НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра автоматизованих систем обробки інформації і систем управління

**ЛАБОРАТОРНА РОБОТА № 4**

з дисципліни «Об’єктно-орієнтоване програмування»

**на тему: «**шаблони класів С++**»**

Виконав:

Студент 2-го курсу групи ІП-72

Кенгерлі Ельмар Фаігович

Київ – 2018 рік

Зміст

[Ціль роботи](file:///C:/Users/Acer/Downloads/Telegram%20Desktop/ООП_Звіт.docx) [3](file:///C:/Users/Acer/Downloads/Telegram%20Desktop/ООП_Звіт.docx)

[Постановка задачі](file:///C:/Users/Acer/Downloads/Telegram%20Desktop/ООП_Звіт.docx) [4](file:///C:/Users/Acer/Downloads/Telegram%20Desktop/ООП_Звіт.docx)

[UML-діаграма класів](file:///C:/Users/Acer/Downloads/Telegram%20Desktop/ООП_Звіт.docx) [5](file:///C:/Users/Acer/Downloads/Telegram%20Desktop/ООП_Звіт.docx)

[Код програми](file:///C:/Users/Acer/Downloads/Telegram%20Desktop/ООП_Звіт.docx) [6](file:///C:/Users/Acer/Downloads/Telegram%20Desktop/ООП_Звіт.docx)

[Приклади роботи](file:///C:/Users/Acer/Downloads/Telegram%20Desktop/ООП_Звіт.docx) [7](file:///C:/Users/Acer/Downloads/Telegram%20Desktop/ООП_Звіт.docx)

[Висновки](file:///C:/Users/Acer/Downloads/Telegram%20Desktop/ООП_Звіт.docx) [8](file:///C:/Users/Acer/Downloads/Telegram%20Desktop/ООП_Звіт.docx)

Ціль роботи

Вивчити особливості шаблонів функцій і шаблонів класів С++. Засвоїти принципи роботи шаблонів класів для роботи з базовими типами і користувацькими типами.

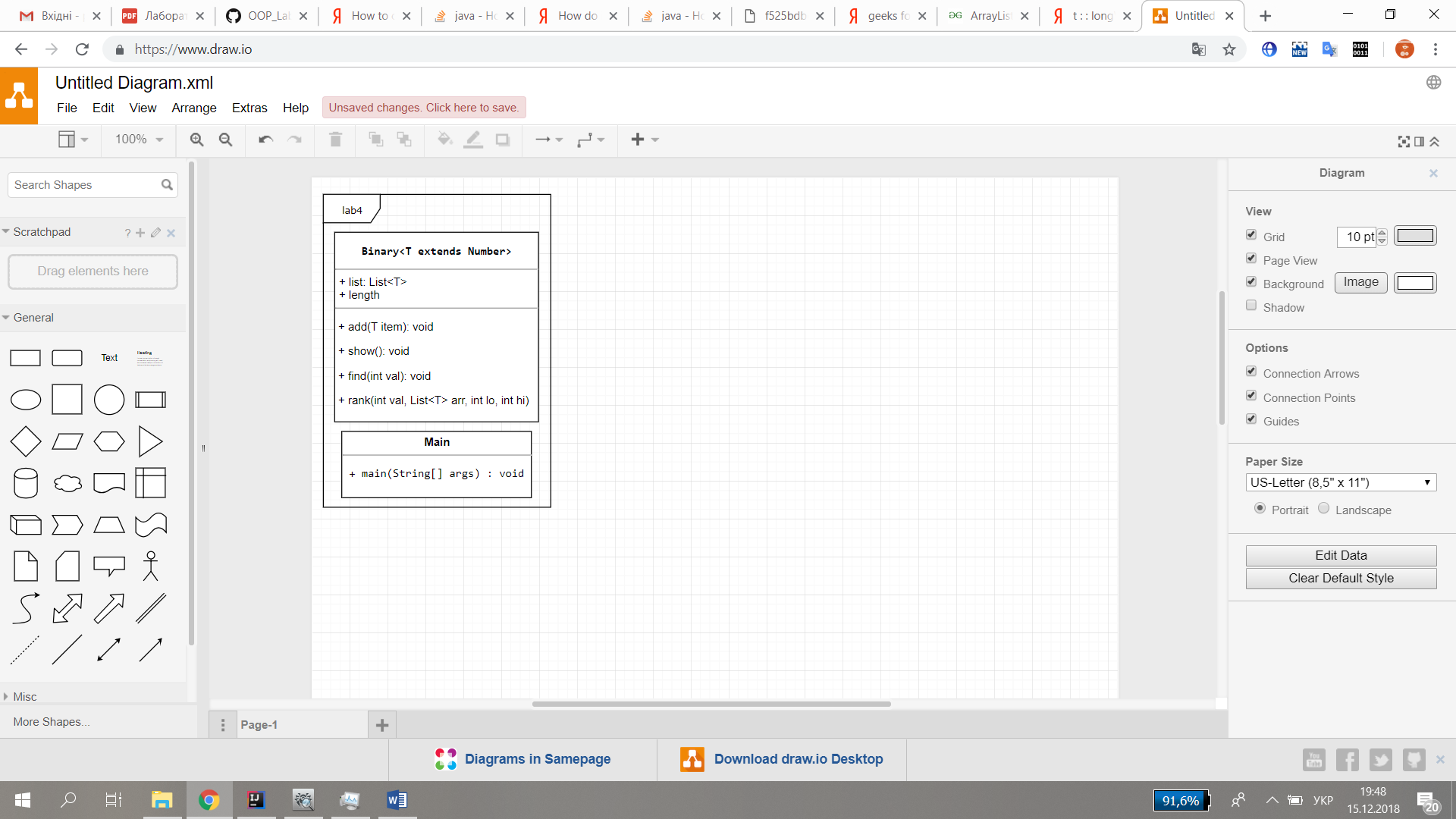
Постановка задачі

Створити наступний шаблон класу:

* масив елементів. Тип даних елементів масиву є параметризованим ( шаблонним );
* число елементів масиву;
* метод пошуку індексу елементу в масиві, згідно варіанту ( 13 варіант, бінарний пошук )
* конструктор для автоматичної ініціалізації унікального масиву елементів, заданої довжини;
* деструктор ( за необхідністю )

Продемонструвати роботу шаблону на різних базових цілочисельних типах даних.

UML-діаграма класів



Код програми

public class Main {

public static void main(String[] args) {

byte a[] = {

3, 25, 23,

0, 2, 1, 6,

40, 53, 23,

123, 50, 23

};

Binary<Byte> bytes = new Binary<>();

Binary<Integer> integers = new Binary<>();

Binary<Long> longes = new Binary<>();

for (byte k: a) {

bytes.add(k);

integers.add((int) k);

longes.add((long) k);

}

System.out.println("Byte:");

bytes.show();

bytes.find(19);

System.out.println("Integer:");

integers.show();

integers.find(25);

System.out.println("Long:");

longes.show();

longes.find(1);

}

}

import java.util.ArrayList;

import java.util.Comparator;

import java.util.List;

class Binary<T extends Number> {

List<T> list = new ArrayList<T>();

int length;

public void add(T item){

list.add(item);

length++;

list.sort(Comparator.comparing(T::longValue));

};

public void show(){

for (T item:list) {

System.out.print(item.toString() + ", ");

}

}

public void find(int val) {

int res = rank(val, list, 0, list.size()-1);

if ( res == -1 )

System.out.println("Значення " + val +

" відсутнє у масиві.");

else

System.out.println("Число " + val

+ " знаходиться на " + res + " позиції.");

}

private int rank(int val, List<T> arr, int lo, int hi) {

if (lo > hi) return -1;

int mid = lo + (hi - lo) / 2;

if (val < arr.get(mid).longValue()) {

return rank(val, arr, lo, mid - 1);

} else if (val > arr.get(mid).longValue()) {

return rank(val, arr, mid + 1, hi);

} else {

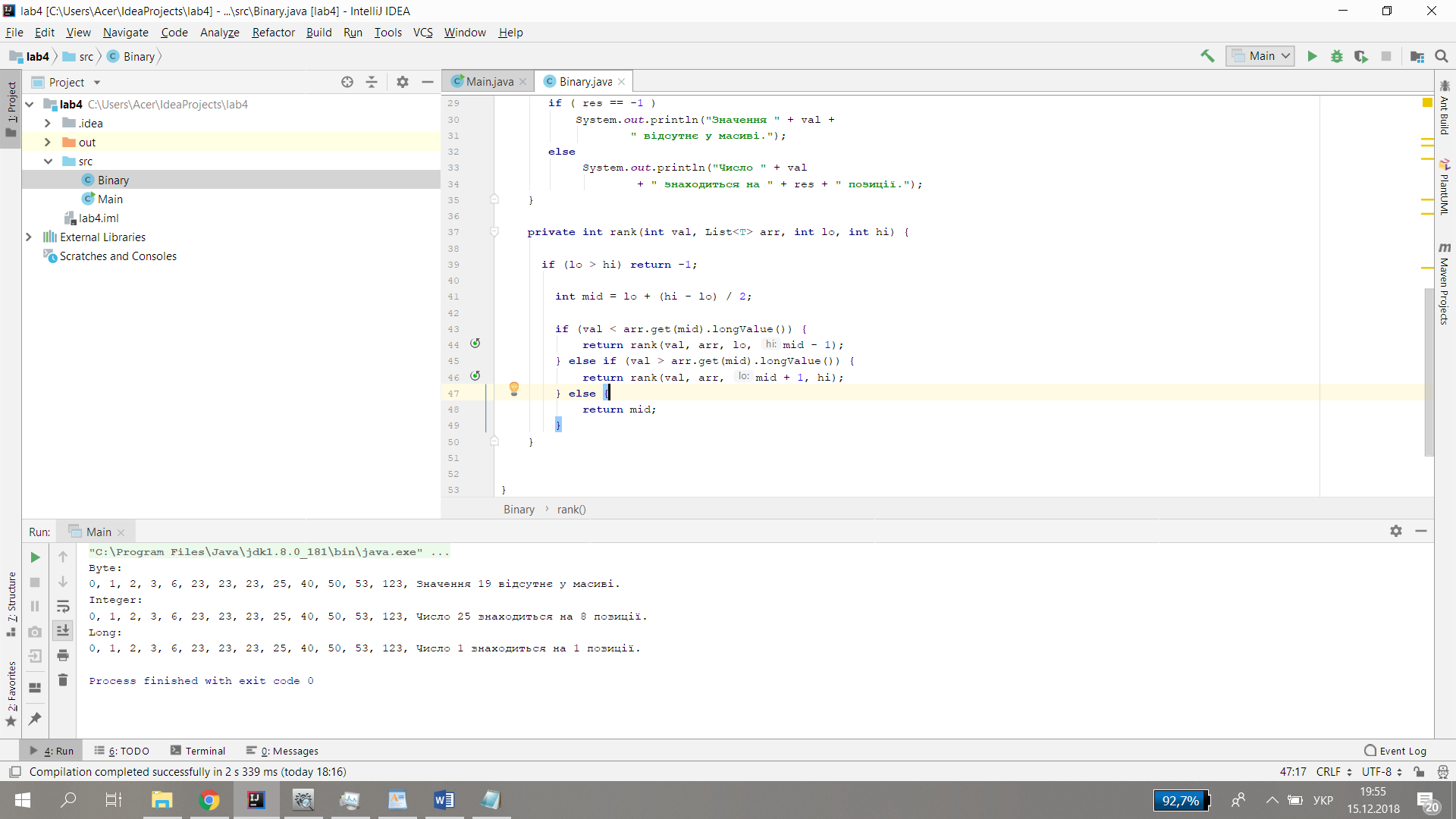
return mid;

}

}

}

Приклади роботи



Висновки

У ході даної лабораторної роботи було створено шаблон класу, визначено число елементів масиву, а також створено його публічні та приватні методи для бінарного пошуку. Шаблон класу перевірений на базових типах long, int та byte. Отримана програма пройшла всі підібрані тести, тобто коректно приймає будь-які цілочисельні типи і робить бінарний пошук в масиві даного типу.