МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ИИТ

Лабораторная работа №3

За 1 семестр

По дисциплине Проектирование программ в интеллектуальных системах

Тема: «Ссылочный тип. Инициализация классов, конструкторы и деструкторы»

Выполнил:

Студент 2 курса

Группы ИИ-19

Летченя С. А.

Проверил:

Монтик Н.С

Брест 2021

Цель:

1. Изучение использования ссылочного типа в пользовательских классах.

2. Программирование классов с использованием конструкторов (по умолчанию, с параметрами, конструктора копирования) и деструкторов.

**Ход работы:**

#include <iostream>

#include <string>

using namespace std;

class Students {

private:

string first\_name\_;

string last\_name\_;

string& name\_1\_ = first\_name\_;

string& name\_2\_ = last\_name\_;

public:

Students() {

this->name\_1\_ = "N\A";

this->name\_2\_ = "N\A";

}

Students(string& name1, string& name2) {

this->name\_1\_ = name1;

this->name\_2\_ = name2;

}

Students(Students& anystudents) {

this->name\_1\_ = anystudents.name\_1\_;

this->name\_2\_ = anystudents.name\_2\_;

}

~Students() {

this->name\_1\_ = "N\A";

this->name\_2\_ = "N\A";

}

void to\_show() {

cout << this->name\_1\_ << " " << this->name\_2\_ << endl;

}

void to\_set(string& name1, string& name2) {

this->name\_1\_ = name1;

this->name\_2\_ = name2;

}

string& to\_get\_firstname() {

return this->name\_1\_;

}

string& to\_get\_lastname() {

return this->name\_2\_;

}

};

class Business {

private:

int income\_;

int cost\_;

int& incomes\_ = income\_;

int& costs\_ = cost\_;

public:

Business() {

this->incomes\_ = 0;

this->costs\_ = 0;

}

Business(int& a, int& b) {

this->incomes\_ = a;

this->costs\_ = b;

}

Business(Business& anybusiness) {

this->incomes\_ = anybusiness.incomes\_;

this->costs\_ = anybusiness.costs\_;

}

~Business() {

this->incomes\_ = 0;

this->costs\_ = 0;

}

void to\_show() {

cout << this->incomes\_ << endl << this->costs\_ << endl;

}

void to\_set(int& a, int& b) {

this->incomes\_ = a;

this->costs\_ = b;

}

int& to\_get\_income() {

return this->incomes\_;

}

int& to\_get\_cost() {

return this->costs\_;

}

int profit() {

return this->incomes\_ - this->costs\_;

}

};

class Human {

private:

string first\_name\_;

int age\_;

string& name\_ = first\_name\_;

int& ages\_ = age\_;

public:

Human() {

this->name\_ = "N\A";

this->ages\_ = 0;

}

Human(string& name1, int& a) {

this->name\_ = name1;

this->ages\_ = a;

}

Human(Human& anyhuman) {

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

this->ages\_ = anyhuman.ages\_;

}

~Human() {

this->name\_ = "N\A";

this->ages\_ = 0;

}

void to\_show() {

cout << this->name\_ << endl << this->ages\_ << endl;

}

void to\_set(string& name, int& a) {

this->name\_ = name;

this->ages\_ = a;

}

string& to\_get\_name() {

return this->name\_;

}

int& to\_get\_age() {

return this->ages\_;

}

};

int main() {

setlocale(LC\_ALL, "rus");

system("color 70");

string a = "Вася", b = "Пупкин", c = "Лёлик", d = "Болик";

Students student1;

Students student2(a, b);

Students student3(student2);

student1.to\_show();

student2.to\_show();

student3.to\_show();

student1.to\_set(c, d);

cout << student1.to\_get\_firstname() << " " << student1.to\_get\_lastname() << endl;

int f = 1000, g = 500;

Business business1;

Business business2(f, g);

Business business3(business2);

business1.to\_show();

business2.to\_show();

business3.to\_show();

cout << business3.profit() << endl;

string name = "Борис"; int age = 99;

Human boris(name, age);

boris.to\_show();

boris.to\_set(c, g);

cout << boris.to\_get\_name() << ' ' << boris.to\_get\_age() << endl;

return 0;

}

<<class>>

Business

<<class>>

Students

-int income\_

-int cost\_

-int& incomes\_

-int& costs\_

+Business()

+Business(int& a, int& b)

+Business(Business &anybusiness)

+~Business()

+void to\_show()

+void to\_set(int& a, int& b)

+void to\_get\_income()

+void to\_get\_cost()

+int profit()

-string first\_name\_

-string last\_name\_

-string& name\_1\_

-string& name\_2\_

+Students()

+Students(string& name1, string& name2)

+Students(Students &anystudents)

+~Students()

+void to\_show()

+void to\_set(string& name1, string& name2)

+void to\_get\_firstname()

+void to\_get\_lastname()

<<class>>

Human

-string first\_name\_

-int age\_

-string& name\_

-int& ages\_

+Human()

+Human(string& name, int& a)

+Human(Human &anyhuman)

+~Human()

+void to\_show()

+void to\_set(string& name, int& a)

+void to\_get\_name()

+void to\_get\_age()

Вывод: Изучил работу с ссылочными типами данных в классе. Инициализацию классов, конструкторов и деструкторов.