



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

О Т Ч Е Т (В А Р И А Н Т 1 2)

по лабораторной работе № 1

Название: **Наследование**

Дисциплина: Объектно-ориентированное программирование

Студент

ИУ6-23Б

(Группа)

10.05.2023

(Подпись, дата)

В.К. Залыгин

(И.О. Фамилия)

Преподаватель

А.М. Минитаева

(Подпись, дата)

(И.О. Фамилия)

Москва, 2023

Цель работы

Изучить механизм наследования объектной модели C++.

Задание

Разработать и реализовать иерархию классов для описанных объектов предметной области, используя механизмы наследования. Протестировать все методы каждого класса. Все поля классов должны быть скрытыми (private) или защищенными (protected). Методы не должны содержать операций ввода/вывода, за исключением процедуры, единственной задачей которой является вывод информации об объекте на экран.

Объект – лестница. Поля: материал, длина лестницы, ширина ступеней, количество ступеней. Методы: процедура инициализации объекта, процедура вывода информации об объекте на экран и функция определения высоты одной ступени.

Объект – складная лестница (стремянка). Поля: материал, длина лестницы, ширина ступеней, количество ступеней и угол, на который раскладывается стремянка. Методы: процедура инициализации объекта, процедура вывода информации об объекте на экран и функция вычисления площади, которую займет на полу стоящая стремянка в разложенном виде.

В отчете привести диаграмму разработанных классов и объектную декомпозицию.

Проект программы

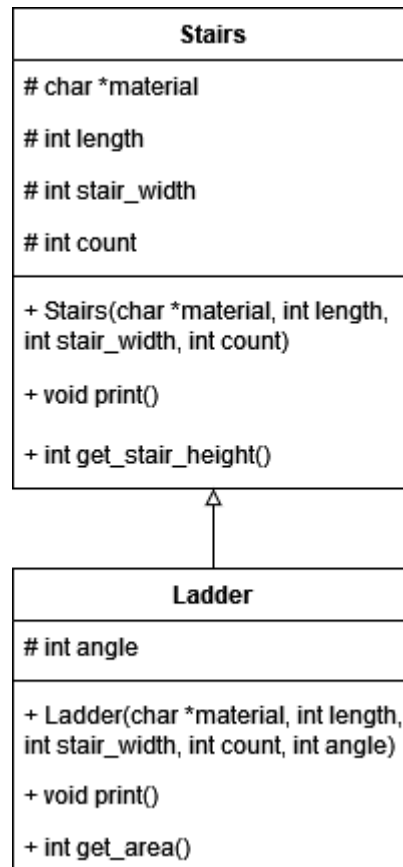


Рисунок 1 - диаграмма классов

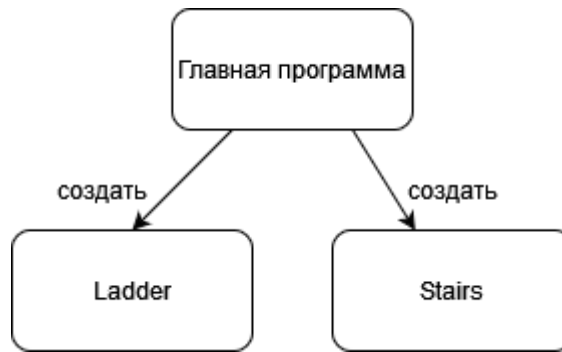


Рисунок 2 - объектная декомпозиция

Текст программы

```

1  #include <cstring>
2  #include <iostream>
3
4  using namespace std;
5
6  You, 3 days ago | 1 author (You)
7  class Stairs {
8  protected:
9      char *material;
10     int length;
11     int stair_width;
12     int count;
13 public:
14     Stairs(char *material, int length, int stair_width, int count)
15         : material(material), length(length), stair_width(stair_width),
16         count(count){}
17
18     void print() {
19         cout << "Stairs; material: " << material << "; length: " << length
20         << "; stair_width: " << stair_width << "; count: " << count << ";";
21     }
22
23     int get_stair_height() { return length / count; }
24 };
25
26 You, 3 days ago | 1 author (You)
27 class Ladder : public Stairs {
28 protected:
29     int angle;
30 public:
31     Ladder(char *material, int length, int stair_width, int count, int angle)
32         : Stairs(material, length, stair_width, count), angle(angle) {}
33
34     void print() {
35         Stairs::print();
36         cout << " angle: " << angle << ";";
37     }
38
39     int get_area() {
40         return length * stair_width;
41     }
42 };
  
```

Рисунок 3 - код программы

```

44 int main() {
45     Ladder l("wood", 300, 50, 5, 70);
46     l.print();
47     cout << endl;
48     cout << "area is " << l.get_area() << ". stair height is " << l.get_stair_height() << endl;
49     return 0;
50 }

```

Рисунок 4 - код программы

Тестовые данные

```

$ g++ main.cpp
$ ./main
Stairs; material: wood; length: 300; stair_width: 50; count: 5; angle: 70;
area is 15000. stair height is 60

```

Рисунок 5 - пример работы

Вывод

Был изучен механизм наследования в объектной модели C++.