# Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 5 з дисципліни

«Основи програмування-2.

Базові конструкції»

«Успадкування та поліморфізм»

Варіант 20

Виконав студент Лошак Віктор Іванович

(шифр, прізвище, ім'я, по батькові)

Перевірив Вітковська Ірина Іванівна

( прізвище, ім'я, по батькові)

Київ 2022

**Лабораторна робота 5**

**Мета:**

Вивчити механізми створення і використання класів і об’єктів.

**Тема:** Створити клас TIntNumber який представляє число у будь-якій системі числення і містить методи для інкрементування\декрементування числа та переведення числа у десяткову систему числення. На основі цього класу створити класи нащадки TIntNumber2 та TIntNumber16, що представляють двійкові та шістнадцяткові числа. Створити m двійкових та n шістнадцяткових чисел. Двійкові числа інкрементувати, шістнадцяткові декрементувати. Перевести числа у десяткові і знайти найбільше з них.

**Математична модель:**

Для імплементації чисел використаємо класи з назвою TIntNumber що міститиме відповідно методи: друкування числа, перевантажені оператори ++,--; а також конструктори з різними типами параметрів. Для імплементації відношення «є» між відповідними класами використаємо механізм наслідування . Для збереження числа у відповідній формі числення використаємо рядок з подальшою його конвертацією до числа в певній системі числення. Для реалізації заданих методів використаємо стандартні доступні бібліотеки та модулі.

**1)Приклад коду на C#**

**Program.cs**

using System;

using System.Collections.Generic;

namespace Lab5\_CSharp\_

{

internal class Program

{

static void Main(string[] args)

{

TIntNumber[] binary = ConsoleReadFabricMethod(typeof(TIntNumber2));

TIntNumber[] hexodecimal = ConsoleReadFabricMethod(typeof(TIntNumber16));

binary.ToList().ForEach(x => x++);

hexodecimal.ToList().ForEach(x => x--);

Console.WriteLine("Number arrays after element modification: ");

Print(binary);

Print(hexodecimal);

var allNumbers = binary.Select(x => x.ConvertToDecimal()).ToList();

allNumbers.AddRange(hexodecimal.Select(x => x.ConvertToDecimal()).ToList());

int indMax = allNumbers.IndexOf(allNumbers.Max());

if (indMax < binary.Length)

{

Console.Write($"The largest element was found in binary array with index {indMax} : ");

binary[indMax].Print();

}

else

{

Console.Write($"The largest element was found in hexodecimal array with index {indMax} : ");

hexodecimal[indMax-binary.Length].Print();

}

}

static void Print(TIntNumber[] arr)

{

for (int i = 0; i < arr.Length; i++)

{

arr[i].Print();

}

Console.Write("\n");

}

static TIntNumber[] ConsoleReadFabricMethod(Type readableType)

{

Console.WriteLine($"Enter {readableType.ToString()} : ");

string[] m = Console.ReadLine().Split(' ');

TIntNumber[] result = new TIntNumber[m.Length];

if (readableType == typeof(TIntNumber2))

{

for (int i = 0; i < m.Length; i++)

{

result[i] = new TIntNumber2(m[i]);

}

}

else if (readableType == typeof(TIntNumber16))

{

for (int i = 0; i < m.Length; i++)

{

result[i] = new TIntNumber16(m[i]);

}

}

else

{

throw new Exception("Wrong type was passed for reading");

}

return result;

}

}

}

**TIntNumber.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab5\_CSharp\_

{

internal class TIntNumber

{

#region Fields

#endregion

#region Properties

public string Value { get; set; }

public int Base { get; set; }

#endregion

#region Constructors

protected TIntNumber(string v, int b)

{

Value = v;

Base = b;

}

#endregion

#region Methods

public static TIntNumber operator++(TIntNumber intNumber)

{

int number = Convert.ToInt32(intNumber.Value, intNumber.Base);

number++;

intNumber.Value = Convert.ToString(number, intNumber.Base);

return intNumber;

}

public static TIntNumber operator--(TIntNumber intNumber)

{

int number = Convert.ToInt32(intNumber.Value, intNumber.Base);

number--;

intNumber.Value = Convert.ToString(number, intNumber.Base);

return intNumber;

}

public int ConvertToDecimal()

{

int number = Convert.ToInt32(this.Value, this.Base);

return number;

}

public void Print() => Console.Write($"({Value} - {ConvertToDecimal()}) ");

#endregion

}

}

**TIntNumber2.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab5\_CSharp\_

{

internal class TIntNumber2: TIntNumber

{

#region Fields

#endregion

#region Constructors

public TIntNumber2(string v):base(v, 2){}

#endregion

#region Methods

#endregion

}

}

**TIntNumber16.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab5\_CSharp\_

{

internal class TIntNumber16: TIntNumber

{

#region Fields

#endregion

#region Constructors

public TIntNumber16(string v) : base(v,16) { }

#endregion

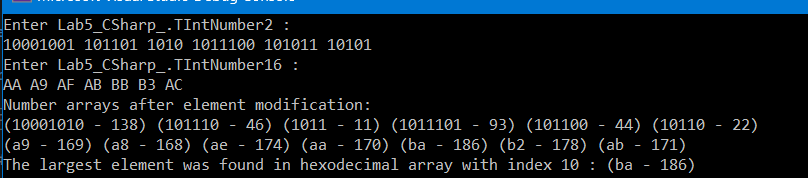
#region Methods

#endregion

}

}

**Виконання коду на C#:**



**Висновок:**

Отже в ході даної лабораторної роботи ми дослідили механізм наслідування в компільованій мові C# а також навчилися працювати з лб’єктами класів що містять перевантажені оператори. Дослідили функціонал бібліотек та ближче познайомилися з концепцією ООП.