Министерство образования и науки РФ

Севастопольский государственный университет

Институт информационных технологий и управления в технических сетях

Лабораторная работа №2

**ПРОСТОЕ НАСЛЕДОВАНИЕ**. ВИРТУАЛЬНЫЕ ФУНКЦИИ.

Выполнил:

ст.гр.ИСб-22д

Воронин И.Ю.

Проверил:

Тимофеев И.С.

Севастополь

2015

1.Цель работы

Приобретение практических навыков при написании объектно-ориентированных программ с использованием механизма наследования и механизма виртуальных функций. Освоение особенностей отладки объектно-ориентированных программ.

2.Вариант задания

Вариант 10

Создать абстрактный базовый класс Вектор (одномерный массив) с виртуальной функцией Сортировка, которая упорядочивает вектор по возрастанию значения элементов. Создать производные классы: вектор целых чисел (int\*) и два вектора вещественных чисел (float\* и double\*).

3.Текст программы

#pragma once

class Vector{

public:

Vector(void);

virtual void Sort()=0

virtual ~Vector(){}

};

#pragma once

#include "Vector.h"

class VectorI:public Vector{

private:

int\* mas;

int amount;

public:

VectorI(int n);

void Sort(void);

void Add(void);

void Print(void);

~VectorI();

};

#include "VectorI.h"

#include <algorithm>

#include <iostream>

VectorI::VectorI(int n){

amount = n;

mas = new int[amount];

}

void VectorI::Sort(void){

for (int i = amount - 1; i >= 0; i--){

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

if (mas[j] > mas[j + 1])

std::swap(mas[j], mas[j + 1]);

}

}

}

void VectorI::Add(void){

std::cin.clear();

std::cin.ignore();

std::cout << std::endl << "Enter array(" << amount << "): " << std::endl;

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

std::cin >> mas[i];

}

}

void VectorI::Print(void){

std::cout << std::endl << "Sort array: " << std::endl;

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

std::cout << mas[i] << ' ';

}

std::cout << std::endl;

}

VectorI::~VectorI(){

delete[] mas;

}

#pragma once

#include "Vector.h"

class VectorF :Vector{

private:

float\* mas;

int amount;

public:

VectorF(int n);

void Sort(void);

void Add(void);

void Print(void);

~VectorF();

};

#include "VectorF.h"

#include <algorithm>

#include <iostream>

VectorF::VectorF(int n){

amount = n;

mas = new float[amount];

}

void VectorF::Sort(void){

for (int i = amount - 1; i >= 0; i--){

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

if (mas[j] > mas[j + 1])

std::swap(mas[j], mas[j + 1]);

}

} }

void VectorF::Add(void){

std::cin.clear();

std::cout << std::endl << "Enter array("<< amount <<"): " << std::endl;

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

std::cin >> mas[i];

}

}

void VectorF::Print(void){

std::cout << std::endl << "Sort array: " << std::endl;

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

std::cout << mas[i] << ' ';

}

std::cout << std::endl;

}

VectorF::~VectorF(){

delete[] mas;

}

#pragma once

#include "VectorF.h"

class VectorD:Vector

{

private:

double\* mas;

int amount;

public:

VectorD(int n);

void Sort(void);

void Add(void);

void Print(void);

~VectorD();

};

#include "VectorD.h"

#include <algorithm>

#include <iostream>

VectorD::VectorD(int n){

amount = n;

mas = new double[amount];

}

void VectorD::Sort(void){

for (int i = amount - 1; i >= 0; i--){

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

if (mas[j] > mas[j + 1])

std::swap(mas[j], mas[j + 1]);}

}

}

void VectorD::Add(void){

std::cin.clear();

std::cin.ignore();

std::cout << std::endl << "Enter array("<< amount <<"): " << std::endl;

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

std::cin >> mas[i];

}

}

void VectorD::Print(void){

std::cout << std::endl << "Sort array: " << std::endl;

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

std::cout << mas[i] << ' ';

}

std::cout << std::endl;

}

VectorD::~VectorD(){

delete[] mas;

}

Функция Main:

#pragma once

#include <iostream>

#include "VectorI.h"

#include "VectorF.h"

#include "VectorD.h"

int main(){

VectorI \*i = new VectorI(6);

VectorD \*d = new VectorD(6);

VectorF \*f = new VectorF(6);

i->Add(); d->Add(); f->Add();

i->Sort(); d->Sort(); f->Sort();

i->Print(); d->Print(); f->Print();

delete i; delete d; delete f;

return 0;

}

4.Выполнения программы

Был создан класс Vector, который содержал в себе виртуальную функцию сортировки. От данного класса наследовались при класса, которые представляли собой массивы различных типов данных: int, double, float. В каждом классе содержались функции ввода и вывода, а также была описана функция сортировки, наследованная от старшего класса. Родительский класс является абстрактным. В самой программе создавались динамические объекты, которые выполняли последовательно действия: ввод массива, сортировка массива, печать массива.

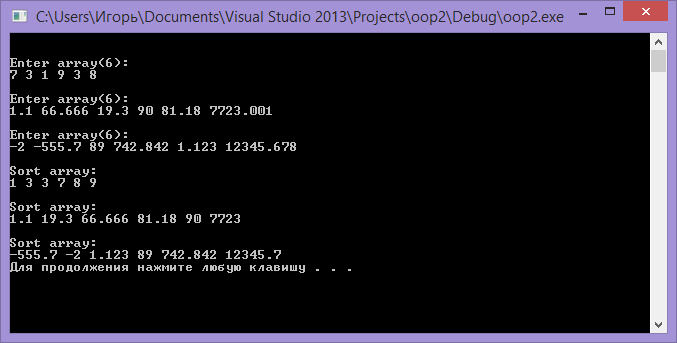


Рисунок 4.1. -Тест программы.

ВЫВОДЫ

В данной лабораторной работе были изучены понятия наследования классов(простое), применение абстрактных классов и виртуальных функций.

Было создано 1 родительский и 3 класса наследника. Данные классы были предназначены для работы с одномерным массивом разных типов данных. Для осуществления проекта использовалась среда Microsoft Visual Studio.