

Отчёт

к экзаменационному проекту № 2

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

Сыктывкар 2023

Классы Body, Cube, Tetrahedron

Заголовочный файл иерархии классов Body.h

Класс Body

В классе есть одно поле: плотность абстрактного физического тела `density`. Это поле типа `double`, как требуется в условии задачи.

В абстрактном классе Body есть публичный конструктор, публичный виртуальный деструктор, публичный чисто виртуальный метод вычисления объема физического тела `Volume()` (класс абстрактный, у него нет своей реализации этого метода, и значение адреса этой функции в таблице виртуальных методов будет нулевым — « = 0 »), обычный публичный метод вычисления массы физического тела `Mass()`.

Все методы константные, кроме конструктора и деструктора, ни один из них не меняет поле класса, и так требуется в условии задачи. У всех методов тип `double`. По умолчанию конструктор задает значение плотности тела, равное 1.0.

Класс Cube

В классе Cube есть только одно поле — длина ребра куба `length`. Она нужна для вычисления объема куба. Поле приватное, имеет тип `double`.

В классе есть публичный конструктор, принимающий плотность куба и длину ребра, который передает значение плотности конструктору родительского класса, а длину ребра приравнивает собственному полю. Также есть публичный виртуальный деструктор и публичная виртуальная константная функция `Volume()`.

Класс Tetrahedron

В классе Tetrahedron есть одно поле — длина ребра тетраэдра `length`, из методов есть публичный конструктор, виртуальный публичный деструктор и публичная виртуальная константная функция `Volume()`.

В конструкторе принимается плотность тетраэдра и длина его ребра. Плотность передается в конструктор родительского класса, длина ребра тетраэдра приравнивается своему полю `length`.

Файл Body.cpp

В начале файла директивой `#include` подключаем файл `Body.h`. Для извлечения квадратного корня функцией `sqrt()` требуется подключение библиотеки `См <cmath>`, предназначенной для проведения математических операций.

В файле описываем реализацию метода `Mass()` класса Body, (область видимости `Body::`) `Volume()` класса Cube (область видимости `Cube::`) и класса Tetrahedron (область видимости `Tetrahedron::`).

В методе `Mass()` возвращаем массу тела в виде плотности `density`, умноженной на вычисленный объем тела `Volume()` (который вычислится у потомков класса).

В методе `Volume()` класса `Cube` вычисляется объем куба как длина ребра `length`, возведенная в куб, и возвращается результат этого вычисления.

В методе `Volume()` класса `Tetrahedron` вычисляется объем тетраэдра как длина ребра `length`, возведенная в куб, умноженная на квадратный корень из двух и разделенная на двенадцать, и возвращается результат этого вычисления.

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 классов.