

Додаток 1

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 № 4 з дисципліни
«Основи програмування-2.
Модульне програмування»

«Успадкування та поліморфізм»

Варіант 20

Виконав студент ІІ-13, Лисенко Анастасія Олегівна
(шифр, прізвище, ім'я, по батькові)

Перевірив Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Лабораторна робота 4 Успадкування та поліморфізм

Мета – вивчити механізми створення і використання класів та об'єктів.

Варіант 20

Постановка задачі.

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.

Код на мові C++:

lab5.cpp:

```
#include "lib.h"

int main()
{
    int m, n, up, down;

    cout << "Enter the amount of binary numbers m: ";
    input(m, down, up);

    TIntNumber2* BinaryArr = GenarateRandomBinNumbers(m, up, down);
    cout << endl << "Randomly generated binary numbers:" << endl;
    PrintArray(BinaryArr, m);

    cout << "Enter the amount of hex numbers n: ";
    input(n, down, up);

    TIntNumber16* HexArr = GenarateRandomHexNumbers(n, up, down);
    cout << endl << "Randomly generated hex numbers:" << endl;
    PrintArray(HexArr, n);

    cout << "Those numbers in decimal system:" << endl<<endl;

    cout << "Binary array into decimal:" << endl;
    PrintArrayDec(BinaryArr, m);
    cout << "Hex array into decimal:" << endl;
    PrintArrayDec(HexArr, n);
    cout << "_____ " << endl;

    cout << "Incremented and Decremented numbers:" << endl << endl;
    IncrementAndDecrement(BinaryArr, HexArr, m, n);

    cout << "Those numbers in decimal system:" << endl << endl;
```

```
cout << "Binary array into decimal:" << endl;
PrintArrayDec(BinaryArr, m);
cout << "Hex array into decimal:" << endl;
PrintArrayDec(HexArr, n);
cout << "_____ " << endl;

cout<<"The biggest number in decimal system: "<< FindTheBiggestDecNumber(BinaryArr, HexArr,
m, n)<<endl<<endl;

vector <string> numbers = IndexesOfBiggest(BinaryArr, HexArr, m, n);
cout << "Searched number in its system/s:" << endl<<endl;
for (size_t i = 0; i < numbers.size(); i+=2)
{
    cout << numbers[i] << " is the biggest number and it was written in " << numbers[i +
1] << " system"<<endl;
}

return 0;
}
```

myLib.h:

```
#pragma once
#include <iostream>
#include <vector>
#include <string>

using namespace std;

class TIntNumber {
public:
    string number;
    int base;
public:

    virtual void operator++() = 0;
    virtual void operator--() = 0;

    int TIntNumberToDecimal();

    void print();
};

class TIntNumber2:public TIntNumber {
public:
    TIntNumber2() { this->number = ""; this->base = 2;};
    TIntNumber2(string num);
    string DecToBin(int num);
    void operator++() override;
    void operator--() override;

};

TIntNumber2* GenarateRandomBinNumbers(int, int, int);

class TIntNumber16:public TIntNumber {
public:
    TIntNumber16() { this->number = ""; this->base = 16;};
    TIntNumber16(string num);
    string DecToHex(int num);
    void operator--() override;
    void operator++() override;

};
```

```
TIntNumber16* GenerateRandomHexNumbers(int, int, int);
void PrintArrayDec(TIntNumber*, int);
void PrintArray(TIntNumber*, int);
int FindTheBiggestDecNumber(TIntNumber*, TIntNumber*, int, int);
vector<string> IndexesOfBiggest(TIntNumber*, TIntNumber*, int, int);
void IncrementAndDecrement(TIntNumber*, TIntNumber*, int, int);
void input(int&, int&, int&);
```

lib.cpp:

```
#include "lib.h"

void TIntNumber::print() {
    cout << this->number<<" ";
}

void input(int& size, int& down, int& up) {
    cin >> size;
    while (size <= 0 || cin.fail())
    {
        cin.clear();
        cin.ignore(INT_MAX, '\n');
        cout << "Enter correct amount:";
        cin >> size;
    }
    cout << "Enter down limit for generated numbers: ";
    cin >> down;
    while (cin.fail())
    {
        cin.clear();
        cin.ignore(INT_MAX, '\n');
        cout << "Enter correct number:";
        cin >> down;
    }

    cout << "Enter up limit for generated numbers: ";
    cin >> up;
    while (cin.fail() || up < down)
    {
        cin.clear();
        cin.ignore(INT_MAX, '\n');
        cout << "Enter correct number:";
        cin >> up;
    }
}

string TIntNumber16::DecToHex(int num)
{
    string str;
    int temp = (num <= 0 ? (-num) : num);
    while (temp != 0) {
        str = (((temp % 16) < 10) ? char(temp % 16 + 48) : char(temp % 16 + 55)) + str; temp
/= 16;
    }
    if (num < 0)
        str = "-" + str;
    if (num == 0)
        str = "0";
    return str;
}
```

```

string TIntNumber2::DecToBin(int num) {
    int temp = num;
    string r = "";
    while (num != 0)

    {
        r = (num % 2 == 0 ? "0" : "1") + r; num /= 2;
    }
    if (temp < 0)
        r = "-" + r;
    if (temp == 0)
        r = "0";
    return r;
}
TIntNumber2::TIntNumber2(string num) {
    this->number = num;
    this->base = 2;
}
TIntNumber16::TIntNumber16(string num) {
    this->number = num;
    this->base = 16;
}
void TIntNumber16::operator--() {
    int new_number = TIntNumberToDecimal();
    new_number--;
    this->number = DecToHex(new_number);
}
void TIntNumber16::operator++() {
    int new_number = TIntNumberToDecimal();
    new_number++;
    this->number = DecToHex(new_number);
}
void TIntNumber2::operator++() {
    int new_number = TIntNumberToDecimal();
    new_number++;
    this->number = DecToBin(new_number);
}
void TIntNumber2::operator--() {
    int new_number = TIntNumberToDecimal();
    new_number--;
    this->number = DecToBin(new_number);
}
int TIntNumber::TIntNumberToDecimal() {
    int new_number = stoi(number, nullptr, base);
    return new_number;
}
TIntNumber2* GenarateRandomBinNumbers(int m, int up, int down) {
    srand(time(NULL));
    TIntNumber2* array = new TIntNumber2[m];
    TIntNumber2 object;
    for (size_t i = 0; i < m; i++)
    {
        object.number = object.DecToBin((rand() % (up - down + 1)) + down);
        array[i] = object;
    }
    return array;
}
TIntNumber16* GenarateRandomHexNumbers(int n, int up, int down) {
    srand(time(NULL));
    TIntNumber16* array = new TIntNumber16[n];
    TIntNumber16 object;
    for (size_t i = 0; i < n; i++)
    {
        object.number = object.DecToHex((rand() % (up - down + 1)) + down);
        array[i] = object;
    }
}

```

```

        return array;
    }
    void IncrementAndDecrement(TIntNumber* a, TIntNumber* b, int m, int n) {
        cout << "Decrementated binary numbers:" << endl;
        for (size_t i = 0; i < m; i++)
        {
            --a[i];
            cout << a[i].number << " ";
        }cout << endl;
        cout << "Incrementated hex numbers:" << endl;
        for (size_t i = 0; i < n; i++)
        {
            ++b[i];
            cout << b[i].number << " ";
        }cout << endl;
        cout << "_____ " << endl;
    }
    void PrintArrayDec(TIntNumber* a, int size) {

        for (size_t i = 0; i < size; i++)
        {
            cout << a[i].TIntNumberToDecimal() << " ";
        }cout << endl;
    }
    void PrintArray(TIntNumber* a, int size) {

        for (size_t i = 0; i < size; i++)
        {
            a[i].print();
        }cout << endl << "_____ " << endl << endl;
    }
    int FindTheBiggestDecNumber(TIntNumber* a, TIntNumber* b, int m, int n) {
        int max = a[0].TIntNumberToDecimal();
        for (size_t i = 0; i < m; i++)
        {
            if (a[i].TIntNumberToDecimal() > max)
                max = a[i].TIntNumberToDecimal();
        }
        for (size_t i = 0; i < n; i++)
        {
            if (b[i].TIntNumberToDecimal() > max)
                max = b[i].TIntNumberToDecimal();
        }
        return max;
    }
    vector<string> IndexesOfBiggest(TIntNumber* a, TIntNumber* b, int m, int n) {
        vector<string> numbers;
        int max = FindTheBiggestDecNumber(a,b, m, n);
        for (size_t i = 0; i < m; i++)
        {
            if (a[i].TIntNumberToDecimal() == max)
            {
                numbers.push_back(a[i].number);
                numbers.push_back("binary");
                break;
            }
        }

        for (size_t i = 0; i < n; i++)
        {
            if (b[i].TIntNumberToDecimal() == max)
            {
                numbers.push_back(b[i].number);
                numbers.push_back("hex");
                break;
            }
        }
        return numbers;
    }

```

Консоль:

```
Enter the amount of binary numbers m: 5
Enter down limit for generated numbers: 1
Enter up limit for generated numbers: 10
```

Randomly generated binary numbers:

```
1001 1001 111 110 1010
```

```
Enter the amount of hex numbers n: 5
Enter down limit for generated numbers: 1
Enter up limit for generated numbers: 10
```

Randomly generated hex numbers:

```
2 5 7 5 1
```

Those numbers in decimal system:

Binary array into decimal:

```
9 9 7 6 10
```

Hex array into decimal:

```
2 5 7 5 1
```

Incremented and Decrementated numbers:

Decrementated binary numbers:

```
1000 1000 110 101 1001
```

Incrementated hex numbers:

```
3 6 8 6 2
```

Those numbers in decimal system:

Binary array into decimal:

```
8 8 6 5 9
```

Hex array into decimal:

```
3 6 8 6 2
```

The biggest number in decimal system: 9

Searched number in its system/s:

1001 is the biggest number and it was written in binary system

Код на мові Python:

main.py:

```
from classes import *

m = int(input("Enter the amount of binary numbers m: "))

down = int(input("Enter down limit for generated numbers: "))
up = int(input("Enter up limit for generated numbers: "))

BinaryArr = GenarateRandomBinNumbers(m, up, down)
print("\nRandomly generated binary numbers:")
PrintArray(BinaryArr)

n = int(input("Enter the amount of hex numbers m: "))

down = int(input("Enter down limit for generated numbers: "))
up = int(input("Enter up limit for generated numbers: "))

HexArr = GenarateRandomHexNumbers(n, up, down)
print("\nRandomly generated hex numbers:")
PrintArray(HexArr)

print("Those numbers in decimal system:\n")

print("Binary array into decimal:")
PrintArrayDec(BinaryArr)
print("Hex array into decimal:")
PrintArrayDec(HexArr)
print("_____")

print("Incremented and Decrementated numbers:\n")
IncrementAndDecrement(BinaryArr, HexArr)

print("Those numbers in decimal system:\n")

print("Binary array into decimal:")
PrintArrayDec(BinaryArr)
print("Hex array into decimal:")
PrintArrayDec(HexArr)
print("_____")

print("The biggest number in decimal system: ", FindTheBiggestDecNumber(BinaryArr, HexArr))

numbers = IndexesOfBiggest(BinaryArr, HexArr)
print("Searched number in its system/s:\n")

for i in range(0, len(numbers), 2):
    print(numbers[i], "is the biggest number and it was written in", numbers[i+1], "system")
```

classes.py

```
from abc import ABC, abstractmethod
from random import randint

class TIntNumber(ABC):
    def __init__(self, number, base):
        self.number = number
        self.base = base
```



```

def TIntToDecimal(self):
    new_number = int(self.number, self.base)
    return new_number

@abstractmethod
def __iadd__(self, other):
    pass

@abstractmethod
def __isub__(self, other):
    pass

def output(self):
    print(self.number, end=" ")

class TIntNumber2(TIntNumber):
    def __init__(self, number):
        super().__init__(number, 2)

    def __iadd__(self, other):
        new_number = self.TIntToDecimal()
        new_number += other
        self.number = bin(new_number)
        return self

    def __isub__(self, other):
        new_number = self.TIntToDecimal()
        new_number -= other
        self.number = bin(new_number)
        return self

class TIntNumber16(TIntNumber):
    def __init__(self, number):
        super().__init__(number, 16)

    def __iadd__(self, other):
        new_number = self.TIntToDecimal()
        new_number += other
        self.number = hex(new_number)
        return self

    def __isub__(self, other):
        new_number = self.TIntToDecimal()
        new_number -= other
        self.number = hex(new_number)
        return self

def GenarateRandomBinNumbers(m, up, down):
    BinaryArr = [TIntNumber2(None)]*m
    for i in range(m):
        object = TIntNumber2(None)
        object.number = bin(randint(down, up))
        BinaryArr[i] = object
    return BinaryArr

def GenarateRandomHexNumbers(n, up, down):
    HexArr = [TIntNumber16(None)]*n
    for i in range(n):
        object = TIntNumber16(None)
        object.number = hex(randint(down, up))
        HexArr[i] = object
    return HexArr

```

```
def IncrementAndDecrement(BinaryArr, HexArr):
    print("Decrementing binary numbers:")
    for i in range(len(BinaryArr)):
        BinaryArr[i] -= 1
        print(BinaryArr[i].number, end=" ")
    print()
    print("Incrementing hex numbers:")
    for i in range(len(HexArr)):
        HexArr[i] += 1
        print(HexArr[i].number, end=" ")
    print()
    print("_____")

def PrintArray(Arr):
    for i in range(len(Arr)):
        Arr[i].output()
    print("\n_____")

def PrintArrayDec(Arr):
    for i in range(len(Arr)):
        print(Arr[i].TIntToDecimal(), end=" ")
    print()

def FindTheBiggestDecNumber(BinaryArr, HexArr):
    max = BinaryArr[0].TIntToDecimal()
    for i in range(len(BinaryArr)):
        if BinaryArr[i].TIntToDecimal() > max:
            max = BinaryArr[i].TIntToDecimal()
    for i in range(len(HexArr)):
        if HexArr[i].TIntToDecimal() > max:
            max = HexArr[i].TIntToDecimal()
    return max

def IndexesOfBiggest(BinaryArr, HexArr):
    numbers = []
    max = FindTheBiggestDecNumber(BinaryArr, HexArr)
    for i in range(len(BinaryArr)):
        if BinaryArr[i].TIntToDecimal() == max:
            numbers.append(BinaryArr[i].number)
            numbers.append("binary")
            break
    for i in range(len(HexArr)):
        if HexArr[i].TIntToDecimal() == max:
            numbers.append(HexArr[i].number)
            numbers.append("hex")
            break
    return numbers
```

Консоль:

```
Enter the amount of binary numbers m: 5
Enter down limit for generated numbers: 1
Enter up limit for generated numbers: 10
```

```
Randomly generated binary numbers:
0b100 0b100 0b1010 0b11 0b10
```

```
Enter the amount of hex numbers m: 5
Enter down limit for generated numbers: 1
Enter up limit for generated numbers: 10
```

```
Randomly generated hex numbers:
0x8 0x7 0x3 0x7 0x5
```

Those numbers in decimal system:

Binary array into decimal:

4 4 10 3 2

Hex array into decimal:

8 7 3 7 5

Incremented and Decrementd numbers:

Decrementd binary numbers:

0b11 0b11 0b1001 0b10 0b1

Incremented hex numbers:

0x9 0x8 0x4 0x8 0x6

Those numbers in decimal system:

Binary array into decimal:

3 3 9 2 1

Hex array into decimal:

9 8 4 8 6

The biggest number in decimal system: 9

Searched number in its system/s:

0b1001 is the biggest number and it was written in binary system

0x9 is the biggest number and it was written in hex system

1. Висновки

Отже в ході даної лабораторної роботи ми дослідили механізм наслідування в компільованій мові C++ та інтерпретованій мові Python, а також навчилися працювати з об'єктами класів що містять перевантажені оператори. Дослідили функціонал бібліотек та ближче познайомилися з концепцією ООП.