Міністерство освіти і науки України

Національний Університет «Львівська Політехніка»

Кафедра АСУ

Звіт до лабораторної роботи №5

з курсу: **«Інноваційні інформаційні технології»**

на тему:

**«Розробка шаблонів класів і функцій»**

**Виконав:**

студент групи КНМ-12

Новосад Б. І.

**Перевірив:**

старший викладач

Ковальчук А.М.

Львів 2015

Мета: Вивчення поняття шаблонів класів і функцій С++ та принципів роботи з ними

**Завдання 18.**

Створити двонапрямлений лінійний список для знаходження скалярного добутку головної і бічної діагоналей.

Хід роботи

**Клас DoublyLinkList**

**package** com.company;  
  
*/\*\*  
 \* Created by Bogdan on 22.10.2016.  
 \*/***public class** DoublyLinkList<T> {  
  
 **private static class** Node<T> {  
 **private** T data;  
 **private** Node next;  
 **private** Node prev;  
  
 **public** Node(T data) {  
 **this**.data = data;  
 }  
  
 **public void** displayNode() {  
 System.out.print(data + **" "**);  
 }  
  
 @Override  
 **public** String toString() {  
 **return** data.toString();  
 }  
 }  
  
 **public** Node first = **null**;  
 **public** Node last = **null**;  
  
 **public void** addFirst(T data) {  
 Node newNode = **new** Node(data);  
  
 **if** (isEmpty()) {  
 newNode.next = **null**;  
 newNode.prev = **null**;  
 first = newNode;  
 last = newNode;  
  
 } **else** {  
 first.prev = newNode;  
 newNode.next = first;  
 newNode.prev = **null**;  
 first = newNode;  
 }  
 }  
  
 **public void** addLast(T data){  
 Node newNode = **new** Node(data);  
 **if** (isEmpty()) {  
 newNode.next = **null**;  
 newNode.prev = **null**;  
 first = newNode;  
 last = newNode;  
  
 } **else** {  
 last.next = newNode;  
 newNode.prev = last;  
 newNode.next = **null**;  
 last = newNode;  
 }  
 }  
  
 **public boolean** isEmpty() {  
 **return** (first == **null**);  
 }  
  
 **public void** displayList() {  
 Node current = first;  
 **while** (current != **null**) {  
 current.displayNode();  
 current = current.next;  
 }  
 System.out.println();  
 }  
  
 **public void** removeFirst() {  
 **if** (!isEmpty()) {  
 Node temp = first;  
  
 **if** (first.next == **null**) {  
 **first** = **null**;  
 **last** = **null**;  
 } **else** {  
 **first** = **first**.**next**;  
 **first**.**prev** = **null**;  
 }  
 System.***out***.println(temp.toString() + **" is popped from the list"**);  
 }  
 }  
  
 **public void** removeLast() {  
 Node temp = last;  
  
 **if** (!isEmpty()) {  
  
 **if** (first.next == **null**) {  
 first = **null**;  
 last = **null**;  
 } **else** {  
 last = last.prev;  
 last.next = **null**;  
 }  
 }  
 System.out.println(temp.toString() + **" is popped from the list"**);  
 }  
  
 **public** T dotProduct(DoublyLinkList<T> bichnaDiagonal, Multiplication<T> multiplication, Addition<T> addition, T rezult){  
  
  
 Node currentG = **this**.first, currentB = bichnaDiagonal.first;  
  
 **while** (currentG!=**null**||currentB!=**null**){  
 T rez = multiplication.multiplie((T)currentB.data,(T)currentG.data);  
 rezult = addition.add(rezult,rez);  
 currentB=currentB.next;  
 currentG = currentG.next;  
 }  
 **return** rezult;  
 }  
}

**Використання класів.**

**package** com.company;  
  
  
**import** java.util.Random;  
  
**public class** Main {  
  
 **public static void** main(String[] args) {  
 *// write your code here* DoublyLinkList<Integer> golovnaDiagonal = **new** DoublyLinkList<>();  
 DoublyLinkList<Integer> bichnaDiagonal = **new** DoublyLinkList<>();  
  
  
  
 **int** n = 3;  
  
 Integer[][] matrix = **new** Integer[10][10];  
  
 Random random = **new** Random();  
  
 **for**(**int** i=0;i<n;i++){  
 **for**(**int** j=0;j<10;j++){  
 matrix[i][j] = random.nextInt(10);  
 }  
 }  
  
  
  
 **for** (**int** i = 0; i < n; i++) {  
 **for** (**int** j = 0; j < n; j++) {  
 System.***out***.print(matrix[i][j] + **"\t"**);  
 }  
 System.***out***.println();  
 }  
  
  
 **for** (**int** i = 0; i < n; i++) {  
 golovnaDiagonal.addLast(matrix[i][i]);  
 bichnaDiagonal.addLast(matrix[i][n-i-1]);  
 }  
  
 golovnaDiagonal.displayList();  
  
 bichnaDiagonal.displayList();  
  
  
 Integer rez = golovnaDiagonal.dotProduct(bichnaDiagonal,**new** IntegetMultiplier(), **new** IntegerAddition(),**new** Integer(0));  
  
  
 System.***out***.println(rez);  
 }  
}

**Результат роботи програми**

8 9 1 7 2 6 0 9 8 5

2 9 0 2 7 3 1 7 5 7

5 3 2 4 8 9 2 8 1 4

0 4 7 7 6 2 9 8 3 8

8 7 1 5 2 3 5 4 3 5

1 6 8 9 0 4 2 8 8 8

1 3 5 2 8 6 2 6 3 3

1 6 1 4 6 3 3 2 8 4

8 1 3 1 2 6 3 3 5 4

1 6 5 5 6 0 8 2 9 8

8 9 2 7 2 4 2 2 5 8

5 5 8 9 3 0 2 1 1 1

189

**Висновок**: шаблонні класи дуже широко використовуються в програмуванні, оскільки надають можливість створення універсальних класів які створюють на етапі компіляції. Колекції є яскравим прикладом цього, оскільки реалізують це найбільш логічно. У лабораторній роботі я створив двозв’язний список, який вміє працювати з будь-якими типами даних, і вміло застосував його на практиці.