## 

Mubeen Khalid

(SP21-BSE-015)

Design Patterns Lab

Iterator Pattern

Iterator.java:

package UpdatedCode;

public interface Iterator {

    public boolean hasNext();

    public Object next();

}

Container.java:

package UpdatedCode;

public interface Container {

    public Iterator getIterator();

 }

NameRepository.java:

package UpdatedCode;

import java.util.Arrays;

import java.util.Comparator;

public class NameRepository implements Container {

   public Student students[] = {

         new Student("001", "Robert", "Male", "1234567890"),

         new Student("002", "John", "Male", "2345678901"),

         new Student("003", "Julie", "Female", "3456789012"),

         new Student("004", "Lora", "Female", "4567890123")

   };

   @Override

   public Iterator getIterator() {

      return new NameIterator();

   }

   private class NameIterator implements Iterator {

      int index;

      @Override

      public boolean hasNext() {

         return index < students.length;

      }

      @Override

      public Object next() {

         if (this.hasNext()) {

            return students[index++];

         }

         return null;

      }

      public boolean hasPrevious() {

         return index > 0;

      }

      public Object previous() {

         if (this.hasPrevious()) {

            return students[--index];

         }

         return null;

      }

      public void moveToLast() {

         index = students.length - 1;

      }

      public void moveToFirst() {

         index = 0;

      }

   }

   public void printStudentsWithNameStartingFrom(String anyStartingValue) {

      NameIterator iterator = new NameIterator();

      iterator.moveToLast();

      while (iterator.hasPrevious()) {

         Student student = (Student) iterator.previous();

         if (student.getName().startsWith(anyStartingValue)) {

            System.out.println(student);

         }

      }

   }

   public void printAllStudents() {

      Arrays.sort(students, Comparator.comparing(Student::getName));

      for (Student student : students) {

         System.out.println(student);

      }

   }

}

Student.java:

package UpdatedCode;

public class Student {

    private String regNo;

    private String name;

    private String gender;

    private String phoneNumber;

    public Student(String regNo, String name, String gender, String phoneNumber) {

        this.regNo = regNo;

        this.name = name;

        this.gender = gender;

        this.phoneNumber = phoneNumber;

    }

    public String getRegNo() {

        return regNo;

    }

    public String getName() {

        return name;

    }

    public String getGender() {

        return gender;

    }

    public String getPhoneNumber() {

        return phoneNumber;

    }

    @Override

    public String toString() {

        return "Student [RegNo=" + regNo + ", Name=" + name + ", Gender=" + gender + ", PhoneNumber=" + phoneNumber + "]";

    }

}

IteratorPatternDemo.java:  
package UpdatedCode;

public class IteratorPatternDemo {

    public static void main(String[] args) {

        NameRepository namesRepository = new NameRepository();

        // Print all students

        System.out.println("All students:");

        namesRepository.printAllStudents();

        // Print students with names starting from "Julie"

        System.out.println("\nStudents with names starting from 'Julie':");

        namesRepository.printStudentsWithNameStartingFrom("J");

    }

}

Output:

