

# Отчёт

к экзаменационному проекту № 3

**Мальцевой Софьи Алексеевны,**  
студентки 3 курса группы АБ335  
направления подготовки  
**09.03.02 «Информационные  
системы и технологии»**

Сыктывкар 2023

# Классы Prism, Box, Cube

## Заголовочный файл иерархии классов Prism.h

### Класс Prism

В классе есть одно поле: высота призмы `height`. Это поле типа `double`, как требуется в условии задачи.

Из условия: призма прямоугольная, но форма основания нам не известна, то есть в основании может быть любой многоугольник. Но прямоугольная призма — это призма, у которой все углы прямые? Тогда в основании может быть только прямоугольник. Но зачем тогда было бы делать класс абстрактным и не иметь возможности создавать объекты этого класса? Первая реализация этого класса включала в себя ещё два поля — длину и ширину основания, и эта реализация в виде файлов `Prism.h.old` и `Prism.cpp.old` есть в архиве с исходным кодом (и всё же это старая реализация, и она не вяжется с тем фактом, что класс `Prism` абстрактный. Но пусть старая реализация останется, даже если она ошибочна). Оба поля типа `double`, приватные и доступные только внутри класса. Поэтому в классах `Box` и `Cube` не было собственных полей, а создание и инициализация объектов происходило посредством конструктора родительского класса.

В абстрактном классе `Prism` есть публичный конструктор, публичный виртуальный деструктор, публичный чисто виртуальный метод вычисления площади основания призмы `Square()` (раз класс абстрактный, у него нет своей реализации этого метода, и значение адреса этой функции в таблице виртуальных методов будет нулевым — используем «`= 0`»), обычный публичный метод вычисления объема призмы `Volume()`.

Все методы константные, кроме конструктора и деструктора, так как, во-первых, ни один из них не меняет поле класса, а во-вторых, так требуется в условии задачи. К тому же, все эти методы имеют тип `double`. По умолчанию конструктор принимает значение высоты, равное `1.0`.

### Класс Box

В классе `Box` есть одно поле — длина основания `width`. Оно необходимо для вычисления площади основания (в условии сказано, что основание прямоугольного параллелепипеда — квадрат). Поле приватное, имеет тип числа с плавающей точкой `double`.

В классе публичный конструктор, принимающий высоту призмы и длину основания, который задает высоту через конструктор родительского класса и длину основания своему собственному полю. Также публичный виртуальный деструктор и публичная виртуальная константная функция `Square()`.

### Класс Cube

В классе `Cube` нет собственных полей, но есть публичный конструктор и виртуальный публичный деструктор. В конструкторе передается высота в конструктор

родительского класса `Box` как высота и длина основания прямоугольного параллелепипеда. Учитывая, что основание куба тоже квадрат, класс `Cube` наследует метод `Square()` у родительского класса `Box`.

### Файл `Prism.cpp`

В начале файла директивой `#include` подключаем файл `Prism.h`. Другие библиотеки в реализации класса не требуются.

В файле описываем реализацию метода `Volume()` класса `Prism` (область видимости `Prism::`), `Square()` класса `Box` (область видимости `Box::`).

В методе `Volume()` возвращаем высоту `height`, умноженную на вычисленную площадь основания `Square()` (которая вычислится уже у потомков класса).

В методе `Square()` возвращаем площадь основания как длину основания `width`, возведенную в квадрат.

### Файл `main.cpp`

Подключаем директивой `#include` заголовочный файл `Prism.h` и стандартную библиотеку `<cstdio>` — библиотеку Си для использования операции вывода `printf()`.

В файл помещаем функцию `main()`, описанную в условии задачи (только вместо ошибочной директивы `#lf` прописана директива `#f` у функции `printf()`).

При реализации иерархии классов были учтены условия. Во-первых, все поля классов являются приватными. Во-вторых, дружественные функции не используются. В-третьих, только конструкторы инициализируют объекты. В-четвертых, все методы, кроме конструкторов и деструкторов, константные. В-пятых, нет дублирования полей базового класса в классах-потомках.