**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

**Кафедра информационных систем управления**

**Итоговый отчет по проекту**

**«Учёт наблюдений за погодой»**

Кривицкой Дарьи Сергеевны

студентки 3 курса 10 группы,

специальность «Экономическая кибернетика»

Оглавление

[Аннотация 3](#_Toc185341901)

[Анализ задач по обработке информации в выбранной предметной области (учет наблюдений за погодой) 4](#_Toc185341902)

[Существующие стандарты для документов и форм 7](#_Toc185341903)

[Существующие информационные системы и их функции 10](#_Toc185341904)

[Предложения по автоматизации выбранной предметной области 11](#_Toc185341905)

[Заключение 12](#_Toc185341906)

[Спецификация к ПО 13](#_Toc185341907)

[Use-case диаграмма по учету наблюдений за погодой 15](#_Toc185341908)

[Диаграммы активности 19](#_Toc185341909)

[Диаграммы классов 22](#_Toc185341910)

[Диаграмма классов 23](#_Toc185341911)

[Описание классов 24](#_Toc185341912)

[Описание связей 26](#_Toc185341913)

[Кодирование проекта 28](#_Toc185341914)

# Аннотация

В рамках данного задания проведен анализ предметной области "Учет наблюдений за погодой" с целью выявления задач по обработке информации и предложений по автоматизации процессов. Рассмотрены основные задачи, связанные со сбором, анализом, хранением и визуализацией данных о погодных явлениях. Изучены существующие международные и национальные стандарты для оформления метеорологических данных. Описаны действующие информационные системы, которые решают задачи прогнозирования и обмена метеорологической информацией. На основании проведенного анализа предложены пути автоматизации, включая сбор данных с метеостанций, внедрение методов машинного обучения для анализа и прогнозирования, а также создание интерфейсов для визуализации и автоматической отчетности.

# Анализ задач по обработке информации в выбранной предметной области (учет наблюдений за погодой)

Предметная область наблюдений за погодой включает разнообразные задачи, связанные с обработкой, анализом, хранением и визуализацией метеорологических данных. Эти задачи охватывают весь цикл работы с информацией — от сбора данных на местных станциях и через спутники до их анализа и предоставления прогнозов погоды для различных целей. Рассмотрим ключевые задачи по обработке информации в данной области.

***1. Сбор данных о погодных условиях***

Сбор данных — это первый и ключевой этап в процессе учета наблюдений за погодой. Для получения точных данных используются различные устройства и технологии:

* *Метеостанции* — автоматические и ручные станции для сбора данных о температуре, влажности, давлении, ветре и осадках.
* *Спутники* — космические аппараты, которые фиксируют погодные условия с помощью датчиков и камер, обеспечивая глобальные данные о состоянии атмосферы.
* *Морские буи и корабли* — используются для наблюдений за погодными условиями на водных поверхностях.
* *Мобильные сенсоры и дроны* — новейшие технологии, позволяющие оперативно собирать данные в труднодоступных местах.

***Основные данные, которые собираются****:*

* Температура воздуха;
* Влажность;
* Скорость и направление ветра;
* Атмосферное давление;
* Видимость (облака, туман);
* Количество и тип осадков;
* Состояние земли (снег, лёд, мокрая или сухая почва);
* Чрезвычайные явления (гроза, ураган, метель).

Автоматизация процесса сбора данных позволяет минимизировать ошибки, ускорить получение информации и обеспечить ее поступление в реальном времени. Это особенно важно для синоптиков и служб предупреждения об опасных природных явлениях.

***2. Хранение и управление данными***

Собранные данные нуждаются в эффективной системе хранения и управления, так как объёмы информации могут быть огромными, особенно при наблюдениях за погодой в глобальном масштабе.

* **Централизованные базы данных**: данные собираются с различных метеостанций и других источников, после чего они обрабатываются и хранятся в централизованной базе данных. Это необходимо для последующего анализа и прогноза.
* **Интеграция с системами больших данных**: системы хранения должны быть оптимизированы для работы с большими объемами данных (Big Data), а также поддерживать их интеграцию с внешними источниками — например, данными других стран или спутниковыми наблюдениями.

**\*** помимо оперативной информации, важно также хранение данных за предыдущие периоды для создания статистических отчётов, долгосрочного анализа и улучшения прогнозов.

**3. Анализ данных и прогнозирование погоды**

Анализ метеорологических данных играет центральную роль в прогнозировании погоды и обнаружении опасных погодных явлений. Эта задача требует как классических методов обработки данных, так и применения современных технологий, таких как машинное обучение и нейросетевые модели.

* **Статистический анализ**: обработка больших объемов данных для нахождения статистических корреляций, аномалий и трендов. Это позволяет улучшить понимание сезонных колебаний и климатических изменений.
* **Моделирование и прогнозирование**: на основе собранных данных и сложных математических моделей строятся прогнозы погоды на короткий (от нескольких часов до дней) и средний (до 10-15 дней) сроки. Используются как локальные данные, так и глобальные данные от международных организаций (например, Европейского центра среднесрочных прогнозов погоды).
* **Модели машинного обучения**: применение методов машинного обучения и искусственного интеллекта для более точного прогноза погоды, а также выявления паттернов изменения климата. Модели обучаются на больших объемах данных и помогают улучшить точность прогнозов.

**4. Мониторинг опасных явлений и автоматическая генерация предупреждений**

Одной из ключевых задач в системе учета погодных наблюдений является мониторинг опасных погодных явлений и своевременное информирование населения и государственных служб. Системы мониторинга фиксируют появление потенциально опасных явлений, таких как ураганы, торнадо, ливни, снегопады и другие экстремальные погодные события. При обнаружении угрозы создаются автоматические уведомления и предупреждения для служб спасения, авиационных и транспортных служб, а также населения. Эти системы должны быть достаточно быстрыми и надежными, чтобы минимизировать ущерб и предотвратить жертвы.

**5. Визуализация и представление данных**

Информационные системы для учета наблюдений за погодой должны предоставлять данные в удобном для восприятия виде:

* **Графики и диаграммы**: наглядное отображение изменений погодных условий по времени, что помогает анализировать тенденции и выявлять аномалии.
* **Картографическое представление**: визуализация погодных явлений на картах (например, карта температур, осадков, ветровых полей) помогает более наглядно и быстро оценить ситуацию в различных регионах.
* **Отчеты и сводки**: формирование периодических отчетов и сводок (ежедневных, ежемесячных, годовых) по погодным условиям, как для внутренних целей (анализ), так и для внешних организаций (синоптики, СМИ, государственные службы). Для того чтобы данные могли быть полезны и использоваться в глобальных системах, они должны соответствовать международным стандартам, таким как METAR (код для метеорологических сводок) или SYNOP (код для метеонаблюдений), для обеспечения совместимости данных.

**6. Обмен данными и интеграция с международными системами**

Поскольку погода является глобальным феноменом, важна интеграция с международными системами, чтобы оперативно получать данные из других стран и регионов, а также делиться информацией на глобальном уровне. Примеры таких систем:

* **WMO (Всемирная метеорологическая организация)** — обеспечивает глобальный обмен данными между странами для улучшения качества прогнозов.
* **ECMWF (European Centre for Medium-Range Weather Forecasts)** — предоставляет данные и прогнозы погоды на основе глобальных наблюдений.

Задачи по обработке информации в области учета наблюдений за погодой охватывают весь цикл работы с метеорологическими данными: от их сбора и хранения до анализа и передачи результатов. В условиях постоянного увеличения объёмов данных и усложнения моделей прогнозирования, автоматизация этих задач становится необходимой для повышения точности прогнозов, своевременного обнаружения опасных погодных явлений и обеспечения эффективного взаимодействия на глобальном уровне.

# Существующие стандарты для документов и форм

Метеорология — это область, в которой стандарты имеют решающее значение для обеспечения точного и своевременного обмена информацией. Международные стандарты позволяют различным странам и организациям синхронизировать свои данные и методы их обработки, а национальные стандарты адаптируют эти требования к локальным условиям и потребностям. Рассмотрим более подробно основные стандарты, форматы данных и требования на международном и национальном уровнях.

**1. Международная организация гражданской авиации (ICAO)**

**Международная организация гражданской авиации (ICAO)** устанавливает стандарты для наблюдений за погодой в аэропортах и в авиационной отрасли. Эти стандарты необходимы для обеспечения безопасности полетов, так как погодные условия оказывают непосредственное влияние на работу авиации.

**Основные требования ICAO:**

* **Стандарты для метеорологических наблюдений в аэропортах**: в аэропортах по всему миру используются одинаковые процедуры наблюдений за погодой для того, чтобы обеспечить глобальную совместимость данных и их корректную интерпретацию пилотами и авиадиспетчерами. Основной формат для таких наблюдений — это **METAR**.
* **Прогнозы погоды для авиации**: для прогнозирования условий, влияющих на безопасность полетов, используется код **TAF** (Terminal Aerodrome Forecast). Этот формат позволяет аэропортам передавать прогнозы погоды на ближайшие часы для планирования взлетов и посадок.

**2. Всемирная метеорологическая организация (ВМО)**

**Всемирная метеорологическая организация (ВМО)** является ведущим международным органом, который разрабатывает стандарты для сбора, передачи, анализа и обмена метеорологическими данными. ВМО координирует действия метеорологических служб по всему миру, что особенно важно для обеспечения глобального мониторинга погодных условий и изменения климата.

**Основные стандарты ВМО:**

* **Код METAR** — стандарт для метеорологических сводок, используемых для регулярных наблюдений за погодными условиями в аэропортах. Данные METAR передаются каждые 30 или 60 минут и включают такие параметры, как температура, атмосферное давление, видимость, скорость и направление ветра, облачность и наличие опасных погодных явлений (грозы, туманы и т. д.).
* **Код SYNOP** — международный стандарт для метеорологических наблюдений, проводимых на наземных метеостанциях. SYNOP используется для передачи широкого спектра данных: температура, влажность, атмосферное давление, осадки, направление и скорость ветра, видимость и другие параметры. Этот формат широко используется для сбора данных с наземных станций по всему миру и является основой для анализа и прогноза погодных условий.

**3. Форматы данных для хранения и передачи метеорологических данных**

Для того чтобы обеспечить стандартизацию передачи данных и удобство их обработки, в метеорологии применяются специализированные форматы для хранения больших объемов данных, собранных в результате наблюдений или численного моделирования.

**Основные форматы данных:**

* **GRIB (Gridded Binary)** — это бинарный формат данных, широко используемый для хранения и передачи прогнозов погоды. Формат GRIB позволяет эффективно сжимать данные, сохраняя при этом их точность, что делает его удобным для обмена большими объемами информации, полученной с метеостанций и в ходе численного моделирования. Он используется для передачи данных о температуре, давлении, скорости ветра и других параметрах в виде регулярных сеток, что особенно важно для прогностических моделей погоды.
* **BUFR (Binary Universal Form for the Representation of meteorological data)** — это универсальный бинарный формат для представления метеорологических данных, разработанный ВМО. BUFR поддерживает различные типы данных: от наблюдений с наземных метеостанций до данных спутникового наблюдения и радиозондов. BUFR позволяет объединять в одном формате различные виды данных, что облегчает их последующую обработку и анализ.

**4. Национальные стандарты**

Каждая страна, наряду с международными стандартами, может внедрять свои национальные стандарты для учета особенностей климата, географического положения и локальных потребностей. Эти стандарты зачастую базируются на международных, но могут быть адаптированы для более точного учета специфических условий страны.

**Примеры национальных стандартов:**

* **Россия** (Росгидромет): в России действует ряд стандартов и нормативов для оформления метеорологических данных, многие из которых основаны на рекомендациях ВМО. Например, Росгидромет использует коды METAR и SYNOP для передачи данных, но могут быть разработаны и дополнительные формы отчетности для внутренних нужд. В России также действуют национальные стандарты по подготовке метеорологических прогнозов для авиации и мореплавания.
* **США** (NOAA): в США действует **Национальное управление по океаническим и атмосферным исследованиям (NOAA)**, которое также использует международные стандарты, такие как METAR, SYNOP, GRIB и BUFR, но в некоторых случаях применяет собственные локальные расширения и адаптации для национальных нужд. Например, в США могут использоваться специфические форматы для передачи данных военной авиации и внутреннего прогнозирования погодных условий.

# Существующие информационные системы и их функции

Некоторые популярные системы:

* **GFS (Global Forecast System)**: глобальная модель для прогноза погоды, разработанная Национальной метеорологической службой США. Обрабатывает данные со всего мира, создавая прогнозы погоды на несколько дней.
* **ECMWF (European Centre for Medium-Range Weather Forecasts)**: система, предоставляющая точные прогнозы погоды на средний срок (до 15 дней).
* **WMO Information System (WIS)**: глобальная система обмена метеорологической информацией, поддерживаемая ВМО, используется для передачи данных между метеорологическими службами разных стран.
* **Local Weather Systems (например, Weather Underground)**: системы для наблюдений на локальных метеостанциях с передачей данных о погоде через интернет.

**Решаемые задачи:**

* Автоматизация сбора данных с метеостанций, спутников и других сенсоров;
* Автоматизация прогнозов и расчетов на основе данных;
* Генерация предупреждений об опасных явлениях;
* Интеграция с другими системами для обеспечения глобального обмена данными.

# Предложения по автоматизации выбранной предметной области

В целях улучшения процессов учета наблюдений за погодой можно предложить следующие направления автоматизации:

* **Автоматизация сбора данных**: подключение датчиков и метеостанций для автоматического измерения показателей погоды и их отправки в централизованную базу данных в реальном времени.
* **Автоматизированный анализ данных**: внедрение машинного обучения и анализа больших данных для улучшения точности прогнозов погоды, выявления паттернов, связанных с изменением климата.
* **Разработка интерфейсов для пользователей**: создание веб-платформы или мобильного приложения для визуализации данных в реальном времени, где будут представлены графики, карты и отчеты о погоде. Также можно реализовать системы уведомлений о погодных катаклизмах.
* **Интеграция с глобальными системами обмена данными**: чтобы обеспечивать обмен данными с международными системами, такими как WIS или ECMWF, для улучшения качества прогноза на основе данных из разных регионов.
* **Автоматизация отчетности**: генерация стандартных отчетов (в формате METAR, SYNOP) и отправка их в соответствующие организации для синхронизации с глобальными данными.

# Заключение

Анализ предметной области учета наблюдений за погодой показывает, что обработка информации в данной сфере является сложным и многогранным процессом, требующим использования как международных, так и национальных стандартов. Основные задачи включают сбор, обработку, хранение и передачу метеорологических данных, которые затем используются для прогнозирования погодных условий, предупреждения о неблагоприятных явлениях и обеспечения безопасности в таких отраслях, как авиация, мореплавание и сельское хозяйство. Эти задачи требуют высокой точности и оперативности, что подчеркивает необходимость стандартизации процессов на международном уровне.

Существующие стандарты, такие как METAR, SYNOP и форматы данных GRIB и BUFR, играют ключевую роль в обеспечении корректности и совместимости передаваемой информации. Международные организации, такие как Всемирная метеорологическая организация (ВМО) и Международная организация гражданской авиации (ICAO), разработали ряд универсальных кодов и форматов, которые обеспечивают унификацию данных о погоде, позволяя странам и организациям эффективно обмениваться информацией. Эти стандарты обеспечивают основу для создания и развития информационных систем в области метеорологии.

Существующие информационные системы выполняют множество функций по сбору, обработке и представлению данных о погоде. Эти системы включают как локальные решения для обработки национальных данных, так и международные инструменты для обмена информацией. Они поддерживают прогнозирование погоды, мониторинг климатических изменений и предупреждения о катастрофических погодных явлениях, что позволяет эффективно реагировать на изменения погодных условий и минимизировать риски.

Автоматизация процессов в этой области является важным направлением для повышения эффективности и скорости обработки данных. Использование облачных технологий, интеграция искусственного интеллекта для анализа данных и прогнозов, а также развитие мобильных приложений и веб-интерфейсов для пользователей позволят сделать систему более адаптивной, быстрой и удобной. Автоматизация рутинных операций, таких как сбор данных с метеостанций и их анализ, позволит сократить количество ошибок и улучшить качество прогнозов, что особенно важно в условиях изменения климата и увеличения количества экстремальных погодных явлений.

# Спецификация к ПО

Система учета наблюдений за погодой представляет собой веб-платформу, предназначенную для предоставления пользователям актуальных данных о погодных условиях в реальном времени, прогнозов на будущее, а также анализа исторических данных. Основная цель системы — автоматизация процесса сбора, хранения и анализа метеорологических данных, что делает её полезной как для широкой аудитории, так и для профессиональных пользователей, таких как метеорологи, фермеры и специалисты по планированию мероприятий.

**Основные функции системы:**

1. **Получение и отображение текущих погодных данных**: Система собирает данные о температуре, влажности, давлении, ветре и осадках из внешних источников (например, API OpenWeatherMap) и отображает их в удобном для восприятия виде.
2. **Прогноз погоды**: Предоставление прогнозов на ближайшие несколько дней с использованием данных от внешних метеорологических служб.
3. **Анализ исторических данных**: Возможность доступа к данным о погоде за прошедшие периоды для анализа изменений и трендов.

**Характеристики пользователей:**

Система ориентирована на несколько групп пользователей:

* **Обычные пользователи**: Люди, интересующиеся погодой в своем регионе.
* **Профессиональные пользователи**: Специалисты, которым необходимы точные данные для анализа и принятия решений.
* **Разработчики и интеграторы**: Пользователи, интегрирующие систему с другими приложениями через API.

**Ограничения и стандарты:**

* Система зависит от доступности внешних метеорологических API, что может влиять на точность данных и их обновление.
* Используются международные стандарты для обмена метеорологической информацией (METAR, SYNOP, GRIB, BUFR).
* Система соответствует стандартам веб-доступности (WCAG 2.1) для обеспечения удобства использования для всех категорий пользователей.

**Требования к данным:**

* Логическая модель данных включает объекты, такие как пользователи, метеорологические данные, исторические данные и отчеты.
* Данные собираются через API внешних источников, хранятся в базе данных и обеспечиваются меры по их целостности и безопасности.
* Пользователи могут экспортировать данные в формате CSV или PDF для дальнейшего анализа.

**Требования к внешним интерфейсам:**

* **Пользовательский интерфейс**: Интуитивно понятный и адаптивный дизайн, поддерживающий различные устройства (настольные компьютеры, планшеты, смартфоны).
* **Интерфейсы ПО**: Взаимодействие с внешними API, базами данных и библиотеками для визуализации данных.
* **Коммуникационные интерфейсы**: Поддержка электронной почты, браузеров и сетевых протоколов (HTTP/HTTPS).

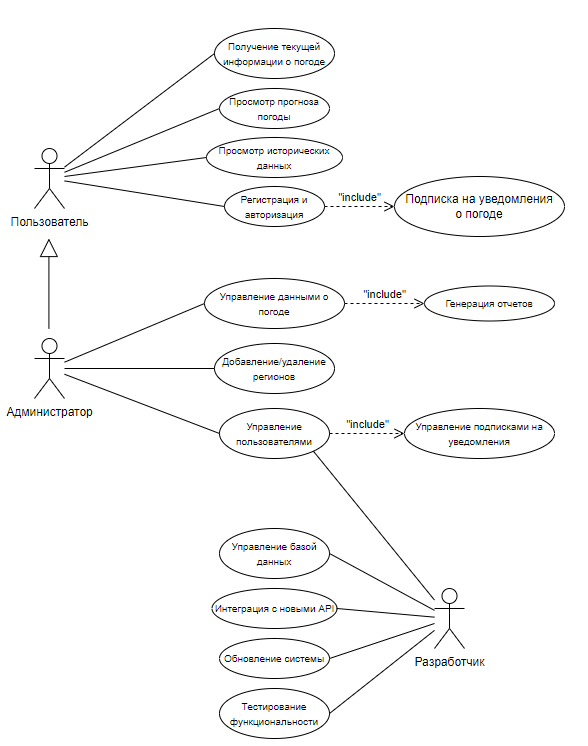
**Атрибуты качества:**

* **Удобство использования**: Простота навигации, адаптивный дизайн и поддержка специальных возможностей.
* **Производительность**: Высокая скорость отклика и масштабируемость системы.
* **Безопасность**: Защита данных, аутентификация и авторизация, резервное копирование и защита от атак.

Данная спецификация обеспечивает четкое понимание требований к системе, что позволяет всем заинтересованным сторонам (разработчикам, тестировщикам, пользователям) иметь единое представление о целях и задачах проекта. Система направлена на создание удобной, надежной и интуитивно понятной платформы для мониторинга и анализа погодных условий.

# Use-case диаграмма по учету наблюдений за погодой

Цель данного отчёта – описать процесс создания диаграммы use-case для системы учёта наблюдений за погодой. Эти диаграммы помогают визуализировать взаимодействие различных пользователей с системой и процессы, связанные с получением, обработкой и представлением метеорологических данных. Основные группы пользователей включают обычных пользователей, администраторов и разработчиков, которые взаимодействуют с системой для доступа к погодным данным, управления её функционированием и разработки новых функций.



Для Use-Case диаграммы проекта, который посвящен учету наблюдений за погодой и предоставлению прогноза, важно учитывать ключевые функциональные возможности системы и взаимодействие с пользователями. Диаграмма будет отражать действия, которые пользователи могут выполнять, а также ключевые компоненты системы, такие как сбор данных, отображение прогноза и т.д.

**Актёры:**

**Пользователь** — основной посетитель сайта, который получает информацию о погоде.

**Администратор** — пользователь с расширенными правами, который управляет системой, добавляет новые регионы, обновляет данные и т.д.

**Разработчик** — отвечает за техническое обслуживание сайта, тестирование и обновление кода.

**Сценарии использования (Use Cases):**

1. ***Получение текущей информации о погоде***

Пользователь выбирает город/регион и получает актуальную информацию о погоде (температура, влажность, ветер и т.д.).

Отношение: Ассоциация с пользователем.

1. ***Просмотр прогноза погоды***

Пользователь запрашивает прогноз погоды на ближайшие дни (например, на 3-5 дней вперед).

Отношение: Ассоциация с пользователем.

***3.* *Просмотр исторических данных***

Пользователь может просмотреть погодные условия в прошлом (например, для планирования или анализа).

Отношение: Ассоциация с пользователем.

***4. Регистрация и авторизация***

Пользователь регистрируется на сайте для сохранения настроек или подписки на уведомления о погоде.

Отношение: Ассоциация, расширяет сценарии просмотра (регистрация предоставляет дополнительные функции).

***4.1. Подписка на уведомления о погоде***

Пользователь подписывается на получение ежедневных уведомлений о погоде через электронную почту.

Отношение: Включение в сценарий регистрации (доступно только после регистрации).

***5. Управление данными о погоде***

Администратор может проверять поступающие данные, корректировать их или вручную добавлять новые записи о погодных условиях.

Отношение: Ассоциация с администратором (прямое действие администратора).

***5.1. Генерация отчетов***

Администратор создает отчеты о погодных условиях (например, для анализа или для профессиональных пользователей).

Отношение: Расширяет сценарий управления данными о погоде.

***6. Добавление/удаление регионов***

Администратор может управлять регионами в базе данных (добавлять новые города или удалять старые).

Отношение: Ассоциация.

***7. Управление пользователями***

Администратор (разработчик) управляет зарегистрированными пользователями, может блокировать доступ или изменять права.

Отношение: Ассоциация с администратором.

***8. Управление подписками на уведомления***

Администратор может управлять системой рассылки уведомлений, включая изменение частоты отправки и содержания.

Отношение: Расширяет сценарий «Управление пользователями» (часть управления сервисами для пользователей).

***9. Управление базой данных***

Разработчик выполняет поддержку базы данных, следит за её производительностью и целостностью.

Отношение: Ассоциация.

***10. Интеграция с новыми API***

Разработчик подключает новые метеорологические API для улучшения точности прогноза или добавления новых источников данных.

Отношение: Ассоциация, связана с обновлением системы.

***11. Обновление системы***

Разработчик обновляет код, исправляет баги, внедряет новые функции (например, добавление новых API для погоды).

Отношение: Ассоциация, также может расширять сценарий тестирования (перед обновлением необходимо проводить тесты).

***12. Тестирование функциональности***

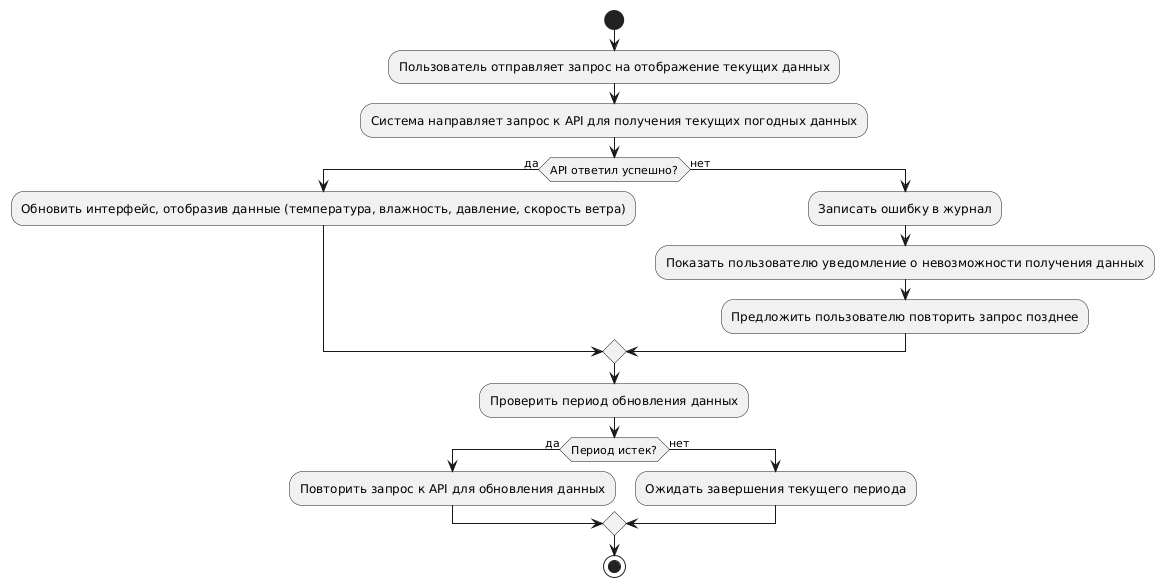
Разработчик проверяет систему на наличие ошибок, проводит юнит- и интеграционные тесты.

Отношение: Ассоциация с разработчиком.

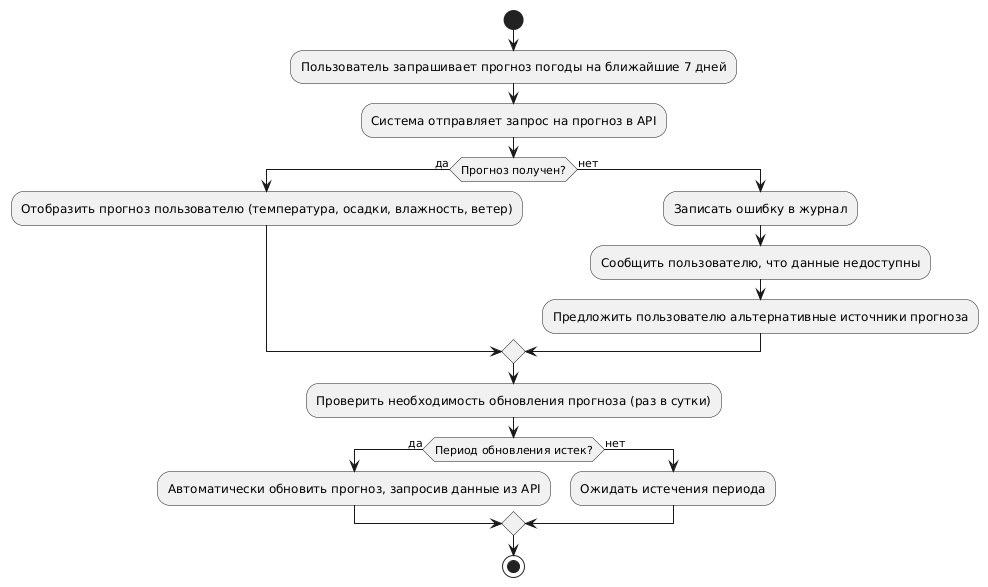
В заключение, диаграмма use case, представленная в данном отчёте, наглядно демонстрирует ключевые сценарии взаимодействия пользователей с системой учёта наблюдений за погодой. Они помогают лучше понять, как различные группы пользователей — от обычных посетителей сайта до администраторов и разработчиков — могут использовать функционал системы для получения данных, их обработки и управления системой. Такой подход обеспечивает чёткое понимание архитектуры системы и её процессов, что способствует эффективному проектированию и дальнейшему развитию проекта. Диаграммы являются важным инструментом для визуализации требований и упрощают процесс коммуникации между разработчиками и другими заинтересованными сторонами проекта.

# Диаграммы активности

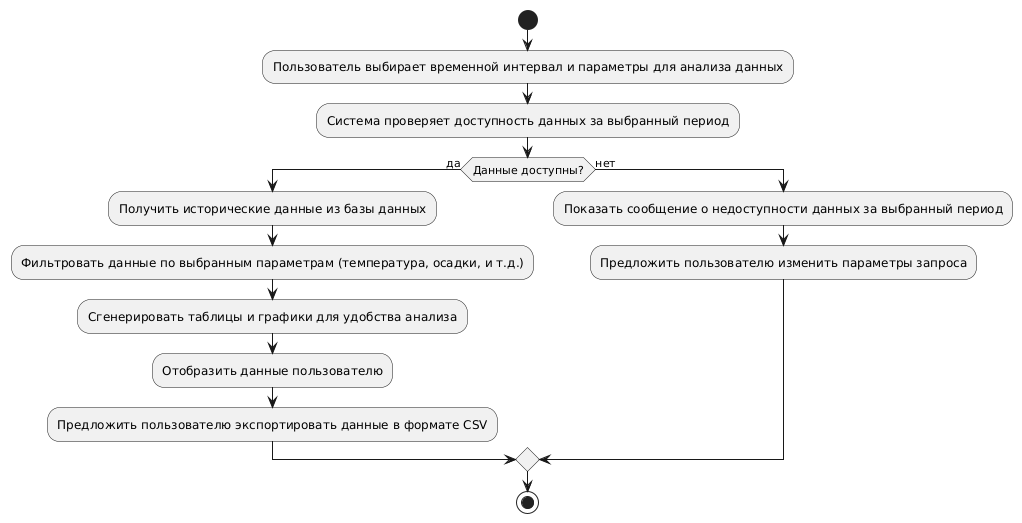
Данный отчёт описывает и анализирует ключевые процессы системы учёта наблюдений за погодой, которая предназначена для предоставления пользователям данных о текущих погодных условиях, прогноза на ближайшие дни, анализа исторических данных и генерации отчётов. Система предназначена для использования как широкой аудиторией, так и профессиональными пользователями, нуждающимися в точных метеорологических данных. В отчёте описаны цели, задачи и функции каждого процесса, а также выводы и рекомендации для улучшения системы. Диаграммы разработаны с использованием PlantUML и включают в себя основные шаги, условия и детали, которые помогают понять, как пользователи взаимодействуют с системой.

*****Получение и отображение текущих погодных данных*

Процесс начинается с получения запроса от пользователя на отображение текущих погодных данных. Система запрашивает данные через API метеорологических служб. В случае успешного получения данных они отображаются пользователю в удобном формате, включая температуру, влажность, атмосферное давление и скорость ветра. Если данные недоступны, система уведомляет пользователя о проблеме и предлагает повторить запрос позже.

***Прогноз погоды*

Процесс включает запрос от пользователя на прогноз погоды на ближайшие 7 дней. Система направляет запрос к внешнему API для получения прогноза. Если данные получены, они отображаются с информацией о температуре, осадках, влажности и ветре. При отсутствии данных система уведомляет пользователя и предлагает альтернативные способы получения прогноза.

***Анализ исторических данных*

Пользователь запрашивает доступ к историческим погодным данным за определённый временной интервал (например, за последние 5 лет). Система фильтрует данные по выбранным параметрам и отображает их в виде таблиц и графиков, также поддерживается экспорт в формате CSV.

Система учёта наблюдений за погодой предоставляет пользователям доступ к актуальной информации о погоде, возможности просмотра прогнозов, анализа исторических данных и создания отчётов. Основные процессы, описанные в отчёте, охватывают ключевые функции системы, направленные на удовлетворение потребностей как рядовых пользователей, так и профессионалов. В отчёте также даны рекомендации для дальнейшего улучшения системы: оптимизация производительности, поддержка безопасности данных и улучшение пользовательского интерфейса. Внедрение этих рекомендаций позволит повысить качество предоставляемых данных и удобство использования системы.

# Диаграммы классов

В этом отчёте представлен анализ диаграммы классов для системы учёта наблюдений за погодой. Диаграмма описывает структуру и взаимодействие основных сущностей, их атрибутов и методов, а также показывает, как данные перемещаются между различными компонентами системы. Система предназначена для предоставления данных о текущей погоде, прогнозов, анализа исторических данных и генерации отчётов для пользователей. Основными компонентами системы являются классы, отвечающие за пользователей, погодные данные, прогнозы, исторические данные, отчёты и взаимодействие с внешними API.

# Диаграмма классов

Описание диаграммы классов для системы управления конференциями и мероприятиями содержит основные сущности и их взаимодействие.

## Описание классов

**Класс Пользователь**

Класс Пользователь представляет собой модель, которая хранит информацию о пользователях системы и отвечает за их взаимодействие с системой. Этот класс содержит атрибуты, такие как:

* ID — уникальный идентификатор пользователя;
* имя — имя пользователя;
* почта — электронная почта пользователя;
* регионИнтересов — предпочтительный регион для показа данных о погоде;
* активныеОтчёты — список созданных пользователем отчётов.

Основные методы включают:

* регистрация() и вход() — для управления доступом пользователя в систему.
* создатьОтчёт() — создание отчёта на основе выбранных параметров.
* получитьИсторическиеДанные() и получитьПрогноз() — получение данных и прогноза для региона интересов пользователя.

**Класс ПогодныеДанные**

ПогодныеДанные представляет текущие погодные условия и является центральным классом для работы с метеорологическими данными. Атрибуты включают:

* температура, влажность, давление, скоростьВетра, осадки — параметры текущей погоды;
* регион — указание местоположения;
* времяНаблюдения — время записи данных.

Методы:

* обновитьДанные() — обновление погодных данных из API.
* получитьДанныеНаТекущийМомент() — получение текущих данных для отображения.

**Класс Прогноз**

Прогноз хранит предсказания погоды на основе временного интервала. Основные атрибуты:

* датаНачала и датаКонца — интервал времени для прогноза;
* данныеПрогноза — данные о погоде по дням.

Методы:

* обновитьПрогноз() — обновление прогноза из внешнего источника.
* получитьПрогнозНаДень() — получение прогноза на выбранную дату.

**Класс ИсторическиеДанные**

ИсторическиеДанные отвечает за хранение погодных наблюдений за прошедшие периоды и анализ данных. Атрибуты:

* период — временной диапазон;
* регион — местоположение данных;
* данные — список объектов ПогодныеДанные для указанного периода.

Методы:

* получитьДанныеЗаПериод() — извлечение данных для анализа.
* фильтроватьПоПараметрам() — фильтрация данных по выбранным параметрам.
* экспортироватьВCSV() — экспорт данных для внешнего анализа.

**Класс Отчёт**

Отчёт позволяет пользователям создавать настраиваемые отчёты о погоде на основе исторических и текущих данных. Атрибуты включают:

* ID — уникальный идентификатор отчёта;
* создатель — связь с пользователем, создавшим отчёт;
* временнойИнтервал — период данных;
* параметры — параметры для отчёта;
* данные — данные о погоде для отчёта;
* формат — формат файла отчёта (например, PDF или CSV).

Методы:

* сгенерировать() — создание отчёта на основе выбранных данных.
* экспортировать() — экспорт отчёта.
* отправитьПоEmail() — отправка отчёта на электронную почту.

**Класс WeatherAPI**

WeatherAPI реализует взаимодействие с внешними метеорологическими сервисами. Атрибуты:

* url — адрес API.
* apiKey — ключ для доступа.

Методы:

* получитьТекущиеДанные(), получитьПрогноз() и получитьИсторическиеДанные() — запросы для получения данных.

**Класс Уведомление**

Уведомление отвечает за отправку уведомлений пользователям. Атрибуты:

* тип — тип уведомления (например, предупреждение, информация);
* содержание — текст уведомления;
* датаОтправки — дата и время отправки.

Методы:

* создатьУведомление() — создание уведомления.
* отправить() — отправка уведомления пользователю.

**Класс НастройкиПользователя**

Этот класс хранит персональные настройки, такие как предпочтительный язык, формат даты и единицы измерения. Методы включают:

* обновитьНастройки() — изменение настроек пользователя.
* получитьНастройки() — получение текущих настроек.

## Описание связей

* Пользователь и Отчёт: Каждый пользователь может создавать и хранить множество отчётов. Связь ассоциации 1:0..\* показывает, что каждый пользователь может иметь несколько отчётов.
* Пользователь и НастройкиПользователя: Один пользователь связан с единственным набором настроек, который можно персонализировать.
* Отчёт и ИсторическиеДанные / ПогодныеДанные: Отчёты могут включать данные о погоде за прошедшие периоды или текущие данные. Это позволяет пользователям анализировать как историческую информацию, так и текущие погодные условия.
* ИсторическиеДанные и ПогодныеДанные: Агрегация между ИсторическиеДанные и ПогодныеДанные показывает, что исторические данные состоят из отдельных записей о погоде.
* Прогноз и ПогодныеДанные: Композиция между Прогноз и ПогодныеДанные показывает, что каждый прогноз содержит данные на определённые даты.
* WeatherAPI и ПогодныеДанные / Прогноз / ИсторическиеДанные: WeatherAPI получает данные из внешних источников для использования в классах ПогодныеДанные, Прогноз и ИсторическиеДанные.
* Уведомление и Пользователь: Уведомление отправляется пользователю для информирования о важных событиях.

Диаграмма классов для системы учёта наблюдений за погодой охватывает все основные элементы, необходимые для функционирования системы, и отражает продуманные взаимосвязи между ними. Такая структура позволяет пользователям получать актуальные погодные данные, анализировать историческую информацию и создавать отчёты. Реализованные связи и методы делают систему гибкой и удобной, а также расширяемой для поддержки различных пользовательских требований и улучшения взаимодействия с метеорологическими сервисами.

# Кодирование проекта

Проект "Погода.бай" представляет собой веб-приложение для получения актуальной погоды, прогноза на несколько дней и анализа исторических данных. Приложение разработано с использованием HTML, CSS и JavaScript, а также интегрируется с внешними API для получения метеорологических данных. Ниже приведено подробное описание кодирования проекта.

**1. Структура проекта**

Проект состоит из двух основных HTML-страниц, стилей (CSS) и скриптов (JavaScript).

**HTML-файлы**

1. **index.html**  
   Это главная страница, на которой пользователь вводит название города и переходит на страницу с деталями погоды.
   * **Элементы**:
     + Поле ввода города с автозаполнением.
     + Кнопка для перехода на страницу с погодой.
     + Сообщение об ошибке (если город не введен).
2. **weather.html**  
   Страница с деталями погоды, где пользователь может выбрать категорию (текущая погода, прогноз, исторические данные).
   * **Элементы**:
     + Боковая панель с кнопками для выбора категории.
     + Основной контент, который динамически обновляется в зависимости от выбранной категории.

**CSS-файл**

Файл style.css содержит стили для всех элементов интерфейса:

* **Общие стили**:
  + Шрифт Rubik, подключенный через Google Fonts.
  + Центральное расположение контента на странице.
  + Адаптивный дизайн для мобильных устройств.
* **Стили для кнопок**:
  + Кнопки с градиентом и эффектом при наведении.
* **Стили для контейнеров**:
  + Контейнеры с тенью и закругленными углами для улучшения визуального восприятия.

**JavaScript-файл**

Файл script.js содержит логику приложения:

* **Функции для работы с API**:
  + Получение текущей погоды.
  + Получение прогноза погоды на несколько дней.
  + (Планируется) Получение исторических данных.
* **Функции для управления интерфейсом**:
  + Переход между страницами.
  + Обновление контента на основе выбранной категории.

**2. Основные функции**

**1. Переход на страницу с погодой**

Функция openWeatherPage() вызывается при нажатии кнопки "Получить погоду".

* Проверяет, введен ли город.
* Сохраняет выбранный город в localStorage.
* Перенаправляет пользователя на страницу weather.html.

**2. Отображение текущей погоды**

Функция showCurrentWeather() вызывается при выборе категории "Текущая погода".

* Использует API OpenWeatherMap для получения данных о текущей погоде.
* Отображает температуру в выбранном городе.

**3. Отображение прогноза погоды**

Функция showWeatherForecast() вызывается при выборе категории "Прогноз погоды".

* Предоставляет пользователю выбор количества дней для прогноза.
* Использует API OpenWeatherMap для получения прогноза.
* Отображает прогноз в виде списка с датой, температурой, описанием погоды и скоростью ветра.

**4. Отображение исторических данных**

Функция showHistoricalData() вызывается при выборе категории "Исторические данные".

* В текущей версии отображает заглушку с сообщением о том, что функционал будет разработан позднее.
* (Планируется) Интеграция с API для получения исторических данных.

**5. Изменение города**

Функция changeCity() позволяет пользователю изменить выбранный город.

* Сохраняет новый город в localStorage.
* Обновляет данные о погоде для нового города.

**3. Интеграция с API**

Приложение использует API OpenWeatherMap для получения данных о погоде.

* **Текущая погода**:
  + URL: https://api.weatherapi.com/v1/current.json
  + Параметры: ключ API, название города.
* **Прогноз погоды**:
  + URL: https://api.openweathermap.org/data/2.5/forecast
  + Параметры: координаты города, количество дней, ключ API.
* **Исторические данные**:
  + (Планируется) Использование API для получения данных за прошлые даты.

**4. Особенности реализации**

* **Использование localStorage**:
  + Сохранение выбранного города для удобства пользователя.
* **Адаптивный дизайн**:
  + Поддержка различных устройств (настольные компьютеры, планшеты, смартфоны).
* **Интуитивно понятный интерфейс**:
  + Простота навигации и удобство использования.

**5. Планируемые улучшения**

* Реализация функционала для получения исторических данных.
* Добавление возможности экспорта данных в формате CSV или PDF.
* Улучшение визуализации данных с использованием графиков и диаграмм.

**6. Заключение**

Проект "Погода.бай" демонстрирует базовый функционал для получения и отображения погодных данных. Он успешно интегрируется с внешними API, предоставляет удобный интерфейс и поддерживает основные функции, такие как текущая погода и прогноз. В будущем планируется расширение функционала, включая поддержку исторических данных и улучшение пользовательского опыта.

Также в последствии было реализовано 6 тестов для проверки корректной работы функционала.

# Диаграмма развёртывания

Диаграмма развертывания (Deployment Diagram) предназначена для отображения архитектуры системы, включая взаимодействие между её компонентами и внешними системами. В данном случае диаграмма описывает архитектуру системы прогноза погоды, которая включает клиентский узел (веб-браузер), серверный узел (веб-сервер) и внешние API (OpenWeatherMap и WeatherAPI).

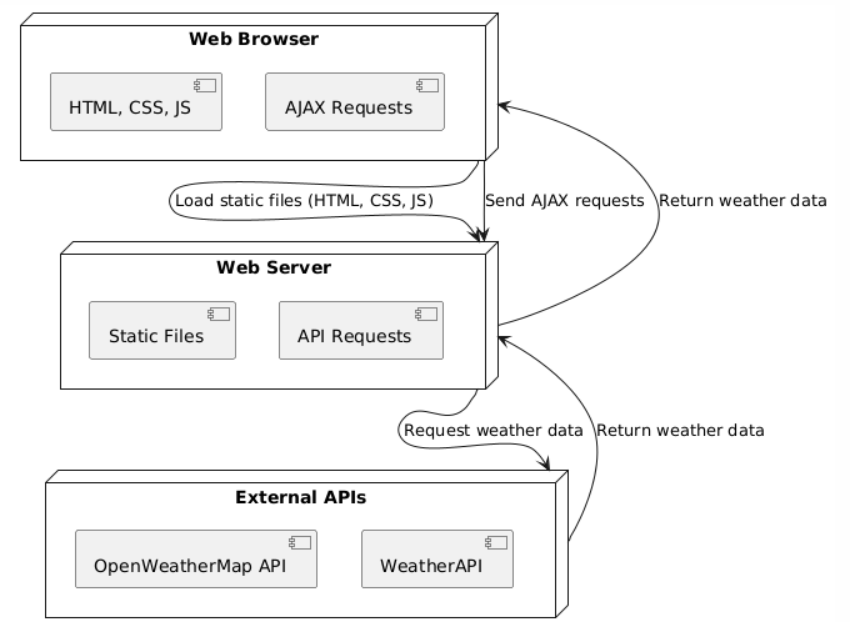


Диаграмма развертывания (Deployment Diagram) предназначена для отображения архитектуры системы, включая взаимодействие между её компонентами и внешними системами. В данном случае диаграмма описывает архитектуру системы прогноза погоды, которая включает клиентский узел (веб-браузер), серверный узел (веб-сервер) и внешние API (OpenWeatherMap и WeatherAPI).

**2. Компоненты системы**

**2.1. Web Browser (Клиентский узел)**

* **Описание**:  
  Клиентский узел представляет собой веб-браузер пользователя, через который осуществляется взаимодействие с системой.
* **Функции**:
  + Загрузка статических файлов (HTML, CSS, JS) с веб-сервера.
  + Отправка AJAX-запросов для получения данных о погоде.
* **Роль в системе**:  
  Предоставляет пользовательский интерфейс для ввода данных (например, названия города) и отображения результатов (текущая погода, прогноз).

**2.2. Web Server (Серверный узел)**

* **Описание**:  
  Серверный узел представляет собой веб-сервер, который хранит статические файлы системы (HTML, CSS, JS) и обрабатывает запросы от клиента.
* **Функции**:
  + Хранение и предоставление статических файлов (HTML, CSS, JS).
  + Перенаправление запросов к внешним API для получения данных о погоде.
  + Возврат данных о погоде клиенту.
* **Роль в системе**:  
  Выступает посредником между клиентом и внешними API, обеспечивая обработку запросов и передачу данных.

**2.3. External APIs (Внешние API)**

* **Описание**:  
  Внешние API представляют собой сервисы, предоставляющие данные о погоде.
* **Функции**:
  + OpenWeatherMap API: предоставляет данные о текущей погоде, прогнозе и исторических данных.
  + WeatherAPI: альтернативный источник данных о погоде.
* **Роль в системе**:  
  Являются основным источником данных для системы прогноза погоды.

**3. Взаимодействие между компонентами**

**3.1. Web Browser → Web Server**

* **Описание**:  
  Веб-браузер отправляет запросы на загрузку статических файлов (HTML, CSS, JS) и AJAX-запросы для получения данных о погоде.
* **Цель**:  
  Загрузка пользовательского интерфейса и получение актуальных данных о погоде.

**3.2. Web Server → External APIs**

* **Описание**:  
  Веб-сервер перенаправляет запросы к внешним API для получения данных о погоде.
* **Цель**:  
  Получение актуальных данных о погоде от внешних источников.

**3.3. External APIs → Web Server**

* **Описание**:  
  Внешние API возвращают данные о погоде веб-серверу.
* **Цель**:  
  Передача данных о погоде для дальнейшей обработки и отправки клиенту.

**3.4. Web Server → Web Browser**

* **Описание**:  
  Веб-сервер возвращает данные о погоде веб-браузеру.
* **Цель**:  
  Отображение данных о погоде пользователю.

**4. Архитектура системы**

Система построена на основе клиент-серверной архитектуры, где:

* **Клиентская часть (Web Browser)** отвечает за взаимодействие с пользователем.
* **Серверная часть (Web Server)** обрабатывает запросы клиента и взаимодействует с внешними API.
* **Внешние API (External APIs)** предоставляют данные о погоде.

**5. Преимущества архитектуры**

* **Простота и масштабируемость**:  
  Клиент-серверная архитектура позволяет легко масштабировать систему, добавляя новые серверы или внешние API.
* **Отделение логики**:  
  Веб-сервер выполняет роль посредника, что упрощает поддержку и развитие системы.
* **Использование внешних API**:  
  Использование внешних API позволяет системе быстро получать актуальные данные о погоде без необходимости самостоятельно собирать и обрабатывать данные.

Диаграмма развертывания системы прогноза погоды наглядно демонстрирует взаимодействие между клиентом, сервером и внешними API. Такая архитектура обеспечивает простоту, масштабируемость и удобство использования системы. Внешние API являются ключевым элементом системы, предоставляя данные о погоде, которые затем обрабатываются и отображаются пользователю через веб-интерфейс.