Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №5

З предмету: «Інженерія програмного забезпечення»

Виконав: студент групи ІО-33

Шуркіна А. О.

Перевірив: Антонюк А. І.

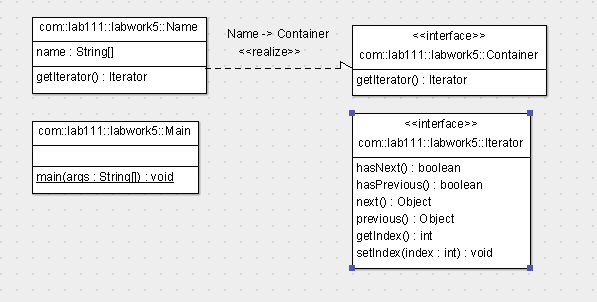
Київ 2014р.

**Варіант**

№зк=3330 mod 9=0

Визначити специфікації класів, які інкапсулюють лінійний список об’єктів та реалізують можливість послідовного обходу у прямому та зворотному порядку оминаючи порожні елементи цієї структури та не розкриваючи її сутності перед користувачем.

**Діаграма класів**

****

**Код програми**

**package** com.lab111.labwork5;

/\*\*

\*

\* **@author** Nastya Shurkina

\* **@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** Nastya Shurkina

\* **@version** 5.0

\*/

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

**public** Iterator getIterator();

}

**package** com.lab111.labwork5;

/\*\*

\*

\* **@author** Nastya Shurkina

\* **@version** 5.0

\*

\*\*/

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

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

**public** Iterator getIterator() {

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

}

**private** **class** NameIterator **implements** Iterator {

**int** index;

**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** Nastya Shurkina

\* **@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);

}

}

}