Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №5**

По дисциплине «Проектирование программного обеспечения интеллектуальных систем»

Тема: «Иерархии классов. Наследование»

**Выполнил:**

Студент 2 курса

Группы ИИ-21

Литвинюк Т. В.

**Проверил:**

Монтик Н. С.

Брест 2022

**Цель:** изучить наследование классов.

**Ход работы:**

#include <iostream>

using namespace std;

class Tree{

public:

string nameOfTree;

int age, reliability;

Tree(){

nameOfTree = "";

age = 0;

reliability = 0;

}

Tree(string n, int a, int r){

nameOfTree = n;

age = a;

reliability = r;

}

Tree(Tree &t){

nameOfTree = t.nameOfTree;

age = t.age;

reliability = t.reliability;

}

bool isReliable(){

if(reliability > 50)

return true;

return false;

}

};

class Door : public Tree{

private:

int height, width; string model;

public:

Door(){

model = "";

height = 0;

width = 0;

}

Door(string m, int h, int w, string n, int a, int r){

model = m;

height = h;

width = w;

nameOfTree = n;

age = a;

reliability = r;

}

Door(Door &d){

model = d.model;

height = d.height;

width = d.width;

}

Door(Door &d, Tree &t){

this->model = d.model;

this->height = d.height;

this->width = d.width;

this->nameOfTree = t.nameOfTree;

this->age = t.age;

this->reliability = t.reliability;

}

void TO\_SHOW(){

cout << "Furniture: Door" << endl;

cout << "Model: " << model << endl;

cout << "Name of Tree: " << nameOfTree << endl;

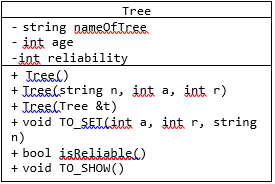
cout << "Age: " << age << endl;

cout << "Reliability: " << reliability << ", is reliable: " << this->isReliable() << endl;

cout << "Height: " << height << endl;

cout << "Width: " << width << endl;

}

}; 

class cupboard : virtual public Tree{

private:

int height, width, depth;

string model;

public:

cupboard(){

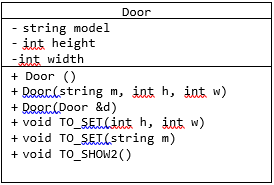
model = "";

height = 0;

width = 0;

depth = 0;

}

 cupboard(string m, int h, int w, int d){

model = m;

height = h;

width = w;

depth = d;

}

cupboard(cupboard &c){

model = c.model;

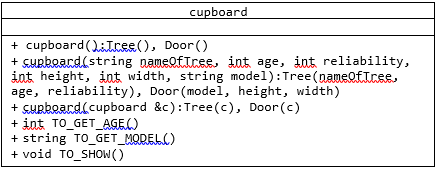
height = c.height;

width = c.width;

depth = c.depth;

}

cupboard(cupboard &c, Tree &t){

 this->model = c.model;

this->height = c.height;

this->width = c.width;

this->depth = c.depth;

this->nameOfTree = t.nameOfTree;

this->age = t.age;

this->reliability = t.reliability;

}

void TO\_SHOW(){

cout << "Furniture: Cupboard" << endl;

cout << "Model: " << model << endl;

cout << "Name of Tree: " << nameOfTree << endl;

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

cout << "Reliability: " << reliability << ", is reliable: " << this->isReliable() << endl;

cout << "Height: " << height << endl;

cout << "Width: " << width << endl;

cout << "Depth: " << depth << endl;

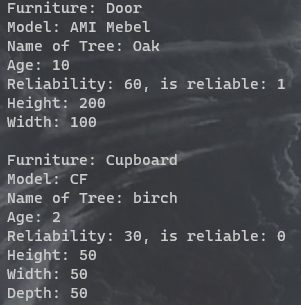
}

};

int main(int argc, char const \*argv[]){

Door d("AMI Mebel", 200, 100, "Oak", 10, 60);

d.TO\_SHOW();

 cout << endl; cupboard c("CF", 50, 50, 50);

Tree t("birch", 2, 30);

cupboard cupboard1(c, t);

cupboard1.TO\_SHOW();

}

**Вывод:** в ходе лабораторной работы я научился использовать ссылочный тип в классах.