

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. Методології програмування»
«Успадкування та поліморфізм»

Варіант 18

Виконав студент

ІП-13 Король Валентин Олегович
(шифр, прізвище, ім'я, по батькові)

Перевірів

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

Лабораторна робота 5

Варіант 18

Створити клас `TIntNumber`, який представляє ціле число у будь-якій системі числення і містить методи для додавання іншого числа, порівняння чисел та переведення числа у вказану систему числення (із заданого переліку). На основі цього класу створити класи-нащадки `TIntNumber2` та `TIntNumber8`, що представляють двійкові та вісімкові числа. Створити m двійкових та n вісімкових чисел. Знайти суму двійкових та вісімкових чисел. Перевести отримане значення суми вісімкових чисел у двійковий формат. Визначити, сума яких чисел менша (двійкових чи вісімкових)

Код програми

C++

main.cpp

```
#include "Functions.h"
using namespace std;

int main()
{
    int m, n;
    cout << "Enter m (number of binary numbers): "; cin >> m;
    cout << "Enter n (number of octal numbers): "; cin >> n;

    cout << "\n---Start entering binary numbers---\n\n";
    vector<TIntNumber2> binNums = inputNums<TIntNumber2>(m, 2);
    cout << "\n---Start entering octal numbers---\n\n";
    vector<TIntNumber8> octNums = inputNums<TIntNumber8>(n, 8);

    cout << "\n---The data you entered---\n";
    cout << "\nBinary numbers: ";
    outputNums<TIntNumber2>(binNums);
    cout << "\nOctal numbers: ";
    outputNums<TIntNumber8>(octNums);

    TIntNumber2 sumBin = countSum<TIntNumber2>(binNums);
    TIntNumber8 sumOct = countSum<TIntNumber8>(octNums);
    cout << "\n\nThe sum of binary numbers: " << sumBin;
    cout << "\nThe sum of octal numbers: " << sumOct;
    TIntNumber2 sumOctInBin = sumOct.convertToBase();
    cout << "\nThe value of the sum of octal numbers in binary format: " << sumOctInBin <<
    "\n";

    findMin(sumBin, sumOctInBin);

    system("pause");
    return 0;
}
```

Functions.h

```
#pragma once
#include <iostream>
#include <iomanip>
#include <stdio.h>
```

```

#include <vector>
#include <math.h>
#include <Windows.h>
#include "Classes.h"
using namespace std;

class TIntNumber2;
class TIntNumber8;

bool isNumeralBase(long long num, int base);
long long convertToDec(long long num, int base);
long long convertFromDec(long long dec, int base);
void findMin(TIntNumber2& bins, TIntNumber2& ocs);

template <class T1>
vector<T1> inputNums(int k, int base)
{
    vector<T1> nums(k);
    for (int i = 0; i < k; ++i) {
        cout << "Enter number: ";
        int num;
        cin >> num;
        if (isNumeralBase(num, base)) {
            nums[i] = T1(num);
        }
        else {
            cout << "You have entered an incorrect number! Try again!\n";
            --i;
        }
    }
    return nums;
}

template <class T2>
void outputNums(const vector<T2>& nums)
{
    for (size_t i = 0; i < nums.size(); ++i) {
        cout << nums[i] << " ";
    }
}

template <class T3>
T3& countSum(vector<T3>& nums)
{
    T3 sum = 0;
    for (size_t i = 0; i < nums.size(); ++i) {
        sum.add(nums[i]);
    }
    return sum;
}

```

TDate.cpp

Тестування:

```
Microsoft Visual Studio Debug Console
Enter m (number of binary numbers): 3
Enter n (number of octal numbers): 2

---Start entering binary numbers---

Enter number: 0101000
Enter number: 00011
Enter number: 100

---Start entering octal numbers---

Enter number: 314564
Enter number: 4235

---The data you entered---

Binary numbers: 101000 11 100
Octal numbers: 314564 4235

The sum of binary numbers: 101111
The sum of octal numbers: 321021
The value of the sum of octal numbers in binary format: 11010001000010001

The sum of binary numbers is less than octal
Press any key to continue . . .

C:\Users\valik\source\repos\lab2.5cpp\Debug\lab2.5cpp.exe (process 16408) exited with code 0.
Press any key to close this window . . .
```


Python

main.py

```
from classes import *
from functions import inputNums, outputNums, countSum, findMin

m = int(input("Введіть m (кількість двійкових чисел): "))
n = int(input("Введіть n (кількість вісімкових чисел): "))
print("\n---Початок введення двійкових чисел---\n")
binNums = inputNums(m, 2)
print("\n---Початок введення вісімкових чисел---\n")
octNums = inputNums(n, 8)

print("\n---Введені вами дані---\n")
print("Двійкові числа: ", end = "")
outputNums(binNums)
print("Вісімкові числа: ", end = "")
outputNums(octNums)

sumBins = countSum(binNums, 2)
sumOcts = countSum(octNums, 8)
print("\nСума двійкових чисел: " + str(sumBins.getNumber()))
print("Сума вісімкових чисел: " + str(sumOcts.getNumber()))
sumOctInBin = TIntNumber2(sumOcts.convertToBase())
print("Отримане значення суми вісімкових чисел у двійковому форматі: " +
str(sumOctInBin.getNumber()))
findMin(sumBins, sumOctInBin)
```

Functions.py

```

from classes import *

def isNumeralBase(num, base):
    while num:
        if num % 10 < base:
            num = num // 10
        else:
            return False
    return True

def inputNums(k, base):
    nums = []
    i = 0
    while i < k:
        num = int(input("Введіть число: "))
        if isNumeralBase(num, base):
            if base == 2:
                obj = TIntNumber2(num)
            elif base == 8:
                obj = TIntNumber8(num)
            else:
                continue
            nums.append(obj)
        else:
            print("Ви ввели некоректне число! Спробуйте ще раз!")
            i -= 1
        i += 1
    return nums

def outputNums(nums):
    for i in range(0, len(nums), 1):
        print(nums[i].getNumber, end = " ")
    print()
    return

def countSum(nums, base):
    if base == 2:
        sum = TIntNumber2()
    elif base == 8:
        sum = TIntNumber8()
    else:
        return 0
    for i in range(0, len(nums), 1):
        sum = sum.add(nums[i])
    return sum

def findMin(bins : TIntNumber2, octs : TIntNumber2):
    flag = bins.compare(octs)
    if flag < 0:
        print("\nCума двійкових чисел менша, ніж вісімкових")
    elif flag == 0:
        print("\nCуми двійкових та вісімкових чисел рівні")
    else:
        print("\nCума вісімкових чисел менша, ніж двійкових")
    return

```

classes.py

```

from abc import ABC, abstractmethod
from converting import convertFromDec, convertToDec

```

```

class TIntNumber(ABC):
    def __init__(self, number = 0, base = 10):
        self.__number = number
        self.__base = base
    @property
    def getBase(self):
        return self.__base
    @property
    def getNumber(self):
        return self.__number

    @abstractmethod
    def add(self, obj : 'TIntNumber'):
        pass
    @abstractmethod
    def compare(self, obj : 'TIntNumber'):
        if self.__number < obj.__number:
            return -1
        elif self.__number == obj.__number:
            return 0
        else:
            return 1
    @abstractmethod
    def convertToBase(self, base):
        if self.__base == 10:
            return convertFromDec(self.__number, base)
        elif base == 10:
            return convertToDec(self.__number, self.__base)
        else:
            return convertFromDec(convertToDec(self.__number, self.__base), base)
    pass

```

```

class TIntNumber2(TIntNumber):
    def __init__(self, number = 0):
        super().__init__(number, 2)
    def add(self, obj : 'TIntNumber2'):
        sum = TIntNumber2(convertFromDec(convertToDec(self.getNumber, 2) +
convertToDec(obj.getNumber, 2), 2))
        return sum
    def compare(self, obj : 'TIntNumber2'):
        return super().compare(obj)
    def convertToBase(self, base = 8):
        return super().convertToBase(base)
    pass

```

```

class TIntNumber8(TIntNumber):
    def __init__(self, number = 0):
        super().__init__(number, 8)
    def add(self, obj : 'TIntNumber8'):
        sum = TIntNumber8(convertFromDec(convertToDec(self.getNumber, 8) +
convertToDec(obj.getNumber, 8), 8))
        return sum
    def compare(self, obj : 'TIntNumber8'):
        return super().compare(obj)
    def convertToBase(self, base = 2):
        return super().convertToBase(base)
    pass

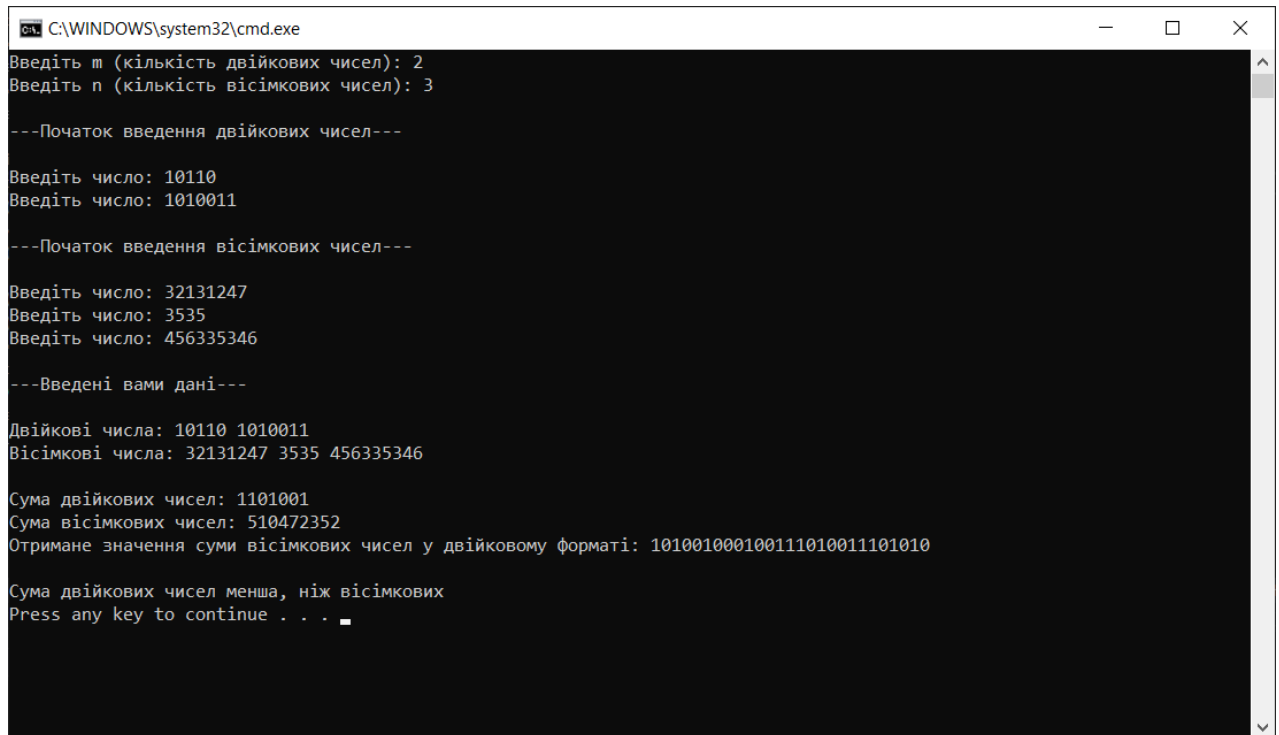
```


converting.py

```
def convertFromDec(dec, base):
    num = 0
    i = 0
    while dec > base - 1:
        num += (dec % base) * 10**i
        i += 1
        dec = dec // base
    return num + dec * 10**i

def convertToDec(num, base):
    dec = 0
    i = 0
    while num:
        dec += (num % 10) * base**i
        i += 1
        num = num // 10
    return dec
```

Тестування:



```
C:\WINDOWS\system32\cmd.exe
Введіть m (кількість двійкових чисел): 2
Введіть n (кількість вісімкових чисел): 3

---Початок введення двійкових чисел---

Введіть число: 10110
Введіть число: 1010011

---Початок введення вісімкових чисел---

Введіть число: 32131247
Введіть число: 3535
Введіть число: 456335346

---Введені вами дані---

Двійкові числа: 10110 1010011
Вісімкові числа: 32131247 3535 456335346

Сума двійкових чисел: 1101001
Сума вісімкових чисел: 510472352
Отримане значення суми вісімкових чисел у двійковому форматі: 101001000100111010011101010

Сума двійкових чисел менша, ніж вісімкових
Press any key to continue . . .
```

Висновки:

Я вивчив та використав на практиці механізми створення класів та об'єктів.