|  |  |
| --- | --- |
| Image result for РТУ герб | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждениевысшего профессионального образования"Российский технологический университет"МИРЭА | |
| Факультет информационных технологий (ИТ) | |
| Кафедра практической и прикладной информатики | |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №6** | |
| **по дисциплине** | |
| **«**Объектно-ориентированное программирование**»** | |
|  | |
| Выполнил студент группы ИМБО-01-18 | Пресняков Р.О. |
| Принял ассистент кафедры | Строганкова Н.В. |

|  |  |  |
| --- | --- | --- |
| Лабораторные работы выполнены | «23» февраля 2019 г. |  |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_201\_\_ г. |  |

Москва 2019

## **Практическая работа №6**

**Решение задач на наследование на языке С++**

**Цель практической работы**

Целью данной практической работы является приобретение практических навыков использования принципа ООП – наследования для разработки программ на языке программирования C++.

**Задачи**

Написать иерархию классов, описывающих имущество налогоплательщиков. Она должна состоять из абстрактного базового класса Property и производных от него классов Appartment, Car и CountryHouse. Базовый класс должен иметь поле worth (стоимость), конструктор с одним параметром, заполняющий это поле, и чисто виртуальный метод рассчета налога, переопределенный в каждом из производных классов. Налог на квартиру вычисляется как 1/1000 ее стоимости, на машину – 1/200, на дачу – 1/500. Также, каждый производный класс должен иметь конструктор с одним параметром, передающий свой параметр конструктору базового класса. В функции main завести массив из 7 указателей на Property и заполнить его указателями на динамические объекты производных классов (первые 3 – Appartment, следующие 2 – Car и последние 2 – CountryHouse). Вывести на экран величину налога для всех 7 объектов. Не забудь также уничтожить динамические объекты перед завершением программы.

**Ход работы**

Программа

#include <iostream>  
using namespace std;  
class property {  
protected:  
 int worth;  
public:  
 property(int worth){  
 this->worth = worth;  
 }  
 virtual double nalog() = 0;  
};  
class car: public property{  
public:  
 double nalog(){  
 return worth/200;  
 }  
 car(int worth): property(worth){  
 nalog();  
 }  
};  
class appartment: public property{  
public:  
 double nalog(){  
 return worth/1000;  
 }  
 appartment(int worth):property(worth){  
 nalog();  
 }  
};  
class countryHouse: public property{  
public:  
 double nalog(){  
 return worth/500;  
 }  
 countryHouse(int worth):property(worth){  
 nalog();  
 }  
};  
int main() {  
property \*a[7];  
double sum;  
appartment q = appartment(125000),w = appartment(123456), e = appartment(987654);  
car r = car(111111), t = car(222222);  
countryHouse y = countryHouse(333333), u = countryHouse(444444);  
  
a[0] = &q;  
a[1] = &w;  
a[2] = &e;  
a[3] = &r;  
a[4] = &t;  
a[5] = &y;  
a[6] = &u;  
  
for (int i=0; i<=6; i++){  
 cout <<"item " << i+1 <<" - " << a[i]->nalog() << ' '<< endl;  
 sum += a[i]->nalog();  
}  
cout << endl;  
cout << sum;  
  
 return 0;  
}

**Вывод**

Все поставленные задачи были выполнены. В ходе выполнения работы был изучен механизм наследования классов в C++. Также для выполнения работы были использованы абстрактные классы и чисто виртуальные методы базовых абстрактных классов. Результаты работы программ представлены на рисунке.

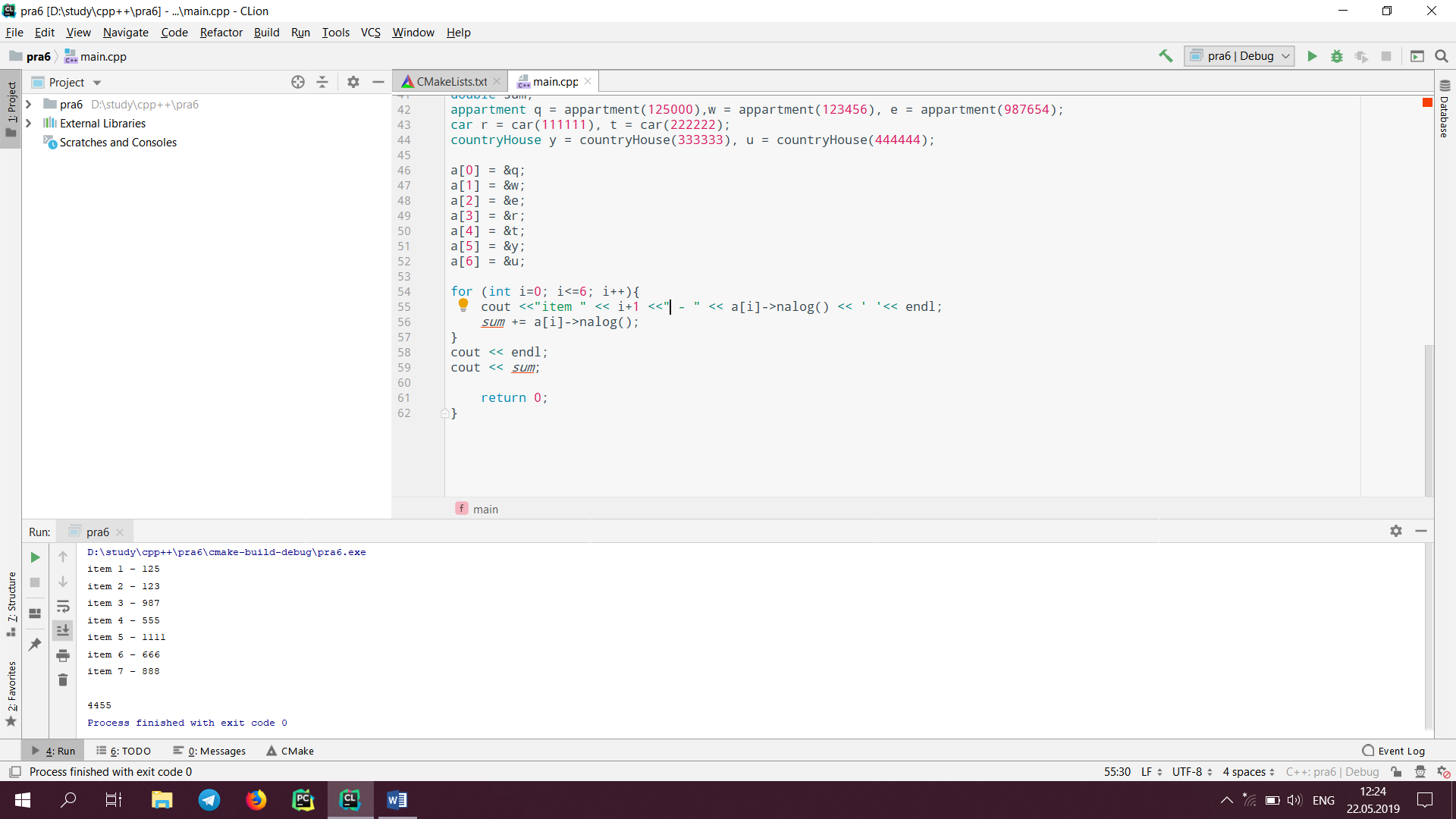


Рис. 1 Результат работы программы