Person p) { // Push a Person object onto the stack
      stackArray[++top] = p;
   }

```

```
48     public Person pop() { // Pop a Person object from the stack
49        return stackArray[top--];
50     }
51
52     public boolean isEmpty() { // Check if stack is empty
53        return (top == -1);
54     }
55  }
56
57  // Reverser class to reverse a list of Person objects
58  class PersonReverser {
59     private List<Person> persons; // List of Person objects to reverse
60
61     public PersonReverser(List<Person> persons) {
62        this.persons = persons;
63     }
64
65     public List<Person> reverse() { // Reverse the list using a stack
66        int stackSize = persons.size();
67        PersonStack stack = new PersonStack(stackSize); // Create a stack of Person objects
68
69        // Push all persons onto the stack
70        for (Person person : persons) {
71           stack.push(person);
72        }
73
74        // Pop all persons from the stack to reverse the list
75        List<Person> reversedPersons = new ArrayList<>();
76        while (!stack.isEmpty()) {
77           reversedPersons.add(stack.pop());
78        }
79
80        return reversedPersons;
81     }
82  }
83
84  // Main class to demonstrate the reversing of a list of Person objects
85  public class ReversePersonApp {
86     public static void main(String[] args) {
87        // Create a list of Person objects
88        List<Person> personList = new ArrayList<>();
89        personList.add(new Person("Alice", 30));
90        personList.add(new Person("Bob", 25));
91        personList.add(new Person("Charlie", 35));
92        personList.add(new Person("Diana", 28));
93
94        System.out.println("Original List of Persons:");
95        for (Person person : personList) {
96           System.out.println(person);
97        }
```

```
98
99        // Reverse the list using the PersonReverser
100       PersonReverser reverser = new PersonReverser(personList);
101       List<Person> reversedList = reverser.reverse();
102
103       System.out.println("\nReversed List of Persons:");
104       for (Person person : reversedList) {
105          System.out.println(person);
106       }
107    }
108 }
109
110 ////////////////////////////////////////////////////////////
111
```