|  |
| --- |
| **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.

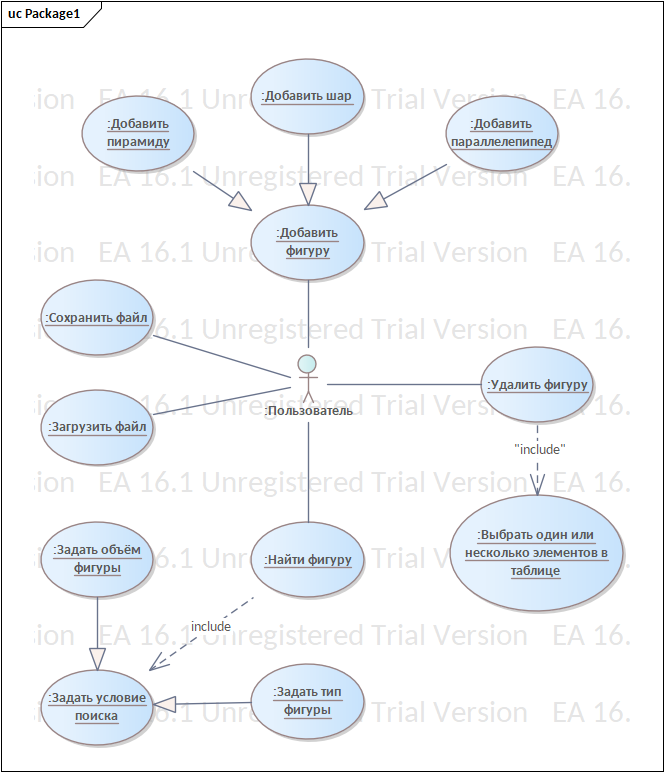


Рисунок 1 – Диаграмма вариантов использования

**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.

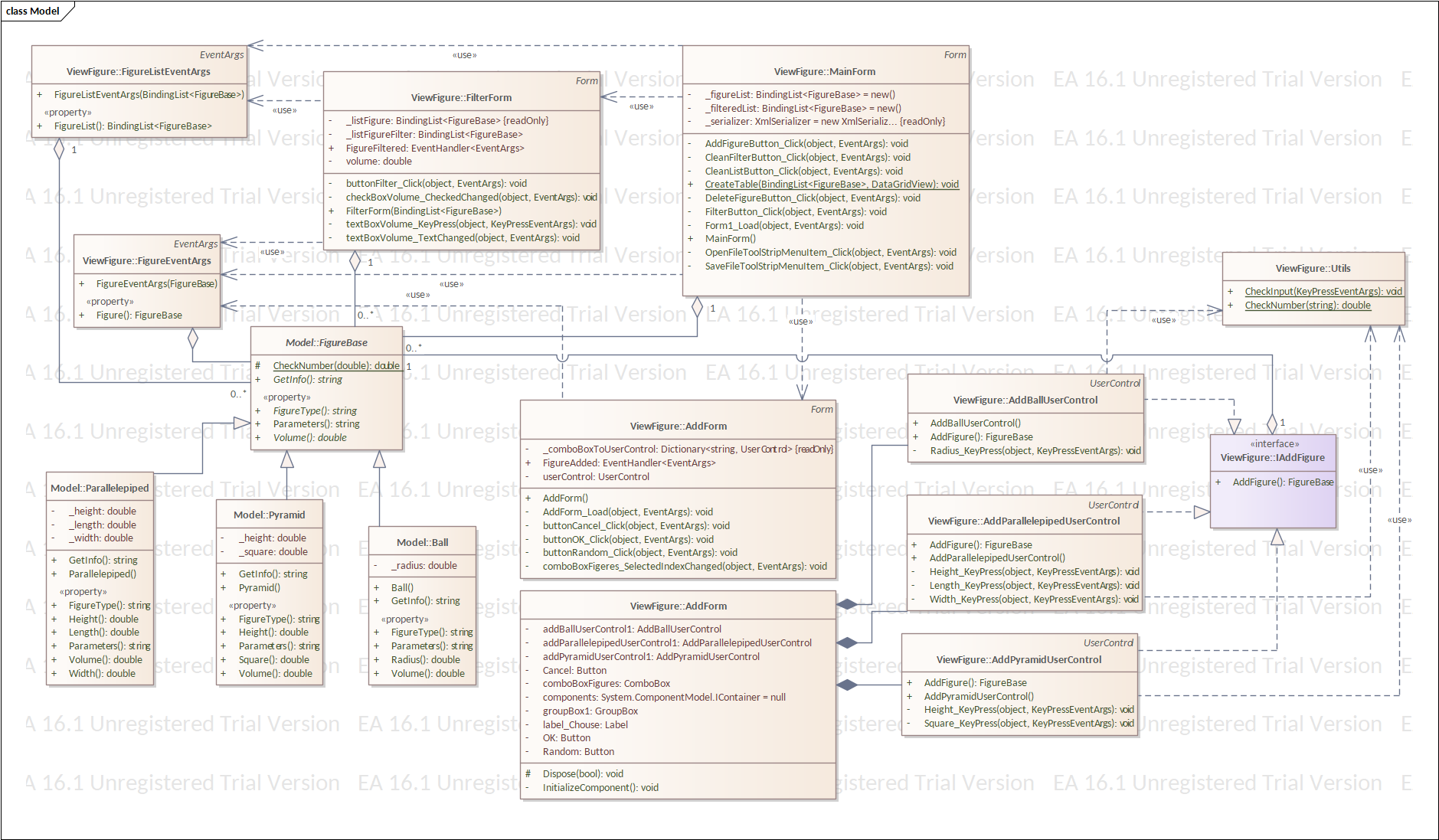


Рисунок 2. UML диаграмма классов

**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 |

**ПРИЛОЖЕНИЕ А**

**Техническое задание на создание автоматизированной системы**

Программа для расчёта объёмных фигур

Разработчик: студент гр. О-5КМ11 НИ ТПУ Подглазова В.В.

Заказчик: Канд. техн. наук, доцент каф. КСУП ТУСУР Калентьев А. А.

Томск 2023

**1 Общие сведения**

**1.1 Полное наименование системы и её условное обозначение**

Полное наименование: «Программа для расчёта объёмных фигур».

Условное обозначение: Система.

**1.2 Наименование предприятий разработчика и заказчика системы**

Заказчик: Канд. техн. наук, доцент каф. КСУП ТУСУР Калентьев А. А.

Разработчик: Студент гр. О-5КМ11 НИ ТПУ Подглазова В.В.

**1.3 Плановые сроки начала и окончания работы по созданию системы**

Начало работ: 17 апреля 2023 г.

Окончание работ: 02 июня 2021 г.

**2 Назначение и цели создания системы**

**2.1 Назначение системы**

Система предназначена для расчёта объёмов трёхмерных фигур: шара, пирамиды, параллелепипеда.

**2.2 Цели создания системы**

Система создаётся в целях сокращения трудозатрат школьников-гуманитариев при расчётах объемов трёхмерных фигур.

**3 Характеристика объектов автоматизации**

Вычисление объемов различных тел может пригодиться в любой сфере жизнедеятельности человека. Например, при выполнении домашнего задания по геометрии в 10 классе.

Поскольку такие расчёты выполняются школьниками при выполнении домашних заданий ежедневно, представляется целесообразным автоматизировать этот процесс.

**4 Требования к системе**

Таблица 4.1 – Префиксы мнемонических идентификаторов требований и их расшифровка

|  |  |
| --- | --- |
| Префикс | Тип требования |
| A | Архитектурное требование |
| С | Требование к программной или аппаратной совместимости |
| D | Требование к структуре данных |
| F | Функциональное требование |
| U | Требование к пользовательскому интерфейсу |

**4.1 Требования к архитектуре**

**А01.** Система должна быть реализована в виде настольного приложения.

**4.2 Требования к структуре данных**

**D01.** Данные о параметрах элементов электрических схем должны храниться в XML-файле с расширением \*.di.

**4.3 Функциональные требования**

**F01.** Система должна рассчитывать объём для следующих фигур:

* параллелепипед;
* пирамида;
* шар (сфера).

**F01.01.** Объём параллелепипеда должен определяться по выражению

где *a* – длина параллелепипеда, м;

*b* – ширина параллелепипеда, м;

*c* – высота параллелепипеда, м.

**F01.02.** Объём пирамиды должен определяться по выражению

где *h* – высота пирамиды, м;

*S* – площадь основания пирамиды, м2.

**F01.03.** Объём шара должен определяться по выражению

где *R* – радиус шара, м;

**F02.** В системе должен быть реализован список элементов электрических схем.

**F02.01.** Каждый элемент должен иметь следующие параметры:

* тип фигуры;
* объём.

**F03.** В системе должна присутствовать функция добавления элементов в список.

**F04.** В системе должна присутствовать функция удаления элементов из списка.

**F05.** В системе должна присутствовать функция поиска элементов по параметрам, указанным в **F02.01**.

**F06.** В системе должна присутствовать функция сохранения списка элементов в файл (**D01**).

**F07.** В системе должна присутствовать функция загрузки списка элементов из файла (**D01**).

**4.4 Требования к пользовательскому интерфейсу**

**U01.** Система должна иметь графический интерфейс пользователя.

**U02.** Данные должны быть представлены в табличном виде.

**U03.** В системе должна быть реализована система обработки ошибок.

**4.5 Требования к программному обеспечению**

**C01.** Система должна работать на операционной системе Windows 10. Работоспособность на других выпусках и версиях не гарантируется.

**C02.** На рабочей станции должен быть установлен .NET Framework версии 6.0.

**4.6 Требования к аппаратному обеспечению**

**C03.** Процессор – не менее 2500 МГц.

**C04.** ОЗУ – не менее 2 ГБ.