Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №4**

По дисциплине «Проектирование программ в ИС»

Тема: «Функции-друзья классов. Перегрузка операторов классов»

**Выполнил:**

Студент 2 курса

Группы ИИ-21

Карагодин Д.Л.

**Проверил:**

Монтик Н.С.

Брест 2022

**Цель:**

1. Изучение использования friend-функций для доступа к классам извне.

2. Изучение особенностей перегрузки операторов в пользовательских классах.

**Ход работы:**

#include <string>

#include <iostream>

using std::cout, std::string, std::cin, std::endl;

class Machine {

private:

int price\_;

string name\_;

int year\_;

double speed\_;

int helf\_;

int capacity\_;

public:

Machine() {

this->name\_ = "N/A(BMW)";

this->year\_ = 2022;

this->speed\_ = 0;

this->helf\_ = 0;

this->capacity\_ = 0;

this->price\_ = 0;}

Machine(const Machine& car2) {

this->name\_ = car2.name\_;

this->year\_ = car2.year\_;

this->speed\_ = car2.speed\_;

this->helf\_ = car2.helf\_;

this->capacity\_ = car2.capacity\_;

this->price\_ = car2.price\_;}

void to\_set() {

cout << "Введите название авто: ";

cin >> this->name\_;

cout << "Введите год выпуска: ";

cin >> this->year\_;

cout << "Введите максимальную скорость: ";

cin >> this->speed\_;

cout << "Введите вес авто: ";

cin >> this->helf\_;

cout << "Введите вместимость авто: ";

cin >> this->capacity\_;

cout << "Введите цену авто: ";

cin >> this->price\_;}

void to\_show() {

cout << "Название авто: " << this->name\_ << endl;

cout << "Год выпуска:" << this->year\_ << endl;

cout << "Максимальная скорость: " << this->speed\_ << endl;

cout << "Вес авто: " << this->helf\_ << endl;

cout << "Вместимость: " << this->capacity\_ << endl;

cout << "Цена авто: " << this->price\_ << endl << endl;}

friend void Compare(const Machine& car1, const Machine& car2);

friend void CheckYear(const Machine& car1);

friend void CheckName(const Machine& car1);

void operator-(const Machine& car2) {

cout << "Разница скоростей: " << this->speed\_ - car2.speed\_ << endl;}

void operator+(const Machine& car2) {

cout << "Общий вес: " << this->helf\_ + car2.helf\_ << endl;}};

class Student {

private:

string name;

string email;

int age;

int qualification;

public:

friend void qualificationup(Student&, int);

friend void emailcreated(Student&);

friend void areppers(Student& First, Student& Second);

Student() { name = ""; age = 0; qualification = 0; email = ""; }

Student(Student& Rick) { name = Rick.name; age = Rick.age; qualification = Rick.qualification; email = Rick.email; }

Student(string name, int age, int qualification) { this->name = name; this->age = age; this->qualification = qualification; }

void info() {

cout << "There is info about Student " << name << " :" << endl;

cout << "age: " << age << endl;

cout << "qualification: " << qualification << endl;

cout << "email: " << email << endl << endl;}

string GetName() { return name; }

friend bool operator== (Student& Sample, Student& Sample2) {

if (Sample.qualification == Sample2.qualification) { return true; }

else return false;};

friend int operator+(Student& First, Student& Second) {

return First.qualification + Second.qualification;};

friend void operator-(Student& First, int reduce) {

First.qualification = First.qualification - reduce;}};

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() {

cout.precision(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;

cout.precision(6);}

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);}

friend double Skalyar\_myltiply(Vector& V1, Vector& V2);

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

friend double Skalyar\_myltiply(Vector& V1, Vector& V2, Vector& V3, Vector& V4);};

void qualificationup(Student& Sample, int upby) { Sample.qualification = Sample.qualification + upby; }

void emailcreated(Student& Sample) { Sample.email = Sample.name + "@mail.ru"; }

void areppers(Student& First, Student& Second) {

if (First.age == Second.age)cout << "the Students are peers";

else cout << "the students aren't peers";}

void Compare(const Machine& car1, const Machine& car2) {

short count1 = 0, count2 = 0;

if (car1.year\_ > car2.year\_) count1++;

else count2++;

if (car1.speed\_ > car2.speed\_) count1++;

else count2++;

if (car1.helf\_ < car2.helf\_) count1++;

else count2++;

if (car1.capacity\_ > car2.capacity\_) count1++;

else count2++;

if (car1.price\_ < car2.price\_) count1++;

else count2++;

if (count1 > count2) cout << "Первая машина лучше\n";

else if (count1 != count2) cout << "Вторая машина лучше\n";

else cout << "На равне машины\n";}

void CheckYear(const Machine& car1) {

if (1896 < car1.year\_ && car1.year\_ < 2022) std::cout << "Всё окей";

else std::cout << "Неверный год";}

void CheckName(const Machine& car1) {

if (!car1.name\_.empty()) std::cout << "Всё окей";

else std::cout << "Неверный год";}

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

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

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

return (V1.x \* V3.x \* V2.x + V1.y \* V3.y \* V2.y + V1.z \* V3.z \* V2.z);}

double Skalyar\_myltiply(Vector& V1, Vector& V2, Vector& V3, Vector& V4) {

return (V1.x \* V4.x \* V2.x \* V3.x + V1.y \* V4.y \* V2.y \* V3.y + V1.z \* V4.z \* V2.z \* V3.z);}

int main() {

setlocale(LC\_ALL, "Rus");

Machine car\_machine1;

car\_machine1.to\_set();

cout << "\n";

Machine car\_machine2;

car\_machine2.to\_set();

cout << "\n";

car\_machine1.to\_show();

car\_machine2.to\_show();

Compare(car\_machine1, car\_machine2);

car\_machine1 - car\_machine2; car\_machine1 + car\_machine2;

Vector V1(1, 2, 3), V2(11, 22, 33), V3(13, 21, 32), V4(14, 25, 37);

cout << Skalyar\_myltiply(V1, V2) << endl << Skalyar\_myltiply(V1, V2,V3) << endl << Skalyar\_myltiply(V1, V2,V3,V4) << endl;

Student st1("Pucha", 18, 8), st2("Parfeev", 18, 2), st3("Xucha", 18, 5);

qualificationup(st1, 2);

emailcreated(st2);

areppers(st2, st3);

return 0;}

**Работа программы:**

Введите название авто: a

Введите год выпуска: 1

Введите максимальную скорость: 2

Введите вес авто: 3

Введите вместимость авто: 4

Введите цену авто: 5

Введите название авто: 6

Введите год выпуска: 67

Введите максимальную скорость: 7

Введите вес авто: 8

Введите вместимость авто: 9

Введите цену авто: 0

Название авто: a

Год выпуска:1

Максимальная скорость: 2

Вес авто: 3

Вместимость: 4

Цена авто: 5

Название авто: 6

Год выпуска:67

Максимальная скорость: 7

Вес авто: 8

Вместимость: 9

Цена авто: 0

Вторая машина лучше

Разница скоростей: -5

Общий вес: 11

154

4235

142318

the Students are peers

<<class>>

Vector

-double x

-double y

-double z

-string real\_or\_no

+Vector()

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

+Vector(const Vector& v2)

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

+void Show()

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

+Vector operator-(double d)

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

+friend double Skalyar\_myltiply(Vector& V1, Vector& V2)

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

+friend double Skalyar\_myltiply(Vector& V1, Vector& V2, Vector& V3, Vector& V4)

<<class>>

Machine

-int price\_

-string name\_

-int year\_

-double speed\_

-int helf\_

-int capacity\_

+Machine()

+Machine(const Machine& car2)

+void to\_set()

+void to\_show()

+friend void Compare(const Machine& car1, const Machine& car2);

+friend void CheckYear(const Machine& car1);

+friend void CheckName(const Machine& car1);

+void operator-(const Machine& car2)

+void operator+(const Machine& car2))

-string name

-string email

-int age

-int qualification

+friend void qualificationup(Student&, int)

+friend void emailcreated(Student&)

+friend void areppers(Student& First, Student& Second)

+Student()

+Student(Student& Rick)

+Student(string name, int age, int qualification)

+void info()

+string GetName()

+friend bool operator== (Student& Sample, Student& Sample2)

+friend int operator+(Student& First, Student& Second)

+friend void operator-(Student& First, int reduce)

<<class>>

Student

**Вывод**: Изучил использование friend-функций для доступа к классам извне.