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

Отчёт по лабораторной работе № 3

на тему: «Методология объектно-ориентированного моделирования»

Выполнили:

Студенты группы ПРО-232Б В.Д. Свайкина, Д. М. Григорьев

Проверил:

Преподаватель Б. С. Юдинцев

Уфа – 2023

# Цель работы:

Ознакомление с основными элементами определения, представления,

проектирования и моделирования программных систем с помощью языка UML.

# Ход работы:

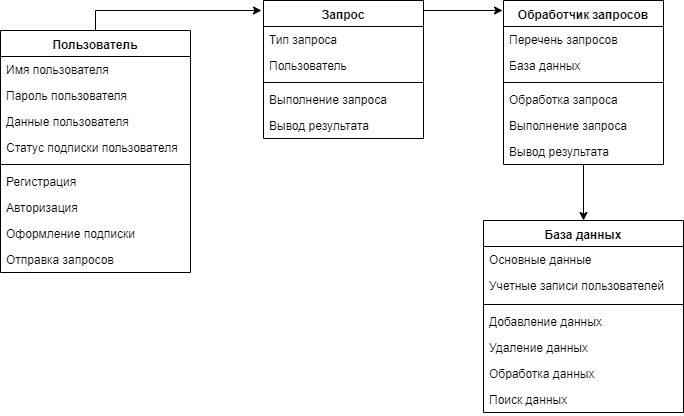


Рисунок 1 Диаграмма классов

Диаграмма классов отражает используемые классы и взаимосвязи между ними. Отражает атрибуты и методы каждого из классов.

Схема классов обработчика данных отражает классы обработчика данных и взаимосвязи между ними. Также показывает методы этих классов.

# Алгоритм обработки данных:

Алгоритм обработки реализован с помощью объектно-ориентированного подхода на языке программирования Python. В классе реализованы различные методы для работы с данными. Присутствуют следующие методы:

* remove\_col\_by\_name() – для удаления столбцов.
* sort\_data\_by\_col() – для сортировки данных.
* print\_result() – для вывода результата.
* И другие

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 2 Схема классов обработчика данных

# Прецеденты:

Таблица 1 Прецедент "Регистрация"

|  |  |
| --- | --- |
| **Прецедент** | Регистрация |
| **Краткое описание** | Данный прецедент описывает процесс создания учетной записи пользователя на сайте |
| **Субъект** | Пользователь, сайт. |
| **Предусловия** | Пользователь должен иметь доступ к сайту на котором будет производиться регистрация. Также необходимо, чтобы пользователь имел действительный адрес электронной почты или номер телефона. |
| **Основной поток** | 1. Пользователь открывает страницу регистрации.  2. Система запрашивает у пользователя необходимую информацию для регистрации, такую как имя, фамилию, электронную почту или номер телефона, пароль.  3. Пользователь вводит необходимую информацию и нажимает кнопку "Зарегистрироваться".  4. Система проверяет корректность введенных данных и создает новую учетную запись для пользователя.  5. Система отправляет подтверждающее письмо или сообщение пользователю для подтверждения регистрации. |
| **Альтернативный поток** | 1. Пользователь не вводит корректные данные.  2. Система выводит сообщение об ошибке и просит пользователя исправить введенные данные.  3. Пользователь исправляет данные и повторяет попытку регистрации. |
| **Постусловия** | Пользователь успешно зарегистрирован на сайте и может использовать свою учетную запись для входа. |

Таблица 2 Прецедент "Авторизация"

|  |  |
| --- | --- |
| **Прецедент** | Авторизация |
| **Краткое описание** | Позволяет пользователю получить доступ к системе, используя свои учетные данные. |
| **Субъект** | Пользователь, сайт. |
| **Предусловия** | Пользователь должен иметь учетную запись в системе и знать свой логин и пароль. |
| **Основной поток** | 1. Пользователь открывает страницу авторизации.  2. Система запрашивает у пользователя логин и пароль.  3. Пользователь вводит свои данные.  4. Система проверяет правильность введенных данных.  5. При успешной проверке система предоставляет пользователю доступ к системе. |
| **Альтернативный поток** | 1. Пользователь не вводит корректные данные.  2. Система выводит сообщение об ошибке и просит пользователя исправить введенные данные.  3. Пользователь исправляет данные и повторяет попытку входа. |
| **Постусловия** | Пользователь получает доступ к системе и может начать работать с сайтом. |

Таблица 3 Прецедент "Сортировка"

|  |  |
| --- | --- |
| **Прецедент** | Сортировка |
| **Краткое описание** | Позволяет пользователю упорядочивать данные в соответствии с заданными критериями. |
| **Субъект** | Пользователь, сайт. |
| **Предусловия** | В системе имеются данные, которые можно отсортировать. |
| **Основной поток** | 1. Пользователь выбирает критерий, по которому нужно отсортировать данные.  2. Пользователь выбирает порядок сортировки (по возрастанию/убыванию).  3. Система сортирует данные согласно выбранным критериям и порядку.  4. Отсортированные данные отображаются на экране. |
| **Альтернативный поток** | 1. Несуществующий критерий – сообщение об ошибке. 2. Несуществующий порядок сортировки – сообщение об ошибке. |
| **Постусловия** | Данные отсортированы в соответствии с выбранными критериями и порядком, и отображены на экране. |

Таблица 4 Прецедент "Фильтрация"

|  |  |
| --- | --- |
| **Прецедент** | Фильтрация |
| **Краткое описание** | Используется для отображения определенного набора данных в системе, соответствующих заданным параметрам фильтрации. |
| **Субъект** | Пользователь, сайт. |
| **Предусловия** | Пользователь должен быть авторизован в системе и иметь доступ к данным, которые он хочет отфильтровать. |
| **Основной поток** | 1. Пользователь запускает функцию фильтрации.  2. Система отображает форму с параметрами фильтрации.  3. Пользователь заполняет параметры фильтрации в соответствии с требуемыми данными.  4. Пользователь нажимает кнопку "Применить".  5. Система фильтрует данные в соответствии с заданными параметрами фильтрации.  6. Система отображает отфильтрованные данные на экране. |
| **Альтернативный поток** | 1. Пользователь отменяет функцию фильтрации.  2. Система возвращает пользователя на предыдущую страницу. |
| **Постусловия** | Пользователь может просмотреть данные, которые соответствуют заданным параметрам фильтрации. |

Таблица 5 Прецедент "Нет доступа к базе данных"

|  |  |
| --- | --- |
| **Прецедент** | Нет доступа к базе данных |
| **Краткое описание** | Процесс обработки случая, когда пользователь не имеет доступа к базе данных. |
| **Субъект** | Пользователь, сайт, база данных. |
| **Предусловия** | Система должна быть в рабочем состоянии и база данных должна быть подключена. Пользователь должен иметь аккаунт и должен быть авторизован в системе. |
| **Основной поток** | 1. Система выдает ошибку о невозможности доступа к базе данных.  2. Пользователь пытается повторно подключиться к базе данных.  3. Если повторная попытка не удалась, пользователь обращается в техническую поддержку. |
| **Альтернативный поток** | 1. Пользователь проверяет свои права доступа к базе данных.  2. Если он обнаруживает ошибку в правах доступа, он обращается к администратору базы данных, чтобы исправить ситуацию. |
| **Постусловия** | Техническая поддержка устанавливает причину отсутствия доступа и восстанавливает его в случае необходимости. Пользователь может продолжить работу в системе. |

**Приложение:**

Ссылка на GitHub: <https://github.com/Kilaxa/pikpo3.git>

# Результат работы:

Изображение выглядит как текст

Автоматически созданное описание