

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 10

Виконав студент ІП-11, Друзенко Олександра Юріївна
(шифр, прізвище, ім'я, по батькові)

Перевірів Вітковська Ірина Іванівна
(прізвище, ім'я, по батькові)

Київ 2022

Мета: вивчити механізми створення і використання успадкування та поліморфізму в класах.

Постановка задачі:

10. Створити клас TNumber, який представляє число і містить методи для знаходження суми цифр та першої / останньої цифри числа. На основі цього класу створити класи-нащадки TIntNumber та TRealNumber. Створити m об'єктів цілих чисел та n об'єктів дійсних чисел (дані згенерувати випадковим чином). Знайти суму перших цифр цілих чисел та суму останніх цифр дійсних чисел.

Виконання мовою C++:

1)код:

lib.h:

```
#pragma once
#include "TNumber.h"
#include "TIntNumber.h"
#include "TRealNumber.h"
#include <vector>
#include <iostream>

void create_array(int count, std::vector<TIntNumber>&); //створення масиву з цілими числами
void create_array(int count, std::vector<TRealNumber>&); //створення масиву з раціональними числами
int sum_first(std::vector<TIntNumber>); //знаходження суми перших цифр для масиву цілих чисел
int sum_last(std::vector<TRealNumber>); //знаходження суми останніх цифр для масиву раціональних чисел

template<class T> //шаблон-функція для виводу масиву в консоль
inline void output_array(std::vector<T> arr)
{
    int a = 0;
    std::cout << '\n';
    for (auto num : arr) {
        std::cout <<a+1<<". " << num.get_str_number() << std::endl;
        a++;
    }
}
```

lib.cpp:

```
#include "lib.h"
#include <stdlib.h>
#include <time.h>
#include <stdio.h>

void create_array(int count, std::vector<TIntNumber>& arr)
{
    for (int i = 0; i < count; i++) {
        arr.push_back(TIntNumber(rand() % 1000 + 1));
    }
}
```

```

void create_array(int count, std::vector<TRealNumber>& arr) {
    for (int i = 0; i < count; i++) {
        float num = (rand() % 1000 + 1) + ((rand() % 100)/100000.0);
        arr.push_back(TRealNumber(num));
    }
}

int sum_first(std::vector<TIntNumber> arr) {
    int res = 0;
    for (TIntNumber num : arr) {
        res += num.find_first_digit();
    }
    return res;
}

int sum_last(std::vector<TRealNumber> arr) {
    int res = 0;
    for (TRealNumber num : arr) {
        res += num.find_last_digit();
    }
    return res;
}

```

lab_5.cpp:

```

#include "lib.h"

int main()
{
    std::vector<TRealNumber> array_real;
    std::vector<TIntNumber> array_int;
    int m, n;

    srand(time(NULL));
    std::cout << "int m = "; std::cin >> m; //робота з масивом цілих чисел
    create_array(m, array_int);
    output_array(array_int);
    std::cout << "\nsum of first digit in array = " << sum_first(array_int);

    std::cout << "\n\nreal n = "; std::cin >> n; //робота з масивом раціональних
чисел
    create_array(n, array_real);
    output_array(array_real);
    std::cout << "\nsum of last digit in array = " << sum_last(array_real);
}

```

TNumber.h:

```

#pragma once
#include <string>
class TNumber
{
protected:
    int number;
public:
    TNumber();
    TNumber(int);
    virtual std::string get_str_number();
    int find_sum();
    int find_first_digit();
    virtual int find_last_digit();
};

```

TNumber.cpp:

```
#include "TNumber.h"

TNumber::TNumber()
{
    number = 0;
}

TNumber::TNumber(int num)
{
    number = num;
}

std::string TNumber::get_str_number()
{
    return std::to_string(number);
}

int TNumber::find_sum()
{
    std::string str = std::to_string(number);
    int res = 0;
    for (int i = 0; i < str.length(); i++) {
        res += str[i] - '0';
    }
    return res;
}

int TNumber::find_first_digit()
{
    return std::to_string(number)[0] - '0';
}

int TNumber::find_last_digit()
{
    std::string str = std::to_string(number);
    return str[str.length() - 1] - '0';
}
```

TIntNumber.h:

```
#pragma once
#include "TNumber.h"
class TIntNumber :
    public TNumber
{
public:
    TIntNumber(int num);
};
```

TIntNumber.cpp:

```
#include "TIntNumber.h"

TIntNumber::TIntNumber(int num):TNumber(num)
{
}
```

TRealNumber.h:

```
#pragma once
#include "TNumber.h"
class TRealNumber :
    public TNumber
```

```

{
    int frac;
public:
    TRealNumber(float);
    std::string get_str_number() override;
    int find_last_digit() override;
};

```

TRealNumber.cpp:

```

#include "TRealNumber.h"
#include <string>

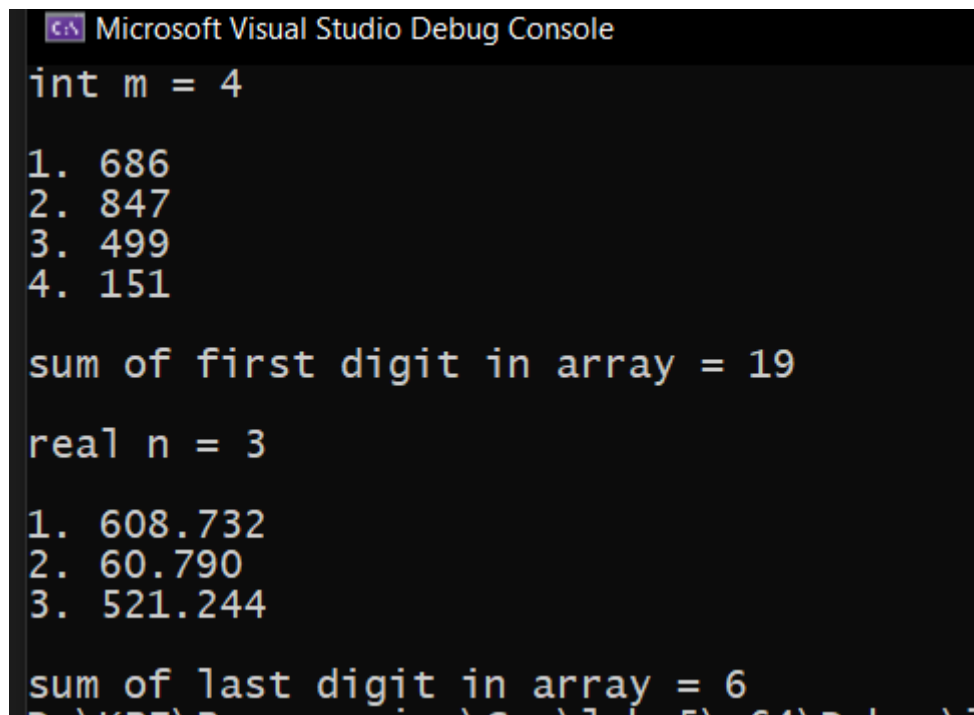
TRealNumber::TRealNumber(float num)
{
    std::string str = std::to_string(num);
    int p = str.find('.');
    number = std::stoi(str.substr(0,p));
    frac = std::stoi(str.substr(p + 1));
}

std::string TRealNumber::get_str_number()
{
    return std::to_string(number) + '.' + std::to_string(frac);
}

int TRealNumber::find_last_digit()
{
    std::string str = std::to_string(frac);
    return str[str.length() - 1] - '0';
}

```

2)Випробування коду на C++:



```

Microsoft Visual Studio Debug Console

int m = 4

1. 686
2. 847
3. 499
4. 151

sum of first digit in array = 19

real n = 3

1. 608.732
2. 60.790
3. 521.244

sum of last digit in array = 6

```

Виконання мовою Python:

1) Код:

main.py:

```
from lib import *

m = int(input('кількість int = '))
int_array = create_int_array(m)
print(int_array)
print('сума перших цифр цілих чисел = ', sum_first(int_array))

n = int(input('кількість real = '))
real_array = create_real_array(n)
print(real_array)
print('сума останніх цифр раціональних чисел = ', sum_last(real_array))
```

lib.py:

```
from random import randint, uniform
from TIntNumber import TIntNumber
from TRealNumber import TRealNumber

def create_int_array(m):
    int_array = []
    for i in range(0, m):
        int_array.append(TIntNumber(randint(1, 1000)))
    return int_array

def create_real_array(n):
    real_array = []
    for i in range(0, n):
        real_array.append(TRealNumber(round(uniform(1.0, 1000.0), 3)))
    return real_array

def sum_first(arr):
    sum = 0
    for x in arr:
        sum += x.find_first_digit()
    return sum

def sum_last(arr):
    sum = 0
    for x in arr:
        sum += x.find_last_digit()
    return sum
```

TNumber.py:

```
class TNumber:
    def __init__(self, num):
        self._number = num

    def __repr__(self):
        return str(self._number)

    def get_number(self):
```

```

        return self._number

    def find_sum(self):
        n = self._number
        sum = 0
        while n > 0:
            digit = n % 10
            sum = sum + digit
            n //= 10
        return sum

    def find_first_digit(self):
        n = str(self._number)
        return int(n[0])

    def find_last_digit(self):
        n = str(self._number)
        return int(n[-1])

```

TIntNumber.py:

```

from TNumber import TNumber

class TIntNumber(TNumber):
    def __init__(self, num):
        super().__init__(num)

```

TRealNumber.py:

```

from TNumber import TNumber

class TRealNumber(TNumber):
    def __init__(self, num):
        super().__init__(int(str(num).partition('.')[0]))
        self.__frac = int(str(num).partition('.')[2])

    def __repr__(self):
        return str(self._number) + '.' + str(self.__frac)

    def get_number(self):
        return float(str(self._number) + '.' + str(self.__frac))

    def find_last_digit(self):
        return int(str(self.__frac)[-1])

```

2)Випробування коду на Python:

```

кількість int = 4
[991, 382, 213, 838]
сума перших цифр цілих чисел = 22
кількість real = 4
[327.904, 722.907, 115.219, 505.383]
сума останніх цифр раціональних чисел = 23

```

Висновок. Отже, на цій лабораторній роботі я вивчила створення та використання успадкування та поліморфізму в об'єктно-орієнтованому програмуванні, надбала навички написання програм роботи з даними методами.