Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный университет”

Кафедра ИИТ

Лабораторная работа №2

По дисциплине: “ЯП”

Тема: “Наследование и виртуальные фукции”

Выполнил:

Студент 2-го курса

Группы ПО-7

Качан В.В.

Проверила:

Дряпко А.В.

Брест, 2021

**Вариант 5**

**Порядок выполнения работы.**

1. Определить иерархию классов (в соответствии с вариантом).

2. Определить в классе статическую компоненту - указатель на нача-

ло связанного списка объектов и статическую функцию для просмотра

списка.

3. Реализовать классы – организация, страховая компания, судостроительная компания, завод.

4. Написать демонстрационную программу, в которой создаются

объекты различных классов и помещаются в список, после чего список

просматривается.

5. Сделать соответствующие методы не виртуальными и посмотреть,

что будет.

6. Реализовать вариант, когда объект добавляется в список при соз-

дании, т.е. в конструкторе (смотри пункт 6 следующего раздела).

**Определения классов:**

**Organization.h**

#ifndef YAP2\_ORGANIZATION\_H  
#define YAP2\_ORGANIZATION\_H  
#include<iostream>  
#include<string>  
#include<list>  
using namespace std;  
class Organization{  
protected:  
 string name;  
public:  
 Organization(string name);  
 virtual void show() = 0;  
};  
#endif //YAP2\_ORGANIZATION\_H

**Factory.h**

#ifndef YAP2\_FACTORY\_H  
#define YAP2\_FACTORY\_H  
#include "Sudos.h"  
#include "Organization.h"  
class Factory: public Sudos{  
protected:  
 string position;  
public:  
 Factory(string name, int number, string position);  
 void show();  
 void set\_position(string position);  
 string get\_position();  
};  
#endif //YAP2\_FACTORY\_H

**Sudos.h**

#ifndef YAP2\_SUDOS\_H  
#define YAP2\_SUDOS\_H  
#include "Organization.h"  
  
class Sudos: public Organization{  
protected:  
 int number;  
public:  
 Sudos(string name, int number);  
 void show();  
 void set\_number(int number);  
 int get\_number();  
};  
#endif //YAP2\_SUDOS\_H

**Company.h**

#ifndef YAP2\_COMPANY\_H  
#define YAP2\_COMPANY\_H  
#include "Organization.h"  
  
class Company: public Organization{  
protected:  
 int price;  
public:  
 Company(string name, int price);  
 void show();  
 void set\_price(int price);  
 int get\_price();  
};  
#endif //YAP2\_COMPANY\_H

**Определение классов:**

**Organization.cpp**

#include "Organization.h"  
  
Organization::Organization(string name) {  
 this->name = name;  
}

**Factory.cpp**

#include "Factory.h"  
  
Factory::Factory(string name, int number, string position): Sudos(name, number) {  
 this->position = position;  
}  
  
void Factory::show() {  
 cout << "Name of organization: " <<this->name << endl << "Number of ships: " << this->number << endl << "Position: " <<this->position <<endl;  
}  
  
void Factory::set\_position(string position) {  
 this->position = position;  
}  
  
string Factory::get\_position(){  
 return this->position;  
}

**Sudos.cpp**

#include "Sudos.h"  
  
Sudos::Sudos(string name, int number): Organization(name) {  
 this->number = number;  
}  
  
void Sudos::show() {  
 cout << "Organization name: " << this->name << endl << "Number of ships: " << this->number << endl;  
}  
  
void Sudos::set\_number(int number) {  
 this->number = number;  
}  
  
int Sudos::get\_number() {  
 return this->number;  
}

**Company.cpp**

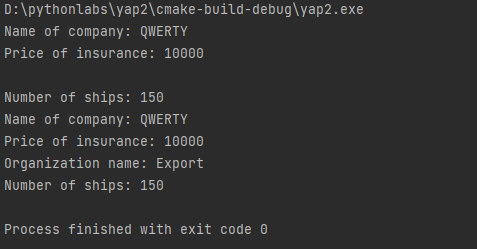
#include "Company.h"  
  
Company::Company(string name, int price): Organization(name) {  
 this->price = price;  
}  
  
void Company::show() {  
 cout<<"Name of organization: "<< this->name << endl<<"Price of insurance: "<<this->price<<endl;  
}  
  
void Company::set\_price(int price) {  
 this->price = price;  
}  
  
int Company::get\_price() {  
 return this->price;  
}

**Демонстрационная программа:**

**main.cpp**

#include "Organization.h"  
#include "Company.h"  
#include "Sudos.h"  
#include "Factory.h"  
  
int main(){  
 list<Organization\*> organization;  
 auto \*firstCompany = new Company("QWERTY", 15000);  
 firstCompany->set\_price(10000);  
 firstCompany->show();  
 organization.push\_back(firstCompany);  
  
 cout<<""<<endl;  
 Sudos \*s1 = new Sudos("Export", 175);  
 s1->set\_number(150);  
 cout<<"Number of ships: "<<s1->get\_number()<<endl;  
 organization.push\_back(s1);  
 std::for\_each(organization.begin(), organization.end(), [](Organization\* organization){  
 organization->show();  
 });  
 return 0;  
}

**Результат выполнения:**



**Вывод:** В ходе лабораторной работы научился наследовать классы, создавать абстрактные классы, а также реализовал виртуальную функцию.