Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное Образовательное учреждение высшего образования «Уфимский университет науки и технологий»

Факультет информатики и робототехники

Кафедра ВМиК

Отчет по лабораторной работе №3

по дисциплине «Проектирование и конструирование ПО»

на тему: ««Методология объектно-ориентированного моделирования.

Этап создания физической модели»

Выполнили:

Студенты группы ПРО-233Б Ермолов И.О.

Хакимов И.А.

Проверил:

Преподаватель Насыров Р.В.

Уфа - 2023

**Лабораторная работа №3**

**1. Цель работы:**

Ознакомление с основными элементами определения, представления, проектирования и моделирования программных систем с помощью языка UML.

**Задание:**

1. Определиться с диаграммами из семейства UML моделей на этапе создания физической модели автоматизированной системы.
2. Разработать UML диаграммы этапа создания физической модели.
3. Задокументировать прецеденты.

Типичное описание должно содержать следующие разделы.

* 1. Краткое описание.
  2. Участвующие субъекты.
  3. Предусловия, необходимые для инициирования прецедента.
  4. Детализированное описание потока событий, которое включает: основной поток, который можно разбить для того, чтобы показать подчиненные потоки событий (подчиненные потоки могут быть разделены дальше на еще более мелкие потоки, с целью сделать читаемость документа более удобной); альтернативные потоки для определения исключительных ситуаций.
  5. Постусловия, определяющие состояние системы, по достижении которых прецедент завершается.

1. Записать не менее 10 требований согласно синтаксиса требований:

[обстоятельства] [субъект] [действие] [объект] [ограничение].

Пример: Когда сигнал получен [обстоятельства] система [субъект] должна установить [действие] разряд сигнала [объект] в течение двух секунд [ограничение].

1. Сформулировать нефункциональные требования.
2. Разработать алгоритм обработки данных.
   1. Алгоритм обработки должен быть реализован с помощью объектно-ориентированного подхода. Обработчик данных реализуется в отдельном классе (**DataProcessor**), который имеет 3 базовых метода: чтение источника данных (**read**), запуск обработки данных (**run**), вывод результата на экран (**print\_result**).
   2. Реализовать в классе необходимые методы обработки (например, очистка, назначение категорий и т.п.).

2

* 1. Использовать паттерн «Фабрика» (Factory) для вызова различных экземпляров DataProcessor в зависимости от типа входного набора данных (например, csv-файл, txt-файл).
  2. Загрузить код на **GitHub**
  3. Разработать UML схему классов обработчика данных.

1. **Написать отчет.** Отчет должен включать комплекс статических и динамических моделей, описание алгоритма обработки данных, скриншоты с результатами выполнения алгоритма (фрагмент входного набора данных и вывод после выполнения обработки), скриншот GitHub-репозитория с загруженными файлами проекта.

3

**Ход работы**

**Нефункциональные требования:**

1. Нефункциональные требования 1.1.Интерфейс пользователя
2. Интерфейс пользователя должен быть на русском языке.
3. Система должна отображать корректно интерфейс Пользователя с разрешением от 1024x600 пикселей.

1.2.Поддержка браузеров

1. Система должна работать для следующих браузеров последних версий:

MS Internet Explorer,

Mozilla Firefox, Google Chrome, Safari, Opera.

1.3.Требования к производительности

1. Система должна стабильно работать с глубиной истории не менее чем

в 3 года.

1.3.2. Система должна стабильно работать при 500 одновременно подключенных пользователей.

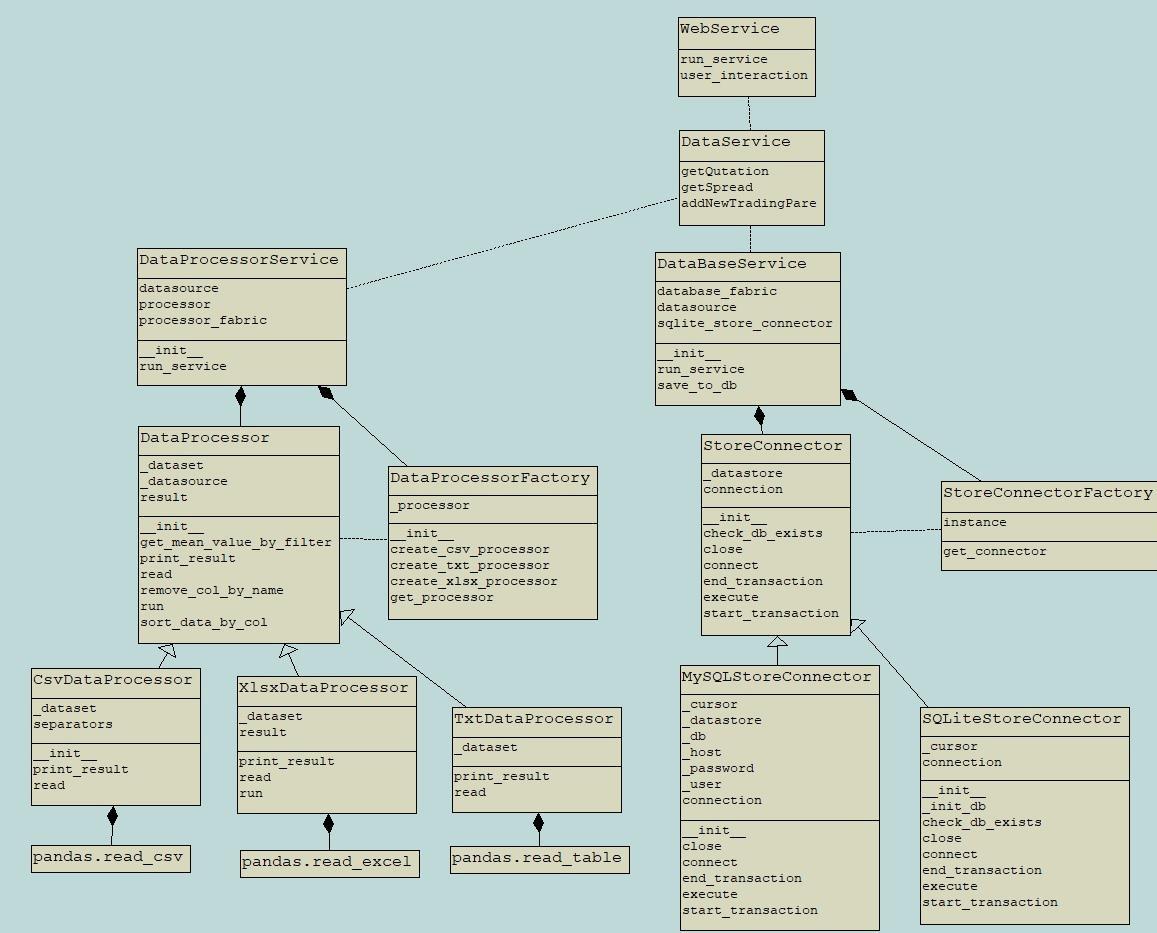
1.3.3. Система должна загружать любую форму не дольше, чем 5 секунд.

1.4.Требования к безопасности

1.4.1. Система НЕ должна позволять НЕ Администраторам физический доступ к интерфейсу администратора.

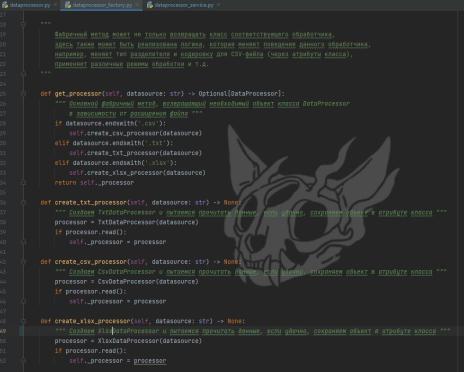
4

**UML-диаграмма классов**

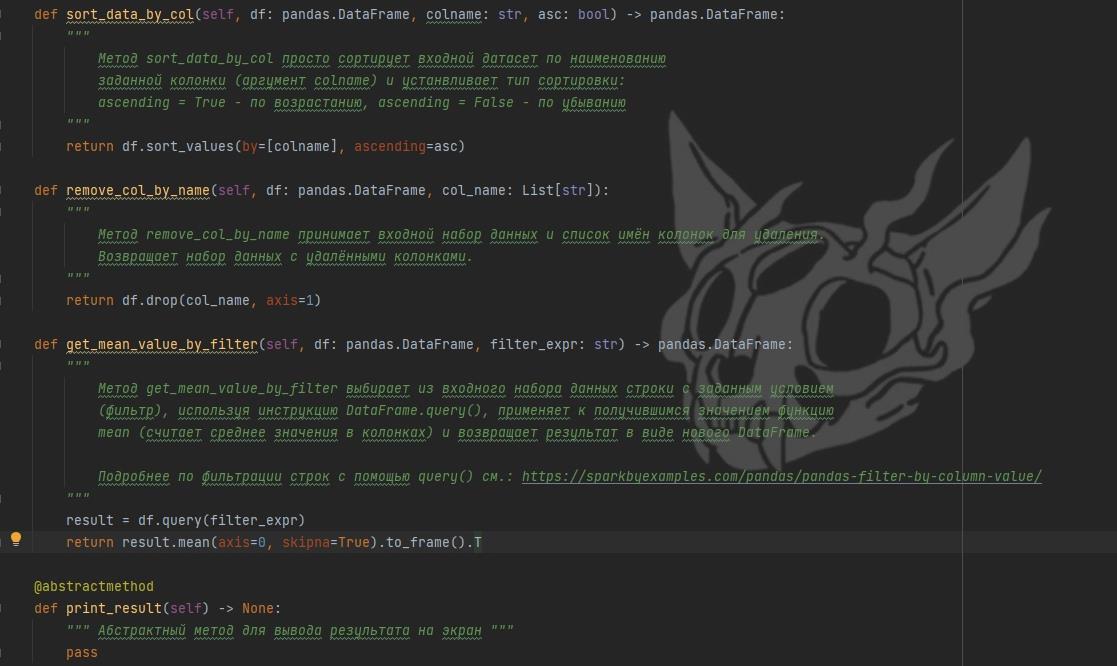


5

**Factory**

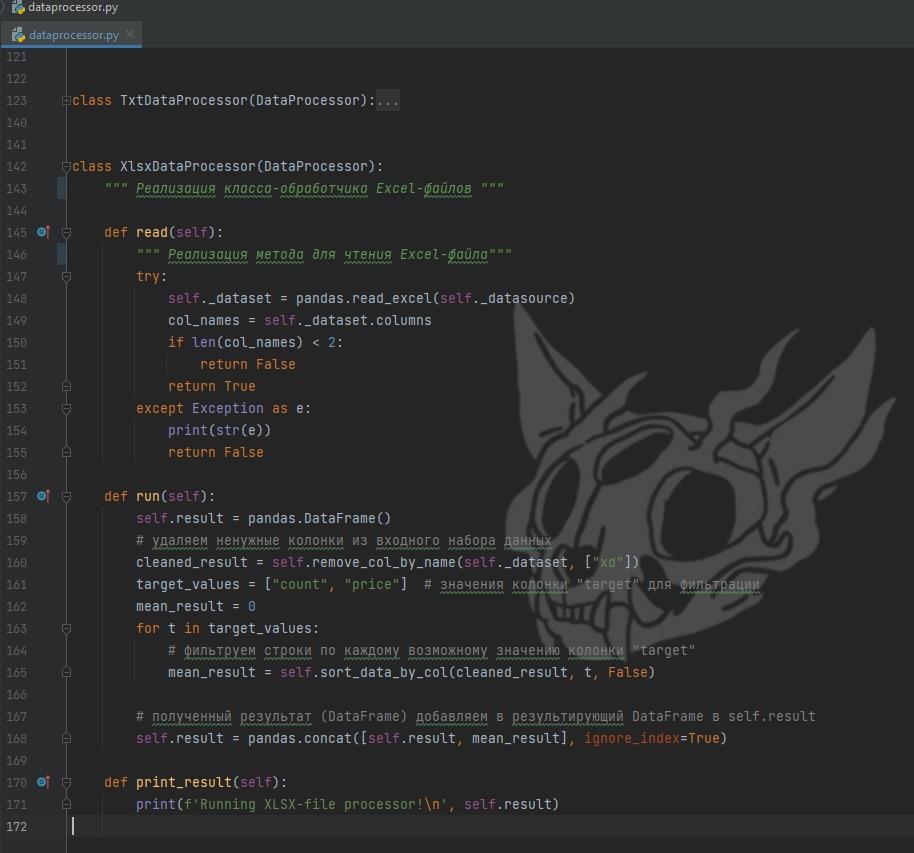


**Методы обработки данных**



6

**Обработчик Excel-файлов**

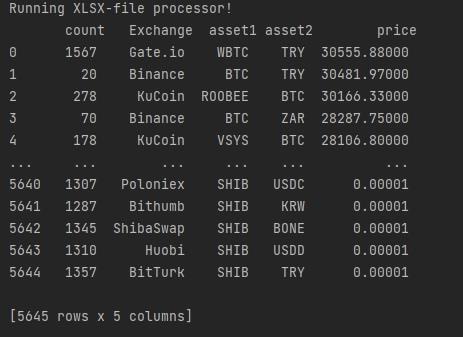


**Алгоритм обработки данных**

Метод **run()** реализует обработку данных. Он удаляет ненужные столбцы с датами при помощи метода **remove\_col\_by\_name(***df DataFrame, filter []).* Затем с помощью функции **sort\_data\_by\_col**(*df DataFrame, colname, asc)* сортирует входной датасет по наименованию заданной колонки (аргумент *colname*) и устанавливает тип сортировки: **ascending = True** – повозрастанию, **ascending = False** – по убыванию.

7

**Обработанные данные**



|  |  |
| --- | --- |
| **Прецеденты** |  |
| Прецедент | Регистрация пользователя |
|  |  |
| Краткое описание | Данный прецедент необходим для регистрации пользователя |
|  | в системе. |
|  |  |
| Субъект | Система, незарегистрированный пользователь |
|  |  |
| Предусловия | Пользователь должен открыть форму для регистрации |
|  |  |
| Основной поток | Пользователь заполняет форму и отправляет запрос на |
|  | регистрацию в системе |
|  |  |
| Альтернативный | Если пользователь ввел некорректные данные или не |
| поток | заполнил обязательные поля, выдается сообщение об ошибке |
|  | и предлагается заполнить форму еще раз |
|  |  |
| Постусловия | После успешного завершения прецедента данные о новом |
|  | пользователе заносятся в базу данных. |
|  |  |

|  |  |
| --- | --- |
| Прецедент | Вход пользователя |
|  |  |
| Краткое описание | Данный прецедент необходим для входа пользователя в |
|  | систему. |
|  |  |
| Субъект | Система, пользователь |
|  |  |
| Предусловия | Пользователь должен открыть форму для входа |
|  |  |
| Основной поток | Пользователь заполняет форму и отправляет запрос на вход |
|  | в систему |
|  |  |
| Альтернативный | Если пользователь ввел некорректные данные (таких данных |
| поток | нет в БД) или не заполнил обязательные поля, выдается |
|  | сообщение об ошибке и предлагается выполнить вход еще |
|  | раз |
|  |  |
| Постусловия | После успешного завершения прецедента пользователь |
|  | входит в систему. |
|  |  |

11

**Выводы к работе.**

* ходе выполнения лабораторной работы мы ознакомились с основными элементами определения, представления, проектирования и моделирования программных систем с помощью языка UML, а также разработали UML

модели для реализации нашей автоматизированной системы.

12