|  |
| --- |
| **Министерство образования и науки Российской Федерации**  Федеральное государственное автономное образовательное  учреждение высшего образования  **«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  **ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»** |

Подразделение: Инженерная школа энергетики

Направление подготовки: 09.04.03 – Прикладная информатика

Отделение: Электроэнергетики и электротехники

**Проектная документация**

**Отчёт по лабораторной работе №5**

по дисциплине: «Основы объектно-ориентированного программирования»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент гр. О-5КМ11 | |  | \_\_\_\_\_\_\_\_\_\_ | Подглазова В.В.. | |
|  | |  |  |  | |
|  | |  |  | 02 июня 2023 г. | |
|  | |  |  |  |  |
| Отчёт принял | доцент, к.т.н. |  | \_\_\_\_\_\_\_\_\_\_ |  | Калентьев А. А. |
|  |  |  |  |  |  |
|  |  |  |  | \_\_\_\_\_\_\_\_\_ 2023 г. | |

Томск 2023

**Содержание**

[ВВЕДЕНИЕ 2](#_Toc136567948)

[1 Основная часть 3](#_Toc136567949)

[1.1 UML диаграмма вариантов использования 3](#_Toc136567950)

[1.2 UML диаграмма классов 4](#_Toc136567951)

[1.3 Описание классов, образующих связь типа «общее-частное» 6](#_Toc136567952)

**ВВЕДЕНИЕ**

Корректная и полная документация сопровождает разработку программного обеспечения (далее – ПО) от появления идеи до выпуска конечного продукта. Написание документации является обязательным критерием разработки и последующей поддержки проекта [1].

Целью данной лабораторной работы является разработка проектной документации на созданный программный продукт.

Для достижения поставленной цели должны быть выполнены следующие задачи:

* Составление UML диаграммы вариантов использования для разработанной программы;
* Составление UML диаграммы классов;
* Описание классов, образующих связь типа «общее-частное»;

**1 Основная часть**

**1.1 UML диаграмма вариантов использования**

Вариант использования (use case) — это описание множества последовательных действий (включая вариации), которые выполняются некоторым субъектом с целью получения результата, значимого для некоторого действующего лица [1]. ВИ предполагает взаимодействие действующих лиц и системы или другого объекта. Действующее лицо представляет собой логически связанное множество ролей, которые играют пользователи системы во время взаимодействия с ней.

Диаграмма вариантов использования для разработанного ПО приведена на рисунке 1.

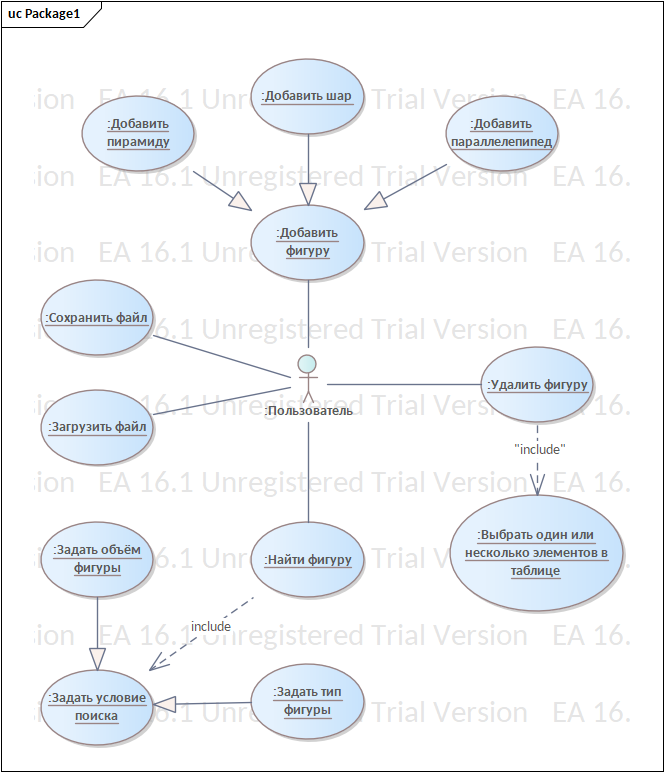


Рисунок 1 – Диаграмма вариантов использования

**1.2 UML диаграмма классов**

Диаграмма классов — это методика моделирования, которая используется практически во всех объектно-ориентированных методах. Эта диаграмма описывает типы объектов в системе и различные виды статических отношений, которые существуют между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования.

Диаграмма классов приведена на рисунке 2.

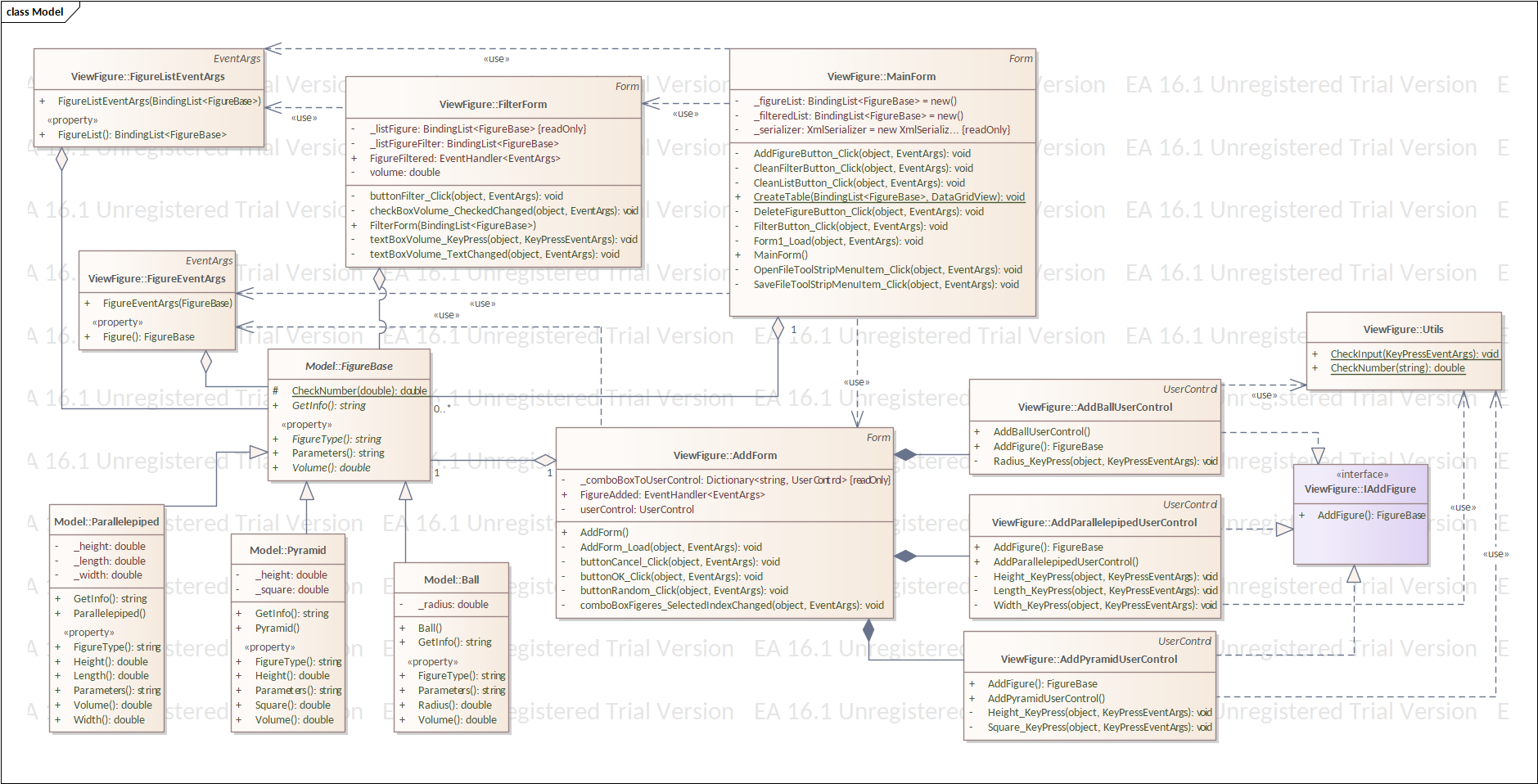


Рисунок 2. UML диаграмма классов

**1.3 Описание классов, образующих связь типа «общее-частное»**

В таблице 1 приведено описание абстрактного класса *FigureBase* с его полями, свойствами и методами.

Таблица 1 – Описание класса *FigureBase*

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс *FigureBase* – абстрактный базовый класс для объёмных фигур | | |
| Свойства | | |
| + FigureType | string | Тип фигуры.  Абстрактное свойство, переопределяется в производных классах. |
| + Volume | double | Объём фигуры.  Абстрактное свойство, переопределяется в производных классах. |
| + Parametrs | string | Параметры фигуры.  Абстрактное свойство, переопределяется в производных классах. |
| Методы | | |
| # CheckNumber | double | Проверяет корректность ввода параметров фигур  double – любой параметр фигуры |
| + GetInfo | string | Информация и фигуре.  Абстрактный метод, переопределяется в производных классах. |

В таблицах 2–4 приведены описания классов Ball, Pyramida и Parallelepiped, которые наследуются от *FigureBase*.

Таблица 2 – Описание класса Parallelepiped

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс Parallelepiped – параллелепипед | | |
| Поля | | |
| –\_length | double | Длина параллелепипеда |
| – \_width | double | Ширина параллелепипеда |
| – \_height | double | Высота параллелепипеда |
| Свойства | | |
| + Length | double | Длина параллелепипеда |
| + Width | double | Ширина параллелепипеда |
| + Height | double | Высота параллелепипеда |
| + FigureType | string | Тип фигуры |
| + Parametrs | string | Параметры фигуры |
| + GetInfo | string | Информация и фигуре |
| + Volume | double | Вычисление объёма параллелепипеда |

Таблица 3 – Описание класса Pyramida

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс Pyramida – пирамида | | |
| Поля | | |
| –\_height | double | Высота пирамиды |
| – \_square | double | Площадь основания пирамиды |
| Свойства | | |
| + Square | double | Ширина пирамиды |
| + Height | double | Высота пирамиды |
| + FigureType | string | Тип фигуры |
| + Parametrs | string | Параметры фигуры |
| + GetInfo | string | Информация и фигуре |
| + Volume | double | Вычисление объёма пирамиды |

Таблица 4 – Описание класса Ball

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс Ball – шар | | |
| Поля | | |
| –\_radius | double | Радиус шара |
| Свойства | | |
| + Radius | double | Радиус шара |
| + FigureType | string | Тип фигуры |
| + Parametrs | string | Параметры фигуры |
| + GetInfo | string | Информация и фигуре |
| + Volume | double | Вычисление объёма шара |