МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cт.преподаватель |  |  |  | Е.О.Шумова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 4 |
| «Наследование классов, базовый класс, производный класс» |
| по курсу: Объектно-ориентированное программирование |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134К |  |  |  | Опарин С.Н. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

1. **Задание**

**Вариант №9:**

В работе необходимо реализовать базовый класс заданной структуры, на основе него создать производные классы. В нём предусмотреть конструктор для установки начальных значений полей. Создать объекты производных классов. Продемонстрировать работу всех методов, реализуемых в классах.

1. **Листинг программы, реализующей алгоритм:**

#include <iostream>

using namespace std;

class dwelling{

protected:

string name;

double area, cost\_per\_day;

int count\_of\_visitors;

public:

dwelling(string n, double a, double c, int cou): name(n), area(a), cost\_per\_day(c), count\_of\_visitors(cou){}

~dwelling(){}

void output();

void cost();

};

void dwelling::output(){

cout << "Название строения - " << name << endl;

cout << "Общая площадь = " << area << endl;

cout << "Цена в сутки = " << cost\_per\_day << endl;

cout << "Количество жильцов - " << count\_of\_visitors << endl;

}

void dwelling::cost(){

cout << "Цена за сутки = " << count\_of\_visitors \* cost\_per\_day << endl;

}

class house : public dwelling{

private:

string material, rent\_or\_private;

double rent\_cost;

public:

house() : dwelling("Дом", 1000, 500, 4), material(""), rent\_or\_private(""), rent\_cost(0){}

house(string name\_of\_building, double ar, double co, int c, string mat, string r\_or\_p, double r\_cost):

dwelling(name\_of\_building, ar, co, c), material(mat), rent\_or\_private(r\_or\_p), rent\_cost(r\_cost){};

void output\_2();//выводит материал дома, является ли он арендным или частным, цену на аренду

void count\_cost();

};

void house::count\_cost(){

cout << "Цена за сутки = " << rent\_cost << endl;

}

void house :: output\_2(){//выводит материал дома, является ли он арендным или частным, цену на аренду

cout << "Материал дома - " << material << endl;

cout << "Категория - " << rent\_or\_private << endl;

cout << "Цена аренды = " << rent\_cost << endl;

}

class cottage: public dwelling{

private:

double area\_of\_cottage;

int count\_of\_floors;

public:

cottage(): dwelling("Коттедж", 1000, 500, 4), area\_of\_cottage(0), count\_of\_floors(0){}

cottage(string name\_of\_building, double ar, double co, int c, double a\_of\_c, int c\_of\_f):

dwelling(name\_of\_building, ar, co, c), area\_of\_cottage(a\_of\_c), count\_of\_floors(c\_of\_f){};

void output\_3();//выводит площадь земельного участка, число этажей

void count\_cost();

};

void cottage::output\_3(){

cout << "Площадь земельного участка = " << area\_of\_cottage << endl;

cout << "Количество этажей = " << count\_of\_floors << endl;

}

void cottage::count\_cost(){

cout << "Цена за сутки = " << count\_of\_floors \* cost\_per\_day << endl;

}

class hostel : public dwelling{

private:

int count\_of\_residents;

public:

hostel():dwelling("Хостел", 1000, 500, 4), count\_of\_residents(0){}

hostel(string name\_of\_building, double ar, double co, int c, int c\_of\_r):

dwelling(name\_of\_building, ar, co, c), count\_of\_residents(c\_of\_r){};

void output\_4();//выводит количество гостей

void count\_cost();

};

void hostel::output\_4(){

cout << "Количество гостей = " << count\_of\_residents << endl;

}

void hostel::count\_cost(){

cout << "Цена за сутки = " << count\_of\_residents \* cost\_per\_day << endl;

}

int main() {

house hou("Дом", 350, 0, 6, "Береза", "rent", 500);

cottage cot("Коттедж", 1000, 750, 10, 1500, 5);

hostel hostl("Хостел", 10, 250, 1, 1);

cout << "Информация о доме: " << endl;

//hou.set\_dwelling\_house();

hou.output();

hou.output\_2();

hou.count\_cost();

cout << endl;

cout << endl;

cout << "Информация о коттедже: " << endl;

//cot.set\_dwelling\_cottage();

cot.output();

cot.output\_3();

cot.count\_cost();

cout << endl;

cout << endl;

cout << "Информация о хостеле:" << endl;

//hostl.set\_dwelling\_hostel();

hostl.output();

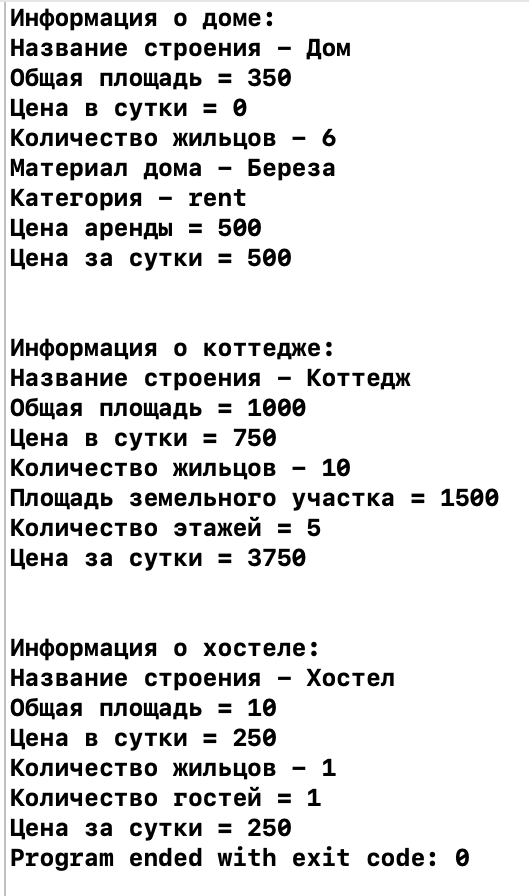
hostl.output\_4();

hostl.count\_cost();

return 0;

}

Скриншот с результатами:



1. **Вывод**

В ходе лабораторной работы я изучил механизм создания нового класса на основе уже существующего, варианты доступа к элементам базового класса из производного.