

**Пояснительная записка**  
**Для задания №1 по архитектуре**  
**вычислительных систем**

Мельник Всеволод  
БПИ207  
10 октября 2021 г.

## Описание полученного задания

Разработать консольное приложение, позволяющее работать с контейнером трёхмерных геометрических фигур. Фигура может быть шаром с целочисленным радиусом, параллелепипедом с тремя целочисленными рёбрами или правильным тетраэдром. Каждая фигура имеет плотность. Также программа может удалить из контейнера те фигуры, для которых площадь поверхности меньше, чем средняя площадь поверхности всех фигур в контейнере.

Работа с программой осуществляется из командной строки одним из следующих способов:

**<Имя программы> -f <входной файл> <выходной файл>**  
**<выходной файл для контейнера после выполнения функции 19>** – ввод данных в программу из файла.

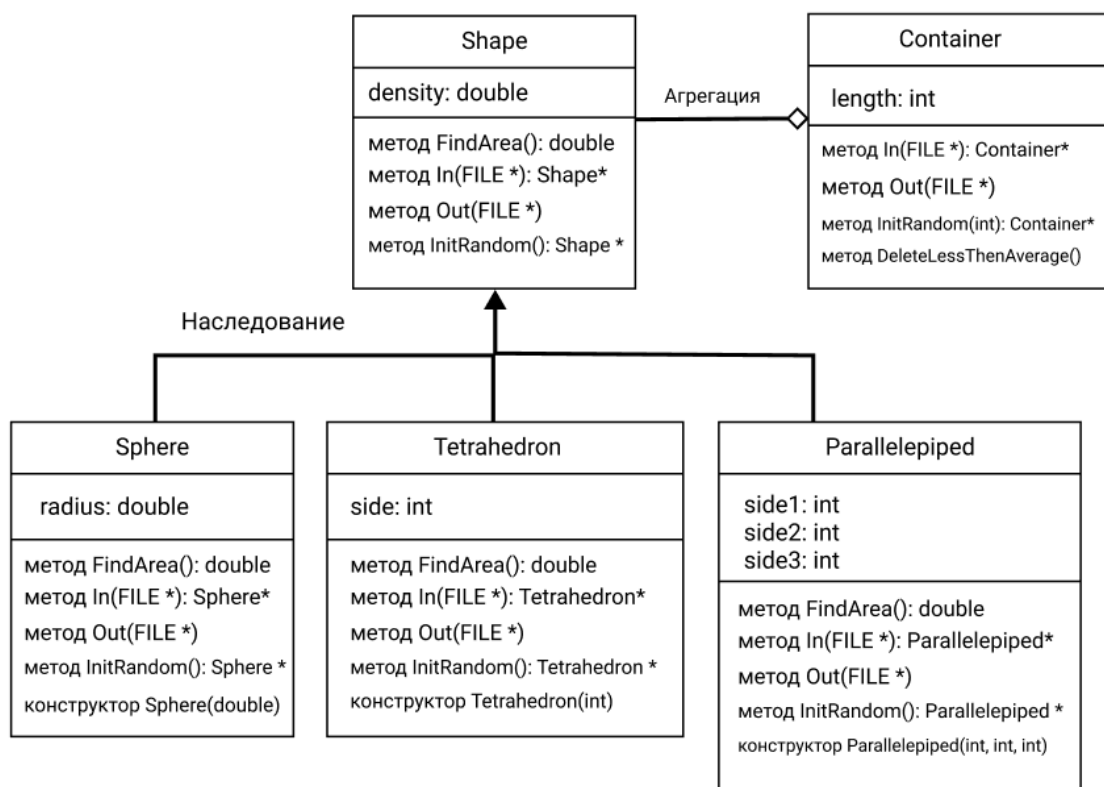
**<Имя программы> -g <количество фигур> <выходной файл>**  
**<выходной файл для контейнера после выполнения функции 19>** – случайная генерация фигур для заполнения контейнера.

**<Имя программы> -h** – справка по программе.

**Важно:** вывод контейнера после удаления фигур осуществляется в отдельный файл для упрощения работы с программой и её тестированием. Его указывать не обязательно. Если этот файл не указан, удаление и последующий вывод осуществляться **не будет**.

Тесты хранятся в папке tests, ответы на тесты - tests\_answers, ответы на тесты после выполнения функции удаления - tests\_function\_answers.

# Структурная схема изучаемой архитектуры ВС с размещенной на ней разработанной программы



## Метрики, определяющие характеристики программы

Код программы содержит 5 интерфейсных модулей и 6 модулей реализации.

Исходный код занимает ~ 11 КБ памяти, примерно на 1 КБ больше, чем в предыдущем задании, исполняемый файл - 58 КБ, примерно на 3 КБ больше, чем в предыдущем задании.

Время выполнения для одной фигуры - 0,008 секунд, для 5000 фигур - 0,048 секунд, для 10000 фигур - 0,070 секунд. Программа работает на 10-30% быстрее, чем предыдущая, которая была написана в процедурной парадигме.

Объектно-ориентированный подход позволил немного ускорить работу программы и значительно ускорить её разработку.