Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №2**

По дисциплине «Проектирование программ в ИС»

Тема: «Перегрузка функций. Разработка и реализация пользовательских классов»

**Выполнил:**

Студент 2 курса

Группы ИИ-21

Карагодин Д.Л.

**Проверил:**

Монтик Н.С.

Брест 2022

**Цель работы:** Изучение и решение индивидуальных заданий по разработке перегруженных функций, описание классов.

**Ход работы:**

1. Изучение перегрузки функций.

2. Программирование классов.

**Код программы:**

#include <iostream>

#include <iomanip>

using namespace std;

class Student {

private:

int age;

double score;

char name[64];

public:

Student() { this->age = 0; this->score = 0; strcpy\_s(this->name, "Noname"); }

Student(int age, double score, const char\* name) {

this->age = age;

this->score = score;

strcpy\_s(this->name, name);

}

Student(Student& st2) {

this->age = st2.age;

this->score = st2.score;

strcpy\_s(this->name, st2.name);

}

void To\_Set(int age, float score, const char\* name)

{

this->age = age;

this->score = score;

strcpy\_s(this->name, name);

}

void To\_Show()

{

cout << setw(-20) << "Age of student: " << age << endl;

cout << setw(-20) << "Academic score: " << score << endl;

cout << setw(-20) << "His Name and Surname: "; puts(name);

cout << endl;

}

int GetAge() { return this->age; }

float GetScore() { return this->score; }

char\* GetName() { return this->name; }

friend double operator-(Student& first, Student& second) {

return first.score -= second.score;

}

double operator+(double up) { return this->score += up; }

friend bool operator==(Student& first, Student& second) {

return !strcmp(first.name, second.name);

}

short IsBetter(Student& st1) {

if (st1.score < score) return 1;

else return -1;

}

short IsBetter(Student& st1, Student& st2) {

short promo = 0;

if (st1.score < score) promo++;

else promo--;

if (st2.score < score) promo++;

else promo--;

return promo;

}

short IsBetter(Student& st1, Student& st2, Student& st3) {

short promo = 0;

if (st1.score < score) promo++;

else promo--;

if (st2.score < score) promo++;

else promo--;

if (st3.score < score) promo++;

else promo--;

return promo;

}

};

class Vector {

private:

double x, y, z;

string real\_or\_no;

public:

Vector() {

this->x = 0;

this->y = 0;

this->z = 0;

this->real\_or\_no = "false";

}

Vector(double x, double y, double z, string real\_or\_no = "true") {

this->x = x;

this->y = y;

this->z = z;

this->real\_or\_no = real\_or\_no;

}

Vector(const Vector& v2) {

this->x = v2.x;

this->y = v2.y;

this->z = v2.z;

this->real\_or\_no = v2.real\_or\_no;

}

void Set(double x, double y, double z, string real\_or\_no = "true") {

this->x = x;

this->y = y;

this->z = z;

this->real\_or\_no = real\_or\_no;

}

void Show() {

setprecision(3);

cout << "First coordinate(X): " << x << endl;

cout << "Second coordinate(Y): " << y << endl;

cout << "Third coordinate(Z): " << z << endl;

cout << "He is real?: " << real\_or\_no << endl << endl;

setprecision(6);

}

double GetX() { return this->x; }

double GetY() { return this->y; }

double GetZ() { return this->z; }

string GetRoN() { return this->real\_or\_no; }

friend Vector operator+(Vector& V1, Vector& V2) {

return Vector(V1.x + V2.x, V1.y + V2.y, V1.z + V2.z);

}

Vector operator-(double d) {

return Vector(this->x - d, this->y - d, this->z - d);

}

friend double operator\*(Vector& V1, Vector& V2) {

return (V1.x \* V2.x + V1.y \* V2.y + V1.z \* V2.z);

}

double Skalyar\_myltiply(Vector& V1) {

return (V1.x \* this->x + V1.y \* this->y + V1.z \* this->z);

}

double Skalyar\_myltiply(Vector& V1, Vector& V2) {

return (V1.x \* this->x \* V2.x + V1.y \* this->y \* V2.y + V1.z \* this->z \* V2.z);

}

double Skalyar\_myltiply(Vector& V1,Vector& V2,Vector& V3) {

return (V1.x \* this->x \* V2.x \* V3.x + V1.y \* this->y \* V2.y \* V3.y + V1.z \* this->z \* V2.z \* V3.z);

}

};

class Character {

private:

short id,health;

string name;

public:

Character() {

id = -1;

name = "Andrey Puchinky";

health = 10;

}

Character(short id, string name, short health) {

this->id = id;

this->name = name;

this->health = health;

}

Character(const Character& ch){

this->id = ch.id;

this->name = ch.name;

this->health = ch.health;

}

void Set(short id = -1, string name = "Andrey Puchinky", short health = 10) {

this->id = id;

this->name = name;

this->health = health;

}

void Show() {

cout << "Id: " << id << endl

<< "Name: " << name << endl

<<"Health: " << health << endl;

}

short GetId() { return id; }

string GetName() { return name; }

short GetHealth() { return health; }

friend bool operator==(Character& ch1, Character& ch2) {

if (ch1.id == ch2.id) return true;

else return false;

}

short operator-(short damage) {

this->health -= damage;

return this->health;

}

short operator+(short health) {

this->health += health;

return this->health;

}

short medic(Character& ch1) {

ch1.health += 10;

health -= 5;

return health;

}

short medic(Character& ch1, Character& ch2) {

ch1.health += 10;

ch2.health += 10;

health -= 10;

return health;

}

short medic(Character& ch1, Character& ch2, Character& ch3) {

ch1.health += 10;

ch2.health += 10;

ch3.health += 10;

health -= 15;

return health;

}

};

int main()

{

Student st1(18, 9.5, "Danik Karagodin"), st2(19, 4.5, "Dima Karagodin"), st3(18, 7.2, "Denis Kar1agodin"), st4(17, 6.8, "Difrel Karagodin");

Vector v1(1, 2, 3), v2(9, 55, 4), v3(8, 3, 83), v4(4, 5, 20);

Character ch1(1, "Pudj", 120), ch2(2, "Pucha", 12), ch3(3, "Pupok", 999),ch4(4,"Parfeevec",20);

cout << "Students:\n";

cout<<st1.IsBetter(st2)<<endl;

cout<<st1.IsBetter(st2,st3) << endl;

cout<<st1.IsBetter(st2,st3,st4) << endl;

cout << "Vectors:\n";

cout<<v1.Skalyar\_myltiply(v2) << endl;

cout<<v1.Skalyar\_myltiply(v2,v3) << endl;

cout<<v1.Skalyar\_myltiply(v2,v3,v4) << endl;

cout << "Characters:\n";

cout << ch1.medic(ch2)<<endl;

cout << ch1.medic(ch2,ch3) << endl;

cout << ch1.medic(ch2,ch4) << endl;

return 228;

}

**Результат программы:**



**Вывод:** научился работать с классами. Изучил и применил на практике перегрузки методов.