Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №3**

По дисциплине «Проектирование программ в ИС»

# Ссылочный тип. Инициализация классов, конструкторы и деструкторы”

**Выполнил:**

Студент 2 курса

Группы ИИ-21

Парфеевец И.А.

**Проверил:**

Монтик Н.С.

Брест 2022

ЦЕЛЬ РАБОТЫ

1. Изучение использования ссылочного типа в пользовательских классах.

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

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить правила использования ссылочного типа в функциях и методах классов.

2. Воспроизвести лекционные примеры, выполнить задания по использованию ссылок.

3. Модифицировать классы индивидуальных заданий, использовав для их инициализации конструкторы всех типов.

Код Программы

#include <iostream>

using namespace std;

class Dandy {

public :

int amount;

int& amountref = amount;

Dandy(int amount, int& amountref) {

this->amount = amount;

this->amountref = amountref;

}

Dandy() {

amount = 0;

}

Dandy(Dandy& dan) {

this->amount = dan.amount;

}

void To\_Show() {

cout << "amount : " << amount<<endl;

cout << "amountref : " << amountref << endl;

}

void To\_Set(int a) {

this->amount = a;

}

int GetAMOUNT() {

return amount;

}

int& showamount(int amount) {

int a = amount;

return a;

}

void reset (Dandy& app) {

app.amount = 0;

}

int& dandy(Dandy& app) {

}

};

//создадим функции для того ,чтобы посмотреть,как меняются значения при передаче через ссылку указатель и значение :

void foo(int a) {

a = 0;

}

void foo1(int& a) {

a = 0;

}

void foo2(int\* a) {

\*a=0 ;

}

class Trend {

public:

int trendtype;

int& trendtyperef = trendtype;

Trend(int trendtype) {

this->trendtype = trendtype;

}

Trend() {

trendtype = 0;

}

Trend(Trend& dan) {

this->trendtype = dan.trendtype;

}

void To\_Show() {

cout << "amount : " << trendtype << endl;

cout << "amountref : " << trendtyperef << endl;

}

void To\_Set(int a) {

this->trendtype = a;

}

int GetTrendtype() {

return trendtype ;

}

void reset(Trend& app) {

app.trendtype = 0;

}

};

class Employee {

public:

int age;

string name;

int workexperience;

int &score = workexperience ;

Employee() { age = 0; name = ""; workexperience = 0; }

Employee(Employee& sample) {

age = sample.age;

name = sample.name;

workexperience = sample.workexperience;

}

Employee(int age , string name, int workexpirience) {

this->age = age;

this->name = name;

this->workexperience = workexpirience;

}

void To\_Show() {

cout << "information about Employee " << name<<endl;

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

cout << "Work experience: " << workexperience<<endl;

cout << "score :" << score<<endl<<endl;

}

int GetExperience() { return workexperience; }

void SetExperience(int a) { this-> workexperience = a; }

void UpPoints(Employee& Promoted, int up) { Promoted.workexperience = Promoted.workexperience + up; }

};

int main() {

int a = 5;

cout << "first values :\n";

Dandy firstamount (a,a);

firstamount.To\_Show();

cout << "now we've changed amount value from 4 to 6 and let's see how our variable amount ref was affected ! " << endl;

firstamount.To\_Set(6);

firstamount.To\_Show();

cout << "let's reset values of firstamount using reset fucnction\n";

firstamount.reset(firstamount);

firstamount.To\_Show(); cout << endl << endl;

//передадим переменню а в функции и посмотрим , поменяется ли она :

foo(a);

cout << "variable a passed as value :" << a<<endl;

foo1(a);

cout << "variable a passed as reference :" << a<<endl;

foo2(&a);

cout << "variable a passed as pointer :" << a<<endl<<endl<<endl;

//|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Employee Andrew(19, "Andre Puchina", 1);

cout << "created object's stats: " << endl;

Andrew.To\_Show();

cout << "Let's icnrease Andrew's experience by 5 :\n";

Andrew.SetExperience(6);

Andrew.To\_Show();

cout << "Let's icnrease Andrew's points by 20 :\n";

Andrew.UpPoints(Andrew, 20);

Andrew.To\_Show();

cout << "as we can see, both and experience and points values are changed !\n";

return 713;

}:

Результат работы программы :

first values :

amount : 5

amountref : 5

now we've changed amount value from 4 to 6 and let's see how our variable amount ref was affected !

amount : 6

amountref : 6

let's reset values of firstamount using reset fucnction

amount : 0

amountref : 0

variable a passed as value :5

variable a passed as reference :0

variable a passed as pointer :0

created object's stats:

information about Employee Andre Puchina

age: 19

Work experience: 1

score :1

Let's icnrease Andrew's experience by 5 :

information about Employee Andre Puchina

age: 19

Work experience: 6

score :6

Let's icnrease Andrew's points by 20 :

information about Employee Andre Puchina

age: 19

Work experience: 26

score :26

as we can see, both and experience and points values are changed !

Вывод:научился работать с сылками в классах.