Министерство образования и науки Украины

Национальный технический университет Украины

«Киевский политехнический институт им. Игоря Сикорского»

Факультет информатики и вычислительной техники

Кафедра вычислительной техники

Лабораторная работа №5

по дисциплине «Инженерия программного обеспечения»

на тему «Структурные шаблоны проектирования

Iterator, Mediator, Observer»

ВЫПОЛНИЛА:

студентка 2-го курса

ФИВТ гр. ИВ-71

Молчанова В.С.

Зачетка – 7110

Вариант – 0

ПРОВЕРИЛ:

Старший преподаватель

к.т.н., с. н. с.

Антонюк А.И.

Киев - 2018

**1. Цель работы:**

Ознакомление с видами шаблонов проектирования ПЗ. Изучение структурных шаблонов. Получение базовых навыков применения шаблонов Iterator, Mediator, Observer.

**2. Вариант задания 7110 % 9 = 0:**

0. Определить спецификации классов, которые инкапсулируют линейный список объектов и реализуют возможность последовательного обхода в прямом и обратном направлениях, обходя пустые элементы этой структуры и не раскрывая ее сущности перед пользователем.

**3. Код программы**

**package** com.lab111.labwork5;

/\*\*

\*

\* **@author** Varvara Molchanova

\* **@version** 5.0

\*/

**public** **interface** Iterator {

**public** **boolean** hasNext();

**public** **boolean** hasPrevious();

**public** Object next();

**public** Object previous();

**public** **int** getIndex();

**public** **void** setIndex(**int** index);

}

**package** com.lab111.labwork5;

/\*\*

\*

\* **@author** Varvara Molchanova

\* **@version** 5.0

\*/

**public** **interface** Container {

**public** Iterator getIterator();

}

**package** com.lab111.labwork5;

/\*\*

\*

\* **@author** Varvara Molchanova

\* **@version** 5.0

\*

\*\*/

**public** **class** Name **implements** Container {

**public** String names[] = { "Bumblebee", "Bison", "Cat", "Wolf", **null**, "Owl" };

**public** Iterator getIterator() {

**return** **new** NameIterator(**names**);

}

}

**package** main.java.com.lab111.labwork5;  
  
**public class** NameIterator **implements** Iterator {  
 **int index**;  
 **private** String **names**[];  
  
 **public** NameIterator(String[] names){  
 **this**.**names** = names;  
 }  
  
 **public int** getIndex() {  
 **return index**;  
 }  
  
 **public void** setIndex(**int** index) {  
 **this**.**index** = index;  
 }  
  
 @Override  
 **public boolean** hasNext() {  
 **if** (**index** < **names**.**length**) {  
 **return true**;  
 }  
 **return false**;  
 }  
  
 @Override  
 **public boolean** hasPrevious() {  
 **if** (**index** > -1) {  
 **return true**;  
 }  
 **return false**;  
 }  
  
 @Override  
 **public** Object next() {  
 **if** (**this**.hasNext()) {  
 **return names**[**index**++];  
 }  
 **return null**;  
 }  
  
 @Override  
 **public** Object previous() {  
 **if** (**this**.hasPrevious()) {  
 **return names**[**index**--];  
 }  
 **return null**;  
 }  
  
}

**package** com.lab111.labwork5;

/\*\*

\* Iterator pattern applied.

\*

\* **@author** Varvara Molchanova

\* **@version** 5.0

\*/

**public** **class** Main {

**public** **static** **void** main(String[] args) {

Name name = **new** Name();

Iterator iter = name.getIterator();

System.***out***.println("Normal order:");

**while** (iter.hasNext()) {

String element = (String) iter.next();

System.***out***.println("Name:" + element);

}

System.***out***.println();

**while** (iter.hasNext()) {

String element = (String) iter.next();

**if** (name != **null**)

System.***out***.println("Name:" + element);

}

System.***out***.println();

System.***out***.println("Reverse order:");

iter.setIndex(name.names.length - 1);

**while** (iter.hasPrevious()) {

String element = (String) iter.previous();

System.***out***.println("Name: " + element);

}

System.***out***.println();

System.***out***.println("Without null:");

**for** (Iterator iter2 = name.getIterator(); iter2.hasNext();) {

String element = (String) iter2.next();

**if** (element != **null**)

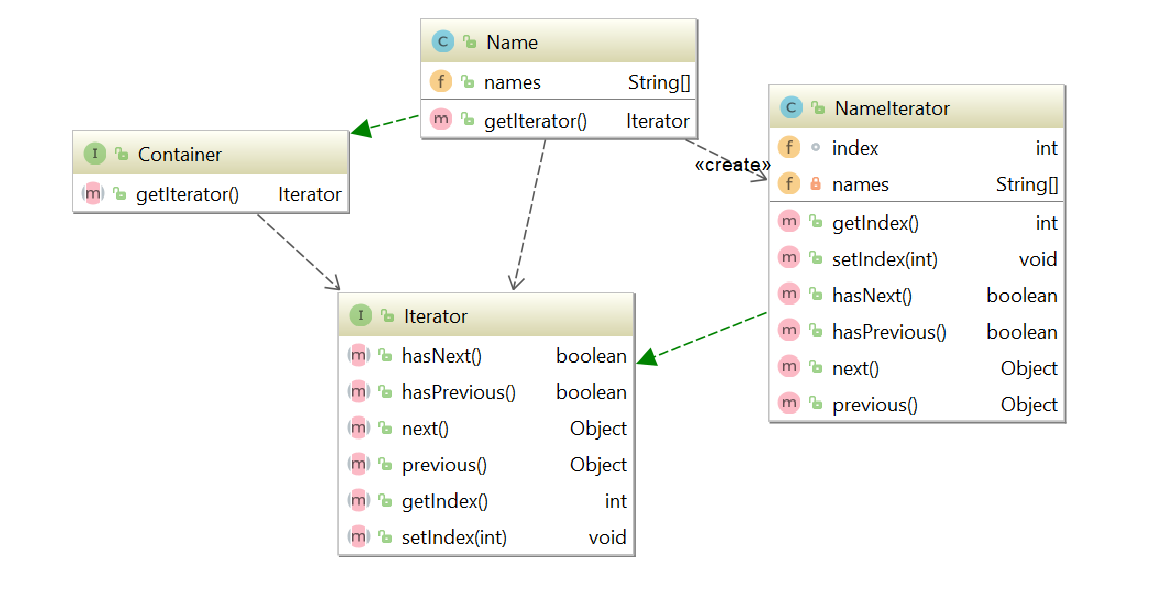
System.***out***.println("Name : " + element);

}

}

}

**4. Диаграмма классов**



**5. Результаты тестирования программы**

Normal order:

Name: Bumblebee

Name: Bison

Name: Cat

Name: Wolf

Name: null

Name: Owl

Reverse order:

Name: Owl

Name: null

Name: Wolf

Name: Cat

Name: Bison

Name: Bumblebee

Without null:

Name: Bumblebee

Name: Bison

Name: Cat

Name: Wolf

Name: Owl

**6. Выводы**

Ознакомились с темой лабораторной работы.

Приобретены соответствующие навыки работы.

Разработаны соответствующую тестовую программу.

Результаты успешной работы тестовой программы, приведенные выше, подтверждают правильность выбранных решений. Конечная цель работы достигнута.