Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Уфимский государственный авиационный технический университет»

Кафедра ВМиК

Отчёт по лабораторной работе №3

на тему: «Методология объектно-ориентированного моделирования. Этап создания физической модели»

по дисциплине: «Проектирование и конструирование программного обеспечения»

Выполнили:

Студенты группы: ПРО-328Б С.А. Гордеева

М.А. Смирнов

И.И. Гареев

Проверил: Б.С. Юдинцев

Уфа – 2022

СОДЕРЖАНИЕ

[Цель работы 3](#_Toc119527011)

[Ход работы 4](#_Toc119527012)

[1.1 Алгоритм обработки данных 4](#_Toc119527013)

[1.2 Описание Прецедентов 5](#_Toc119527014)

[1.3 Диаграмма классов 8](#_Toc119527015)

[Вывод 9](#_Toc119527016)

# Цель работы

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

**Задание**

1. Рассмотреть материал по объектно-ориентированному моделированию

2. Определиться с диаграммами из семейства UML моделей на этапе создания физической модели автоматизированной системы (тематика выбрана в процессе лабораторной работы 1).

3. Разработать UML диаграммы этапа создания физической модели.

4. Задокументировать прецеденты.

5. Разработать алгоритм обработки данных.

6. Написать отчет.

# Ход работы

## Алгоритм обработки данных

Проверяем данные на выбросы – пустые ячейки в данных:

- при их наличии – удаляем строчки с выбросами.

Листинг кода:

col\_namess = self.\_dataset.columns

        for i in range(len(col\_namess)):

            self.\_dataset [col\_namess[i]].replace('', np.nan, inplace=True)

            self.\_dataset.dropna(subset=[col\_namess[i]], inplace=True)

Начинаем сортировку, разбиваем строчки по значениям цены на сегменты: эконом, средний, люкс (соответсвенно присваиваем категории 1, 2 и 3) – колонка price\_category.

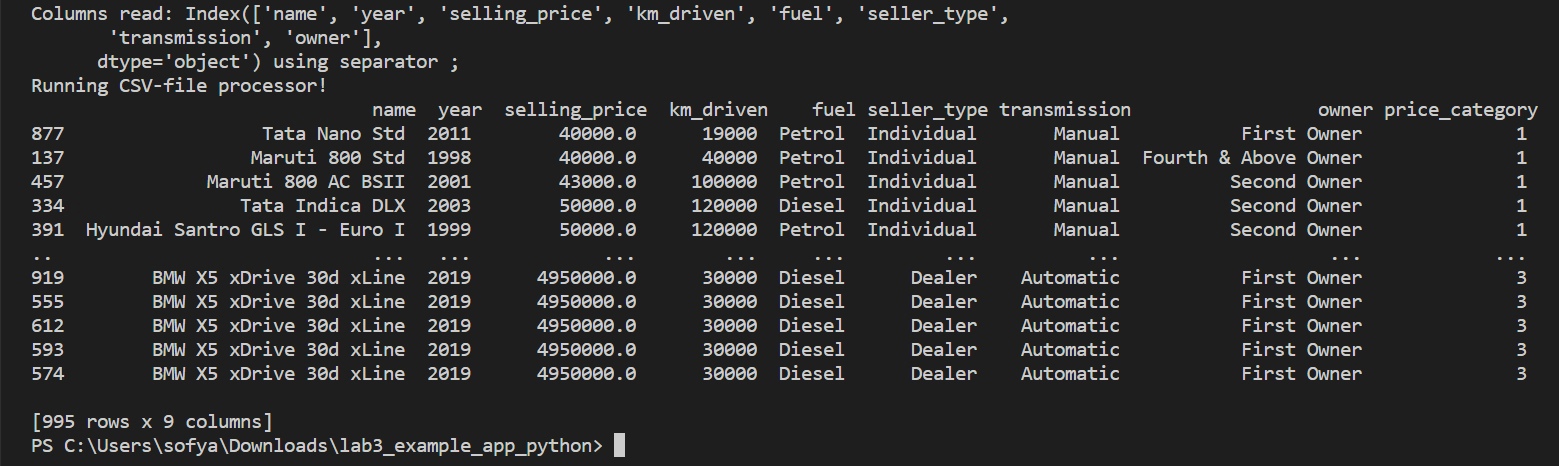
Листинг кода:

# создаем колонку для разбития на категории

        self.\_dataset ["price\_category"] = pandas.cut(self.\_dataset["selling\_price"],[0,500000,1500000,np.inf], labels = [1,2,3])

        self.result = self.sort\_data\_by\_col(self.\_dataset, "selling\_price", True)

Пример работы:



## Описание Прецедентов

1. «Открытие файла»

|  |  |
| --- | --- |
| Прецедент | Открытие файла |
| Краткое описание | Данный прецедент необходим для открытия файла в обработчике данных. |
| Субъект | Обработчик данных. |
| Предусловия | Указали источник данных (csv-файл) для открытия, загрузили его в папку с обработчиком данных. |
| Основной поток | В случае успешного открытия файла – считываем данные. |
| Альтернативный поток | В случае если файл не открылся – выводим сообщение об ошибке. Пытаемся заново открыть файл. |
| Постусловия | После успешного завершения прецедента, обработчик данных сможет считать данные. |

1. «Считывание данных»

|  |  |
| --- | --- |
| Прецедент | Считывание данных |
| Краткое описание | Данный прецедент необходим для считывания данных. |
| Субъект | Обработчик данных. |
| Предусловия | Открыли файл. |
| Основной поток | C помощью библиотеки pandas считываем данные (пытаемся преобразовать данные файла в pandas.DataFrame, используя различные разделители). |
| Альтернативный поток | В случае неудачи – выводим сообщение об ошибке. |
| Постусловия | После успешного завершения прецедента, обработчик данных сможет обработать данные. |

1. «Обработка данных»

|  |  |
| --- | --- |
| Прецедент | Обработка данных |
| Краткое описание | Данный прецедент необходим для обработки данных. |
| Субъект | Обработчик данных. |
| Предусловия | Считали данные из файла. |
| Основной поток | Проверяем данные на выбросы:  - при их наличии – удаляем строчки с выбросами.  Начинаем сортировку, разбиваем строчки по значениям цены на сегменты: эконом, средний, люкс. |
| Альтернативный поток | Отсутствует. |
| Постусловия | После успешного завершения прецедента, обработчик данных сможет передать данные в БД. |

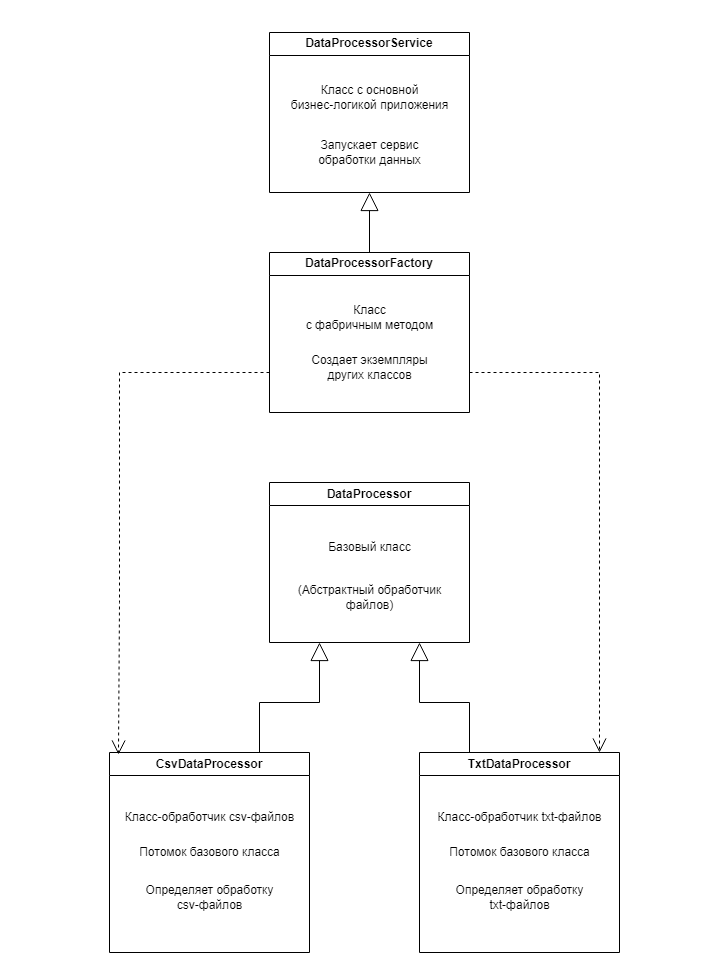
1. «Установка соединения с БД»

|  |  |
| --- | --- |
| Прецедент | Установка соединения с БД |
| Краткое описание | Данный прецедент необходим для установки соединения с БД. |
| Субъект | Обработчик данных, БД. |
| Предусловия | Обработали данные. |
| Основной поток | В случае успешного соединения – загружаем обработанные данные в БД. |
| Альтернативный поток | В случае если соединение не было установлено – выводим сообщение об ошибке. Возвращаем БД к этапу ожидания данных. |
| Постусловия | После успешного завершения прецедента, в БД появятся данные. |

1. «Поиск авто»

|  |  |
| --- | --- |
| Прецедент | Поиск авто |
| Краткое описание | Данный прецедент необходим для поиска авто на сайте. |
| Субъект | Пользователь. |
| Предусловия | Пользователь должен зайти на сайт. |
| Основной поток | Пользователь выбирает необходимые для поиска любые из следующих параметров фильтрации по цене  Далее система отображает список авто по выбранным параметрам. |
| Альтернативный поток | Отсутствует. |
| Постусловия | После успешного завершения прецедента, пользователю представлен список интересующих авто. |

## Диаграмма классов



# Вывод

В ходе данной лабораторной работы, мы ознакомились с основными элементами определения, представления, проектирования и моделирования программных систем с помощью языка UML.

Ссылка на Github:

<https://github.com/MaxSmir/PiKPO_Project>