Министерство образования Республики Беларусь

Учреждения образования

“Брестский государственный технический университет”

Кафедра интеллектуально-информационных технологий

ОТЧЁТ

По лабораторной работе №2

Перегрузка функций. Разработка и реализация пользовательских классов

Выполнил:

студент 2 курса

группы ИИ-23

Макаревич Н. Р.

Проверил:

Монтик Н. С.

Брест-2023

ЦЕЛЬ РАБОТЫ

1. Изучение перегрузки функций.

2. Программирование классов.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить правила перегрузки функций.

2. Реализовать задания по перегрузке функций.

3. Реализовать разработанные классы на языке С++ в виде консольных приложений, обеспечивающих использование их объектов.

#include <iostream>

#include <vector>

using namespace std;

class Arif {

private:

int a;

int b;

int c;

public:

Arif() { }

Arif(const Arif& temp) {

a = temp.a;

b = temp.b;

c = temp.c;

}

Arif(int a, int b, int c) {

this->a = a;

this->b = b;

this->c = c;

}

int getA() {

return a;

}

int getB() {

return b;

}

int getC() {

return c;

}

void setA(int x) {

a = x;

}

void setB(int x) {

b = x;

}

void setC(int x) {

c = x;

}

void printAll() {

cout << a << " " << b << " " << c << endl;

}

Arif operator + (const Arif& temp) {

return Arif{

a + temp.a,

b + temp.b,

c + temp.c

};

}

Arif operator = (const Arif& temp) {

a = temp.a;

b = temp.b;

c = temp.c;

return \*this;

}

Arif operator ++ () {

++a;

++b;

++c;

return \*this;

}

void AddUp(int i) {

a += i;

}

void AddUp(int i, int j) {

a += i;

b += j;

}

void AddUp(int i, int j, int k) {

a += i;

b += j;

c += k;

}

};

class Film {

private:

string name;

vector <string> actors;

double length;

public:

Film() { }

Film(string name, vector <string> actors, int length) {

this->name = name;

this->length = length;

this->actors = actors;

}

Film(const Film& temp) {

this->name = temp.name;

this->length = temp.length;

this->actors = temp.actors;

}

void setLength(double temp) {

length = temp;

}

void setName(string temp) {

name = temp;

}

void setActors(vector <string> temp) {

actors = temp;

}

double getLength() {

return length;

}

string getName() {

return name;

}

void showAll() {

cout << name << " " << length << endl;

for (int i = 0; i < actors.size(); i++)

cout << actors[i] << " ";

cout << endl;

}

vector <string> getActors() {

return actors;

}

Film operator = (const Film& temp) {

name = temp.name;

length = temp.length;

actors = temp.actors;

return \*this;

}

Film operator + (const Film& temp) {

length += temp.length;

return \*this;

}

Film operator --() {

--length;

return \*this;

}

void AddActors(vector <string> newActors) {

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

actors.push\_back(newActors[i]);

}

}

void AddActors(string\* newActors, int size) {

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

actors.push\_back(newActors[i]);

}

}

void AddActors(string actor) {

actors.push\_back(actor);

}

};

int main()

{

Arif nums(1,1,1);

nums.AddUp(1);

nums.printAll();

nums.AddUp(1, 2);

nums.printAll();

nums.AddUp(1, 1, 3);

nums.printAll();

cout << endl;

vector <string> temp1 = {"1", "2"};

Film film("Batman", temp1, 1.23);

film.AddActors("3");

film.showAll();

vector <string> temp2 = {"4", "5"};

film.AddActors(temp2);

film.showAll();

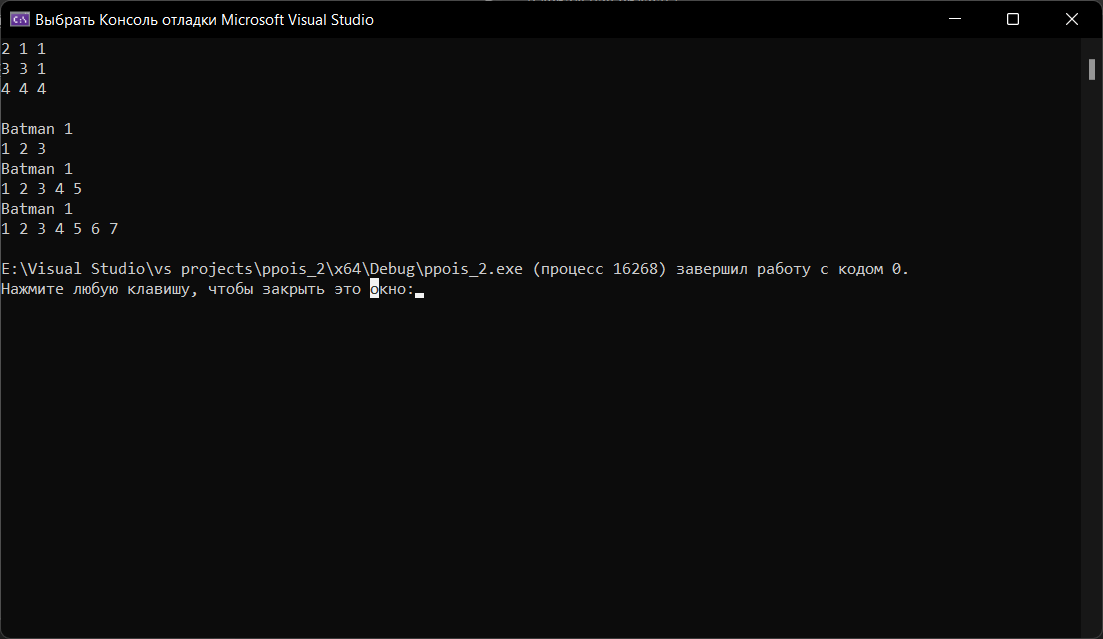
string temp3[2] = {"6", "7"};

film.AddActors(temp3, 2);

film.showAll();

return 0;

}



Вывод:

Изучил перегрузки функций.