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. Модульне програмування»

«Успадкування та поліморфізм»

Варіант 17

Виконав студент ІП-11 Куценко Артемій Ілліч

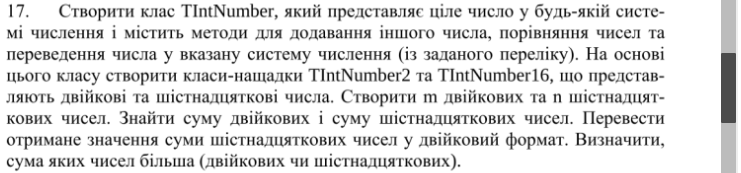
(шифр, прізвище, ім'я, по батькові)

Перевірив Вітковська Ірина Іванівна

(прізвище, ім'я, по батькові)

Київ 2022

**Умова задачі:**



**Код на С++:**

**main.cpp**

#include "pch.h"

int main() {

int m, n;

cout << "Input m: ";

cin >> m;

TIntNumber2\* M = genArrBin(m);

cout << "\nInput n: ";

cin >> n;

TIntNumber16\* N = genArrHex(n);

TIntNumber2 binSum = findSumBin(M,m);

TIntNumber16 hexSum = findSumHex(N,n);

hexSum.changeSystem(2);

cout << "\nHex sum in binary form: " << hexSum.getNumber();

cout << "\n\n";

binSum.compare(&hexSum);

delete[] M;

delete[] N;

system("pause>0");

}

**Class.h**

#pragma once

#include <iostream>

#include <cmath>

#include <string>

using namespace std;

class TIntNumber {

protected:

int system;

int dec;

string number;

public:

int getDec();

string getNumber();

virtual void changeSystem(int sys) = 0;

virtual void add(TIntNumber\* addition) = 0;

void compare(TIntNumber\* other);

};

class TIntNumber2 : public TIntNumber {

public:

void changeSystem(int sys);

void add(TIntNumber\* addition);

TIntNumber2();

TIntNumber2(string number);

};

class TIntNumber16 : public TIntNumber {

public:

void changeSystem(int sys);

void add(TIntNumber\* addition);

TIntNumber16();

TIntNumber16(string number);

};

**Class.cpp**

#include "pch.h"

int TIntNumber::getDec() {

return dec;

}

string TIntNumber::getNumber() {

return number;

}

TIntNumber2::TIntNumber2(string number) { // конструктор для класу TIntNumber2

this->number = number;

this->system = 2;

}

TIntNumber2::TIntNumber2() {

this->system = 2;

}

TIntNumber16::TIntNumber16(string number) { // конструктор для класу TIntNumber16

this->number = number;

this->system = 16;

}

TIntNumber16::TIntNumber16() {

this->system = 16;

}

void TIntNumber2::changeSystem(int sys) {

if (this->system == 2 && sys == 10) {

int dec = 0;

for (int i = 0; i < number.length(); i++) {

if (number[number.length() - i - 1] == '1') {

dec += pow(2, i);

}

}

this->dec = dec;

this->system = 10;

}

else if (this->system == 10 && sys == 2) {

string bin;

int whole = this->dec;

int rest;

do {

rest = whole % 2;

whole /= 2;

bin = to\_string(rest) + bin;

} while (whole);

this->number = bin;

this->system = 2;

}

}

void TIntNumber16::changeSystem(int sys) {

if (this->system == 10 && sys == 16) {

string hex;

int whole = this->dec;

int rest;

const char HEX[]{ '0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f' };

do {

rest = whole % 16;

whole /= 16;

hex = HEX[rest] + hex;

} while (whole);

this->number = hex;

this->system = 16;

}

else if (this->system == 16 && sys == 10) {

int dec = 0;

const char HEX[]{ '0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f' };

for (int i = 0; i < number.length(); i++) {

for (int j = 0; j < 16; j++) {

if (number[number.length() - i - 1] == HEX[j]) {

dec += j \* pow(16, i);

j = 16;

}

}

}

this->dec = dec;

this->system = 10;

}

else if (this->system == 16 && sys == 2) {

string bin;

const char HEX[]{ '0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f' };

const string BIN[]{ "0000", "0001", "0010", "0011", "0100", "0101", "0110", "0111", "1000", "1001", "1010", "1011", "1100", "1101", "1110", "1111" };

for (int i = 0; i < number.length(); i++) {

for (int j = 0; j < 16; j++) {

if (number[i] == HEX[j]) {

bin += BIN[j];

break;

}

}

}

while (bin[0] == '0' && bin.length() > 1) {

bin.erase(0, 1);

}

this->number = bin;

this->system = 2;

}

}

void TIntNumber2::add(TIntNumber\* addition) {

this->changeSystem(10);

addition->changeSystem(10);

this->dec = this->dec + addition->getDec();

this->changeSystem(2);

}

void TIntNumber16::add(TIntNumber\* addition) {

this->changeSystem(10);

addition->changeSystem(10);

this->dec = this->dec + addition->getDec();

this->changeSystem(16);

}

void TIntNumber::compare(TIntNumber\* other) {

if (this->dec == other->getDec()) {

cout << this->dec << " == " << other->getDec();

}

else if (this->dec < other->getDec()) {

cout << this->dec << " < " << other->getDec();

}

else {

cout << this->dec << " > " << other->getDec();

}

}

**Pch.h**

#pragma once

#include "class.h"

TIntNumber2\* genArrBin(int size);

TIntNumber16\* genArrHex(int size);

TIntNumber2 findSumBin(TIntNumber2\* M, int size);

TIntNumber16 findSumHex(TIntNumber16\* M, int size);

**Pch.cpp**

#include "pch.h"

TIntNumber2\* genArrBin(int size) {

string str;

TIntNumber2\* M = new TIntNumber2[size];

cout << "Input binary numbers: ";

for (int i = 0; i < size; i++) {

cin >> str;

TIntNumber2 binNumber(str);

M[i] = binNumber;

}

return M;

}

TIntNumber16\* genArrHex(int size) {

string str;

TIntNumber16\* M = new TIntNumber16[size];

cout << "Input hex numbers: ";

for (int i = 0; i < size; i++) {

cin >> str;

TIntNumber16 binNumber(str);

M[i] = binNumber;

}

cout << '\n';

return M;

}

TIntNumber2 findSumBin(TIntNumber2\* M, int size) {

TIntNumber2 sum("0");

cout << "\nYour row: ";

for (int i = 0; i < size; i++) {

cout << M[i].getNumber() << " ";

sum.add(&M[i]);

}

cout << "\nDecimal row: ";

for (int i = 0; i < size; i++) {

cout << M[i].getDec() << " ";

}

cout << "\nDecimal sum: " << sum.getDec();

cout << "\nBinary sum: " << sum.getNumber();

cout << '\n';

return sum;

}

TIntNumber16 findSumHex(TIntNumber16\* M, int size) {

TIntNumber16 sum("0");

cout << "\nYour row: ";

for (int i = 0; i < size; i++) {

cout << M[i].getNumber() << " ";

sum.add(&M[i]);

}

cout << "\nDecimal row: ";

for (int i = 0; i < size; i++) {

cout << M[i].getDec() << " ";

}

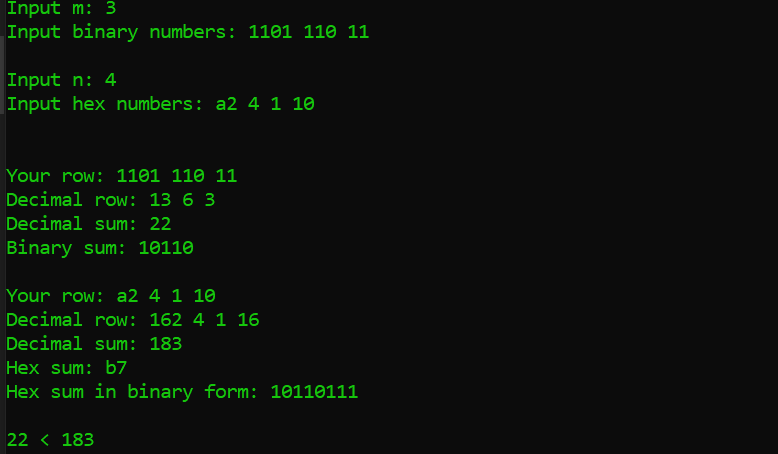
cout << "\nDecimal sum: " << sum.getDec();

cout << "\nHex sum: " << sum.getNumber();

return sum;

}

**Скріншот роботи програми на C++:**



**Код на Python:**

**Oplab5PY.py**

from funcs import\*

m = int(input("Input m: "))

M = genListBin(m)

n = int(input("\nInput n: "))

N = genListHex(n)

binSum = findSumBin(M,m)

hexSum = findSumHex(N,n)

hexSum.changeSystem(2)

print("Hex sum in binary form: " + hexSum.number)

print("\n\n")

binSum.compare(hexSum)

**funcs.py**

from class1 import\*

def genListBin(m: int):

number\_list=input("Input binary numbers: ").split(' ')

M = []

for i in range(m):

a = TIntNumber2(2, number\_list[i])

M.append(a)

return M

def genListHex(m: int):

number\_list=input("Input hex numbers: ").split(' ')

M = []

for i in range(m):

a = TIntNumber16(16, number\_list[i])

M.append(a)

return M

def findSumBin(M, m):

sum = TIntNumber2(2, "0")

print("\nYour row: ")

for i in range(m):

print(M[i].number, end=" ")

sum.add(M[i])

print("\nYour decimal row: ")

for i in range(m):

M[i].changeSystem(10)

print(M[i].dec, end = " ")

sum.changeSystem(10)

print("\nDecimal sum: " + sum.number)

sum.changeSystem(2)

print("Binary sum: " + sum.number)

print("\n\n")

return sum

def findSumHex(M, m):

sum = TIntNumber16(16, "0")

print("\nYour row: ")

for i in range(m):

print(M[i].number, end=" ")

sum.add(M[i])

print("\nYour decimal row: ")

for i in range(m):

M[i].changeSystem(10)

print(M[i].dec, end = " ")

sum.changeSystem(10)

print("\nDecimal sum: " + sum.number)

sum.changeSystem(16)

print("Hex sum: " + sum.number)

return sum

**Class.py**

from abc import abstractmethod

class TIntNumber:

def \_\_init\_\_(self, system, number):

self.system = system

self.number = number

self.dec = 0

@abstractmethod

def changeSystem(self, sys):

pass

def compare(self, c):

if (self.dec == c.dec):

print(self.number + " == " + c.number)

elif (self.dec > c.dec):

print(self.number + " > " + c.number)

elif (self.dec < c.dec):

print(self.number + " < " + c.number)

class TIntNumber2(TIntNumber):

def changeSystem(self, sys):

if (self.system == 2 and sys == 10):

self.dec = 0

for i in range(len(self.number)):

if (self.number[len(self.number) - i - 1] == '1'):

self.dec += 2\*\*i

self.number = str(self.dec)

self.system = 10

elif (self.system == 10 and sys == 2):

self.number = ''

whole = self.dec

rest = 0

while(whole):

rest = whole % 2

whole //= 2

self.number = str(rest) + self.number

self.system = 2

def add(self, addition):

self.changeSystem(10)

addition.changeSystem(10)

self.dec = self.dec + addition.dec

self.changeSystem(2)

class TIntNumber16(TIntNumber):

def changeSystem(self, sys):

HEX = ['0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f']

if (self.system == 16 and sys == 10):

self.dec = 0

for i in range(len(self.number)):

for j in range(16):

if (self.number[len(self.number) - i - 1] == HEX[j]):

self.dec += j \* (16\*\*i)

break

self.number = str(self.dec)

self.system = 10

elif (self.system == 10 and sys == 16):

self.number = ''

whole = self.dec

rest = 0

while(whole):

rest = whole % 16

whole //= 16

self.number = HEX[rest] + self.number

self.system = 16

elif (self.system == 16 and sys == 2):

BIN = ["0000", "0001", "0010", "0011", "0100", "0101", "0110", "0111", "1000", "1001", "1010", "1011", "1100", "1101", "1110", "1111"]

self.dec = 0

binary = ''

for i in range(len(self.number)):

for j in range(16):

if (self.number[len(self.number) - i - 1] == HEX[j]):

binary = BIN[j] + binary

self.dec += j \* (16\*\*i)

break

while (binary[0] == '0' and len(binary) > 1):

binary = binary[1:]

self.number = binary

self.system = 2

def add(self, addition):

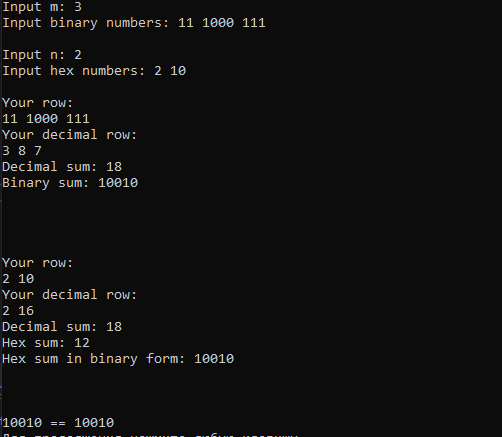
self.changeSystem(10)

addition.changeSystem(10)

self.dec = self.dec + addition.dec

self.changeSystem(16)

**Скріншот роботи програми на Python:**



**Висновок:** під час виконання лабораторної роботи я вивчив механізми успадкування та поліморфізму