МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«Белорусский государственный университет транспорта»

Кафедра «Информационно-управляющие системы и технологии»

Отчет  
по лабораторным работам

по дисциплине «Объектно-ориентированное программирование»

Выполнил Проверил

студент группы ГИ-21 ст. преп. каф. «ИУСиТ»

Зайцев И. С. Голдобина Т. А.

Гомель, 2020

Оглавление

[Лабораторная работа № 5 Тема: «принцип полиморфизма, виртуальные функции, абстрактные классы» 3](#_Toc55555927)

[1.1 Задания 3](#_Toc55555928)

[1.2 Контрольные вопросы 6](#_Toc55555929)

Лабораторная работа № 5  
Тема: «принцип полиморфизма, виртуальные функции, абстрактные классы»

### **Цель**

Изучить реализацию принципа полиморфизма через использование виртуальных функций при наследовании.

## Задания

### Задание 1.1

### Условие

Создать класс Табуретка со свойствами: Высота (h, в см), Качество изделия (низкое, среднее, высокое). Определить два виртуальных метода: «количество древесины», которое требует табуретка, по формуле 4·h+12, если качество низкое, и 5·h+14, если качество среднее или высокое, а также «стоимость», равная d·2, для низкого качества, d·3, для среднего качества, d·4, для высокого качества, где d – количество древесины, которое требует данный объект. Определить также метод «Информация», который возвращает строку, содержащую информацию об объекте: Высоту, качество материала, количество древесины и стоимость. Создать также класс наследник Стул с дополнительным свойством: высота спинки (h2, в см), и переопределить метод «количество древесины», по формуле d+2h2+5, где d – количество древесины, которые требует табуретка с такими же параметрами (Метод «стоимость» не переопределять). В главной программе создать экземпляры классов Табуретка и Стул, и вывести информацию в таком виде: «табуретка» + информация о табуретке и «стул» + информация о стуле.

Программный код

#include <iostream>

using namespace std;

enum quality

{

low, middle, high

};

class Stool

{

public:

float h;

quality q;

virtual float amount\_wood()

{

if(q==low)

{

return 4\*h+12;

}

else

{

return 5\*h+14;

}

}

virtual float price()

{

float d = amount\_wood();

if(q==low)

{

return d\*2;

}

else if(q==middle)

{

return d\*3;

}

else

{

return d\*4;

}

}

string information()

{

string str="Height: " + to\_string(h) + "\nQuality: ";

switch(q)

{

case low: {str += "low";} break;

case middle: {str += "middle";} break;

case high: {str += "high";} break;

}

str += "\nQuantity: " + to\_string(amount\_wood()) + "\nPrice: " + to\_string(price());

return str;

}

};

class Chair : public Stool

{

public:

float spinka;

virtual float amount\_wood()

{

return Stool::amount\_wood()+2\*spinka+5;

}

};

int main()

{Stool first;

cout << "Enter the height of the stool: ";

cin >> first.h;

cout << "Product quality:\n 1.low;\n 2.middle;\n 3.high.\nChoose quality: ";

int n;

cin >> n;

switch(n)

{

case 1: {first.q = low;} break;

case 2: {first.q = middle;} break;

case 3: {first.q = high;} break;

}

cout << "Stool:\n" << first.information() << endl;

Chair second;

cout << "Enter the height of the chair: ";

cin >> second.h;

cout << "Product quality:\n 1.low;\n 2.middle;\n 3.high.\nChoose quality: ";

cin >> n;

switch(n)

{

case 1: {second.q = low;} break;

case 2: {second.q = middle;} break;

case 3: {second.q = high;} break;

}

cout << "Enter the height of spinka: ";

cin >> second.spinka;

cout << "Chair:\n" << second.information() << endl;

return 0;

}

### Результат

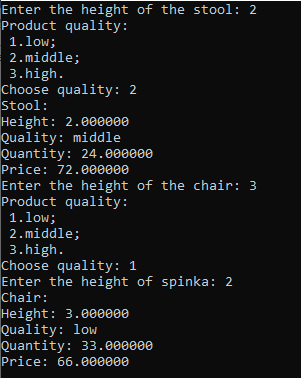


Рисунок 1 – Результат задания 1

## 1.2 Контрольные вопросы

1. Какая функция называется виртуальной?

Виртуальная функция – это функция-член класса, объявленная в базовом классе и переопределенная в производном. Для ее объявления используется ключевое слово virtual.

1. Чем виртуальные функции отличаются от перегружаемых?
2. У перегруженных функций совпадают только имена, а количество и/или типы параметров должны отличаться, чтобы компилятор мог выбрать нужный вариант функции. У виртуальных функций же должны совпадать и имена, и количество параметров, и типы параметров.
3. Виртуальные функции не могут быть статическими членами класса.
4. Какой класс называется абстрактным?

Класс, содержащий хотя бы одну чисто виртуальную функцию, называется абстрактным.

1. В чем состоит различие раннего и позднего связывания?

Различие раннего и позднего связывания состоит непосредственно в том, когда компилятор принимает решение о вызове функции. При отсутствии виртуальных функций решение о вызове принимается компилятором на этапе компиляции программы(раннее связывание), а при наличии виртуальных функций компилятор откладывает решение до фактического запуска программы(позднее связывание).

1. Опишите назначение виртуального деструктора.

Назначение виртуального деструктора заключается в том, чтобы при вызове деструктора для производного класса не вызывался деструктор базового класса и не происходило удаление той части объекта, которая относится к базовому классу.

### Вывод

В результате лабораторной работы научились использовать виртуальные функции при наследовании.