|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**  **Ивангородский гуманитарно-технический институт (филиал)**  **федерального государственного автономного образовательного учреждения высшего образования**  **«Санкт-Петербургский государственный университет аэрокосмического приборостроения»**  **(ИФ ГУАП)** | | | | | | |
|  | | | | | | |
| КАФЕДРА № 2 «Прикладной математики, информатики и информационных таможенных технологий» | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| ОЦЕНКА | | | | | | |
|  | | | | | | |
| ПРЕПОДАВАТЕЛЬ | | | | | | |
|  | | | | | | |
| к.т.н., | | |  |  |  | А.В. Дагаев |
| должность, уч. степень, звание | | |  | подпись, дата |  | инициалы, фамилия |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6 | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| Конструирование классов на основе принципа наследования | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| по дисциплине:  С++ | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| РАБОТУ ВЫПОЛНИЛ | | | | | | |
|  | | | | | | |
| СТУДЕНТ ГР. № | | 922K |  |  |  | Ю. В. Булыгин |
|  | | номер группы |  | подпись, дата |  | инициалы, фамилия |
|  | | | | | | |
|  | | | | | | |
| Студенческий билет № |  | |  | | | |
|  |  | |  | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| Ивангород 2021 | | | | | | |

**Цель работы**

Изучить механизм открытого (public) наследования в C++, познакомиться с понятием   «виртуальная функция»,   освоить технологию конструирования и способы документирования программы,  включающей в себя классы­ наследники,  изучить возможности инструментальных сред разработки по автоматической генерации кода

**Задание**

1. В соответствии с вариантом задания разработать базовый класс. В базовый класс следует включить свойства и методы,   общие для заданных классов ­наследников.   Базовый класс должен включать в себя не менее двух свойств и двух методов,   один из которых   – виртуальная функция.

2. Разработать классы,   производные от базового класса   (наследники).   Классы-­наследники должны наследовать от базового класса хотя бы одно свойство, а также должны иметь хотя бы одно собственное свойство.

В классы­ наследники должны быть включены следующие методы:

a. Метод, наследуемый от базового класса без переопределения.

b. Виртуальная функция базового класса, переопределённая в производном классе.

c. Собственные методы производного класса.

В состав производного класса должен быть включён хотя бы один метод, изменяющий какое ­либо свойство класса.

3. Разработать программу, выполняющую следующие действия:

a. Создание нескольких объектов на основе классов–наследников.

b. Объединение объектов в массив (массив указателей на базовый класс).

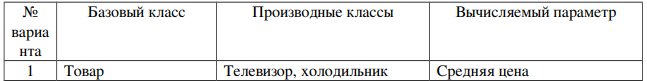
c. Отображение значений свойств объектов на экране в цикле.

d. Изменение свойств объектов по номеру элемента массива.

e. Вычисление заданного параметра.

f. Выход из программы.

4. Действия над объектами   (просмотр,   изменение,   вычисление параметра)   должны быть доступны через меню;   последовательность     выполнения действий   – произвольная, в цикле.



**Среда разработки**

Microsoft Visual Studio 2019

**Описание программы**

Описание полей и методов базового класса:

class tovar

protected:

float cena;

int kod;

public:

virtual void info()

float get\_price()

void set\_kod(int \_kod)

void set\_cena\_tov(float \_cena)

Описание полей и методов дочерних класса:

class televizor : public tovar

string name\_tv;

float cena;

public:

virtual void set\_nametv(string \_name)

void set\_cenatv(float \_cena)

televizor(string \_name, float \_cena)

virtual void info()

virtual float get\_price()

class holodilnik : public Tovar

string name\_hol;

float cena;

public:

holodilnik(string \_name\_hol, float \_cena)

void set\_namehol(string \_name)

void set\_cenahol(float \_cena)

virtual void info()

virtual float get\_price()

**Логика работа программы**

В замкнутом цикле вызывается меню с выбором действий: 1- просмотр товаров; 2 – изменение полей товаров; 3 – вычисление средней цены; 4- выход из программы. Выбрав необходимый из меню пункт посредством ввода значения, осуществляются заданные пользователем действия и вывод результатов на экран.

**Результат работы программы**

На приведённых ниже рисунках представлены примеры работы программы.

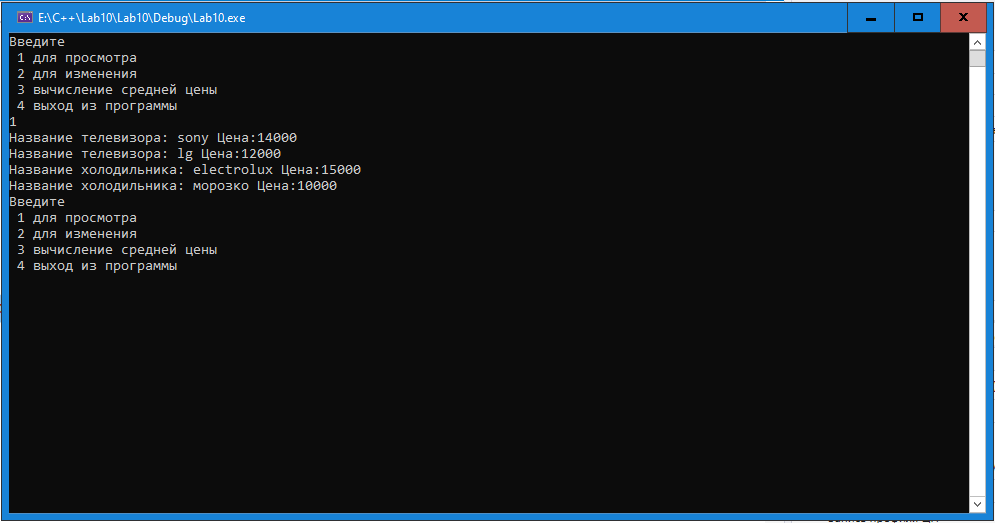


Рисунок 1 –просмотр данных о товарах

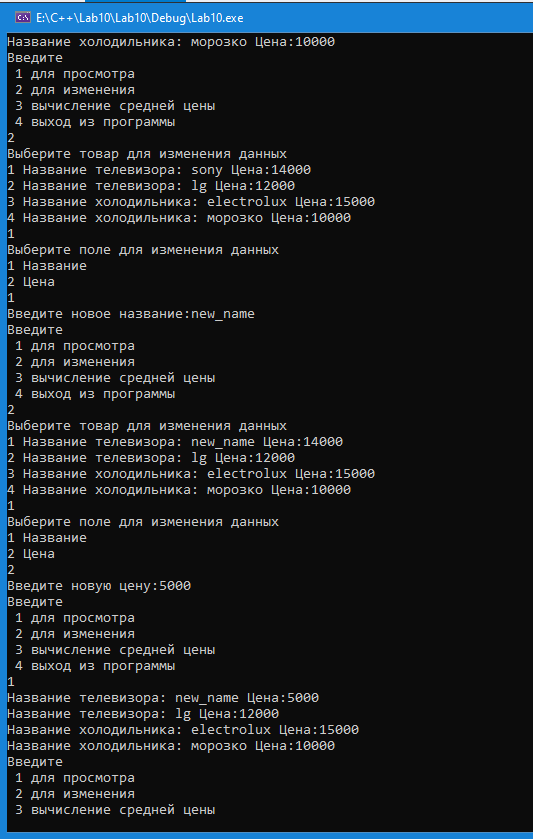


Рисунок 2 – изменение значений полей товара

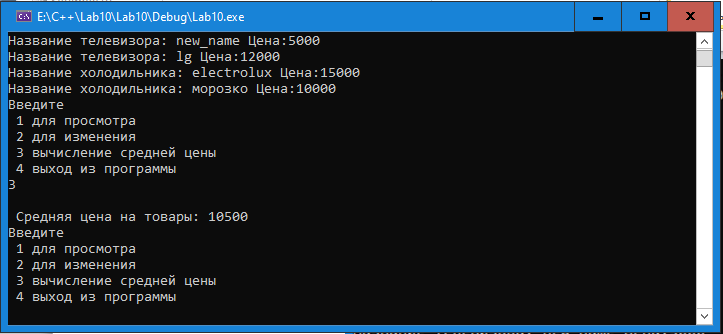


Рисунок 3 – вычисление средней цены

**Выводы**

В результате проделанной работы были получены следующие результаты:

В соответствии с вариантом задания был разработан базовый класс. В базовый класс были включены свойства и методы,   общие для заданных классов ­наследников.   Базовый класс включал в себя не менее двух свойств и двух методов,   один из которых   – виртуальная функция.  Были разработаны классы,   производные от базового класса   (наследники).   Классы-наследники должны наследуют от базового класса одно свойство, а также имеют одно собственное свойство.

В классы­ наследники были включены следующие методы:

a. Метод, наследуемый от базового класса без переопределения.

b. Виртуальная функция базового класса, переопределённая в производном классе.

c. Собственные методы производного класса.

В состав производных классов были включены методы, изменяющие свойства класса . Разработана программа, выполняющая следующие действия:

a. Создание нескольких объектов на основе классов–наследников.

b. Объединение объектов в массив (массив указателей на базовый класс).

c. Отображение значений свойств объектов на экране в цикле.

d. Изменение свойств объектов по номеру элемента массива.

e. Вычисление заданного параметра.

f. Выход из программы.

Действия над объектами   (просмотр,   изменение,   вычисление параметр)   доступны через меню.

**Код программы**

#include <iostream>

#include <string>

#include <windows.h>

using namespace std;

class televizor;

class tovar

{

protected:

float cena;

int kod;

public:

virtual void info() // Виртуальная функция базового класса, переопределённая в производном классе

{

cout << "Koд товара: " << kod << " Цена:" << cena << endl;

}

virtual float get\_price()

{

return cena;

}

void set\_kod(int \_kod)

{

kod = \_kod;

}

void set\_cena\_tov(float \_cena)

{

cena = \_cena;

}

};

class televizor : public tovar

{

string name\_tv;

float cena;

public:

virtual void set\_nametv(string \_name)

{

name\_tv = \_name;

}

void set\_cenatv(float \_cena)

{

cena = \_cena;

}

televizor(string \_name, float \_cena)

{

name\_tv = \_name;

cena = \_cena;

}

virtual void info() //Виртуальная функция базового класса, переопределённая в производном классе

{

cout << "Название телевизора: " << name\_tv << " Цена:" << cena << endl;

}

virtual float get\_price()

{

return cena;

}

};

class holodilnik : public tovar

{

string name\_hol;

float cena;

public:

holodilnik(string \_name\_hol, float \_cena)

{

name\_hol = \_name\_hol;

cena = \_cena;

}

void set\_namehol(string \_name)

{

name\_hol = \_name;

}

void set\_cenahol(float \_cena)

{

cena = \_cena;

}

virtual void info() //

{

cout << "Название холодильника: " << name\_hol << " Цена:" << cena << endl;

}

virtual float get\_price()

{

return cena;

}

};

int main()

{

setlocale(LC\_ALL, "Russian");

televizor tel1("sony", 14000);

televizor tel2("lg", 12000);

holodilnik hol1("electrolux", 15000);

holodilnik hol2("морозко", 10000);

tovar\* pt[4];

//holodilnik\* ph[2];

int m = 0, i, sr = 0;

bool x = true;

pt[0] = &tel1;

pt[1] = &tel2;

pt[2] = &hol1;

pt[3] = &hol2;

while (x)

{

cout << "Введите \n 1 для просмотра \n 2 для изменения \n 3 вычисление средней цены\n 4 выход из программы \n";

cin >> m;

if (m == 1)

{

for (i = 0; i < 4; i++)

{

pt[i]->info();

}

}

if (m == 2)

{

int x1,x2;

float cena;

string nazv;

cout << "Выберите товар для изменения данных \n";

for (i = 0; i < 4; i++)

{

cout << i + 1 << " ";

pt[i]->info();

}

cin >> x1;

cout << "Выберите поле для изменения данных \n";

cout << "1 Название \n";

cout << "2 Цена \n";

cin >> x2;

switch (x2)

{

case 1:

{

cout << "Введите новое название:";

cin >> nazv;

if (x1 == 1) tel1.set\_nametv(nazv);

if (x1 == 2) tel2.set\_nametv(nazv);

if (x1 == 3) hol1.set\_namehol(nazv);

if (x1 == 4) hol2.set\_namehol(nazv);

break;

}

case 2:

{

cout << "Введите новую цену:";

cin >> cena;

if (x1 == 1) tel1.set\_cenatv(cena);

if (x1 == 2) tel2.set\_cenatv(cena);

if (x1 == 3) hol1.set\_cenahol(cena);

if (x1 == 4) hol2.set\_cenahol(cena);

break;

}

}

}

if (m == 3)

{

float sum=0;

for (i = 0; i < 4; i++)

sum += (\*(pt + i))->get\_price();

cout <<"\n Средняя цена на товары: "<< sum / 4<<"\n";

}

if (m == 4)

{

x = false;

}

}

}