**Московская предпрофессиональная олимпиада школьников**

Кейсовые задачи.

Инженерно-конструкторское направление. Профиль Информационные технологии

**Кейс №5**

**Сбор и обработка данных температуры**

Оглавление

[1. Анализ технических требований 3](#__RefHeading___Toc1228_1083481191)

[2. Обоснование выбора языка программирования и используемых программных средств 3](#__RefHeading___Toc1140_1083481191)

[3. Описание основных этапов разработки 4](#__RefHeading___Toc2560_2438665916)

[4. Структурная и функциональная и схема 4](#__RefHeading___Toc2562_2438665916)

[5. Алгоритм работы программного продукта 5](#__RefHeading___Toc2564_2438665916)

[6. Описание проведенных испытаний в соответствии с регламентом кейса 6](#__RefHeading___Toc2566_2438665916)

[7. Результаты разработки 6](#__RefHeading___Toc2568_2438665916)

[8. Программный код 6](#__RefHeading___Toc2570_2438665916)

# Анализ технических требований

Проанализировав условия кейса, наша команда определила технические требования для выполнения задачи:

* Необходимо разработать программный модуль, выполняющий сбор данных от внешнего сервиса и методы их обработки
* Обеспечить хранение данных в БД
* Реализовать программный модуль для обработки клиентских запросов
* Разработать графический интерфейс пользователя
* Произвести тестирование продукта

# Обоснование выбора языка программирования и используемых программных средств

Таблица 1: Выбор инструмента

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критерий\название | Python | Java | Node.js | C++ |
| Скорость | - | - | - | + |
| Опыт работы | + | - | - | - |
| Удобство | + | - | + | - |
| Наличие необходимых библиотек | + | - | - | - |
| Ресурсозатратность | - | - | + | + |
| Предпочтения участников | + | - | + | - |

Таблица 2: Выбор базы данных

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий\название | MongoDB | MySQL | PostgreSQL |
| Удобство использования | + | + | - |
| Оптимизация под хранение данных | + | - | - |
| Скорость | + | - | - |

В результате проведенного анализа существующих инструментов мы решили использовать язык Python и базу данных MongoDB для этого кейса.

# Описание основных этапов разработки

На первом этапе разработки наша команда ознакомилась с условиями выполнения кейса и типом конечного продукта. Исходя из этого, было решено создать программный модуль, выполняющий функцию сбора данных, предоставляемых через API сервисом и загрузки этих данных в нашу базу данных.

На втором этапе были выбраны инструменты для выполнения поставленной задачи, разработаны алгоритмы, построены функциональные и структурные схемы для конечного продукта, способа загрузки данных в БД. Был выбран способ оптимального хранения данных в БД.

Затем был реализован программный модуль с функцией обработки клиентских запросов о значениях температуры в городах, квартирах и т. д. Параллельно с вышеперечисленными действиями проводилось оформление работы, написание текста, оформление схем.

Конечной инстанцией стала разработка графического интерфейса пользователя (UI), тестирование продукта.

# Структурная и функциональная и схема

На рисунке 1 представлена структурная схема взаимодействий между отдельными блоками конечного продукта.

ER- модель:

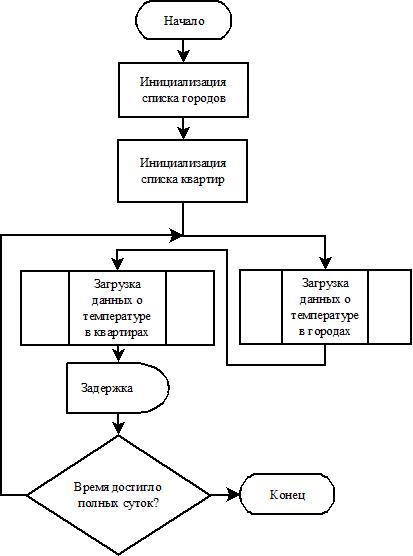
  
Рисунок 1: Структурная схема

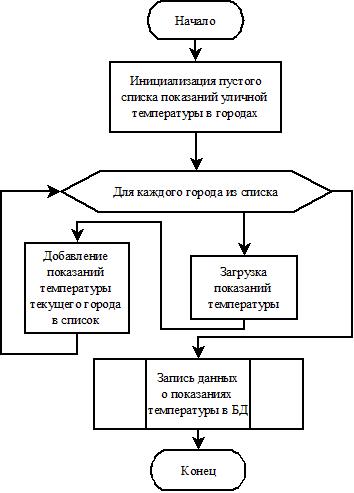
# Алгоритм работы программного продукта

На рисунке 2 представлена функциональная схема предоставленных сервисом данных о температуре для городов и квартир из списка в БД.

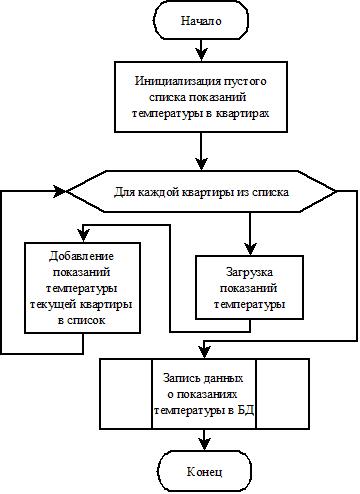
На рисунке 3 представлена функциональная схема загрузки предоставленных сервисом данных о температуре для каждого города из списка в БД.

На рисунке 4 представлена функциональная схема загрузки предоставленных сервисом данных о температуре для каждой квартиры из списка в БД.

  
Рисунок 3: Алгоритм работы сервиса

  
Рисунок 2: Функциональная схема для работы с данными городов

# Описание проведенных испытаний в соответствии с регламентом кейса

  
Рисунок 4: Функциональная схема для работы с данными квартир

Первое испытание - получение данных температуры в реальном времени в определённой квартире по запросу пользователя.

Затем был произведен сбор данных в течении суток реального времени. На основе собранных данных были построены графики, один из которых отображает изменение средней температуры в квартирах в одном из городов, а другой показывает изменение показаний уличной температуры в одном из городов.

Потом была сделана случайная выборка квартир, по одной из каждого города, на основе показаний температуры которых был построен график изменения этих показаний.

В последнем испытании была создана диаграмма, отображающая максимальную температуру, полученную в квартирах в каждом из пяти районов выбранного города.

# Результаты разработки

В ходе разработки, был создан программный код на языке Python. Нами был написан программный модуль, выполняющий сбор данных с внешнего сервиса и обеспечено хранение информации в базе данных и взаимодействие с ней.

Также с помощью фреймворка Django был реализован программный модуль, предназначенный для обработки клиентских запросов и отображения графического интерфейса пользователя. Для создания UI мы использовали HTML5, CSS3, JavaScript (JQuery). По окончанию работы было произведено тестирование продукта и оптимизация процессов.

# Программный код

https://github.com/MrFzovpec/TemperaturePerdprof