Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа №5

По дисциплине “Программирование программного обеспечения”

По теме “ Объектно-ориентированное моделирование. Структурные диаграммы UML.”

Цель работы: “Изучение методологии объектно-ориентированного моделирования средствами UML”

Студент: Глухова Д. В.

ФИТ 3 курс 2 группа

Преподаватель: Якубенко К.Д.

1. **Теоретические вопросы подготовки к лабораторной работе**
2. Перечислите структурные диаграммы, которые входят в UML 2.0.

Диаграмма классов, диаграмма компонентов, диаграмма композитной структуры, диаграмма кооперации, диаграмма развёртывания, диаграмма объектов, диаграмма пакетов.

1. Укажите назначение структурных диаграммы.

Диаграммы для визуализации, специфицирования, конструирования и документирования статических аспектов системы

1. Опишите нотации, которые используются для построения Classes диаграмм.

Класс — это описание набора объектов с одинаковыми атрибутами, операциями.

Атрибуты (свойства) описывают содержимое класса.

Операции (методы) представляют собой некоторый сервис, предоставляемый каждым экземпляром класса.

Зависимость – семантически представляет собой связь между двумя элементами модели, в которой изменение одного элемента (независимого) может привести к изменению семантики другого элемента (зависимого).

Ассоциация – это структурная связь между элементами модели, которая описывает набор связей, существующих между объектами.

Агрегация – особая разновидность ассоциации, представляющая структурную связь целого с его частями. Как тип ассоциации, агрегация может быть именованной. Одно отношение агрегации не может включать более двух классов (контейнер и содержимое).

Наследование – связь, в которой специализированный элемент (потомок) строится по спецификациям обобщенного элемента (родителя).

Реализация – это семантическая связь между классами, когда один из них (поставщик) определяет соглашение, которого второй (клиент) обязан придерживаться.

1. Для чего применяются расширения диаграмм UML?

Для уточнения или специализации общих элементов метамодели при разработке конкретных элементов метамодели при разработке конкретных моделей UML

1. Что означают понятия «стереотип» и «тегированное значение» в контексте расширенных диаграмм?

Стереотипы являются одним из трех типов механизмов расширяемости в унифицированном языке моделирования. Они позволяют проектировщикам расширять словарь UML для создания новых элементов моделирования, получаемых из существующих, но имеющих определенные свойства, которые подходят для конкретной проблемы предметной области или для другого специализированного использования. Тегированное значение – часть стандартного UML-элемента, содержащая о нём дополнительную информацию.

1. **Постановка задачи (описание функциональных требований)**

Функциональные требования к системе можно разделить на требования для различных ролей приложения – учителя, ученика, родителя и администратора образовательного учреждения.

Функционал для учителя:

* Регистрация и авторизация: возможность создания учетной записи и входа в систему.
* Управление расписанием: создание, редактирование и просмотр расписания уроков.
* Ведение успеваемости: выставление оценок и комментариев к ним для каждого ученика.
* Создание и редактирование заданий: возможность добавления домашних заданий и материалов для уроков.
* Общение с учениками и родителями: инструменты для отправки уведомлений и сообщений.
* Просмотр статистики успеваемости: анализ успеваемости и активности учеников через отчеты.
* Отмена или изменение заданий: возможность редактирования или отмены ранее установленных заданий.

Функционал для ученика:

* Регистрация и авторизация: создание личного кабинета и вход в систему.
* Просмотр расписания: доступ к расписанию уроков и информации о предстоящих занятиях.
* Просмотр заданий: доступ к домашним заданиям, срокам их выполнения и комментариям учителей.
* Обратная связь: возможность оставлять комментарии и задавать вопросы учителям.
* Просмотр успеваемости: возможность отслеживания собственных оценок и прогресса.

Функционал для администратора:

* Просмотр списка всех пользователей: доступ к информации о всех учителях, учениках и родителях.
* Управление учетными записями: добавление, изменение и удаление пользователей.
* Просмотр и управление расписанием: возможность редактирования расписаний и учебных планов.
* Подтверждение регистрации новых пользователей: проверка и подтверждение заявок на регистрацию учителей и родителей.

1. **Описание программных средств**

Для построения диаграмм IDEF0 в рамках нашего проекта использовался веб-ресурс Draw.io, разработанный компанией JGraph Ltd. Этот инструмент предназначен для создания разнообразных диаграмм и визуальных представлений информации. Адрес веб-ресурса – https://www.drawio.com. Draw.io доступен на всех платформах, которые имеют веб-браузер и подключение к Интернету, что делает его универсальным и удобным для пользователей.

Draw.io предлагает широкий функционал, который позволяет пользователям создавать профессиональные графические диаграммы с минимальными затратами времени и усилий. В числе доступных возможностей — построение графиков, смысловых карт, UML-диаграмм, диаграмм Венна, а также Agile и Kanban досок. Эти инструменты особенно полезны в рамках проектного управления и разработки программного обеспечения, так как они помогают визуализировать процессы, повышая их наглядность и понятность.

Кроме того, Draw.io поддерживает создание диаграмм мозговых штурмов, архитектурных диаграмм технических систем и других типов визуальных представлений, необходимых для анализа и планирования. Интуитивно понятный интерфейс и наличие различных шаблонов делают процесс создания диаграмм доступным даже для новичков. Пользователи могут легко добавлять, редактировать и настраивать элементы диаграмм, что способствует более эффективному обмену идеями и информацией.

Одной из ключевых особенностей Draw.io является возможность совместной работы в реальном времени. Пользователи могут одновременно редактировать диаграммы, что особенно полезно для команд, работающих над общими проектами. Интеграция с популярными облачными сервисами, такими как Google Drive, SharePoint и Dropbox, позволяет удобно сохранять и делиться созданными диаграммами.

Важно отметить, что Draw.io придерживается принципов конфиденциальности и безопасности. Пользователи имеют возможность хранить свои данные в выбранном облачном хранилище или на локальном устройстве, что обеспечивает дополнительный уровень контроля над информацией. Платформа не имеет доступа к пользовательским данным, что делает её надежным инструментом для работы с чувствительной информацией.

В заключение, Draw.io является мощным и многофункциональным инструментом, который отвечает современным требованиям к визуализации данных и совместной работе. Его доступность на различных платформах и богатый функционал делают его идеальным выбором для построения диаграмм IDEF0 и других визуальных представлений в рамках образовательных и профессиональных проектов.

1. **Описание практического задания**

Material: содержит информацию об учебных материалах, таких как id, название, описание, URL файла и методы для управления материалами.

* id: уникальный идентификатор учебного материала;
* name\_code: короткий код или идентификатор материала;
* name: название материала;
* description: описание содержания материала;
* file\_url: url или путь к файлу, представляющему данный материал;
* upd\_info(): метод для обновления информации о материале;
* upd\_url(): метод для обновления url файла материала;
* get\_details(): метод для получения подробной информации о материале;

Lesson: описывает уроки, включая id, название, описание, длительность, дату, список материалов и методы для управления уроками.

* id: уникальный идентификатор урока;
* name\_code: короткий код или идентификатор урока;
* name: название урока;
* duration: продолжительность урока в минутах;
* description: описание содержания урока;
* date: дата и время проведения урока;
* materials: список учебных материалов, связанных с уроком;
* getdetails(): метод для получения подробной информации об уроке;
* addmaterial(material: material): метод для добавления нового учебного материала к уроку;
* remmaterial(material: material): метод для удаления учебного материала из урока;
* schedule(): метод для планирования урока;
* cancel(): метод для отмены урока;

Teacher: включает атрибуты учителя, такие как id, имя, фамилия, ник, а также методы для установки ставки, получения информации и обновления профиля.

* id: уникальный идентификатор учителя;
* nickname: ник или псевдоним учителя;
* name: имя учителя;
* surname: фамилия учителя;
* set\_rate(): метод для установки ставки или почасовой оплаты учителя;
* get\_info(): метод для получения информации об учителе;
* update\_info(): метод для обновления информации об учителе;
* start\_chat(): метод для начала чата с учителем;
* pause\_chat(): метод для приостановки чата с учителем;
* end\_chat(): метод для завершения чата с учителем;

Student: содержит данные студента, включая id, имя, фамилию, ник, средний балл, методы для получения списка оценок, средней оценки и обновления профиля.

* id: уникальный идентификатор студента;
* surname: фамилия студента;
* name: имя студента;
* nickname: ник или псевдоним студента;
* avg\_rate: средний балл успеваемости студента;
* get\_list\_of\_rate(): метод для получения списка оценок студента;
* get\_avg\_rate(): метод для получения средней оценки студента;
* get\_info(): метод для получения информации о студенте;
* update\_info(): метод для обновления информации о студенте;
* get\_lesson(): метод для получения информации об уроках студента;

User: представляет общие атрибуты пользователя, такие как id, логин, пароль, email, роль, а также методы для входа и выхода из системы.

* id: уникальный идентификатор пользователя;
* login: логин пользователя;
* password: пароль пользователя;
* email: электронная почта пользователя;
* role: роль пользователя (например, студент, учитель, администратор);
* login(login:string, password:string): метод для входа в систему;
* logout(): метод для выхода из системы;

Assignment: описывает задания, включая id, название, описание, дату дедлайна, и методы для получения деталей и обновления информации.

* id: уникальный идентификатор задания;
* title: название задания;
* description: описание задания;
* duedate: дата дедлайна выполнения задания;
* getdetails(): метод для получения подробной информации о задании;
* upd\_info(): метод для обновления информации о задании;

Отношение ассоциации используют, чтобы показать, что между классами (например, между двумя классами) существует некоторая связь. Обычно с помощью него на диаграмме классов показывают, что один класс пользуется функционалом другого класса. Такая связь между:

* Student и Lesson.
* Teacher и Lesson.
* Student и Assignment.
* Teacher и Assignment.
* Teacher и Material.

Отношение наследования используется, чтобы показать, что один класс является родителем (базовым классом или суперклассом) для другого класса (потомка, производного класса). Такая связь между User и Teacher, Student.

Отношение агрегации между двумя классами показывает, что один из них включает в себя другой класс в качестве составной части. При этом класс-часть может и существовать обособленно от класса-целого (далее мы раскроем смысл этой фразы). Такая связь между:

* Lesson и Material.
* Lesson и Assignment.

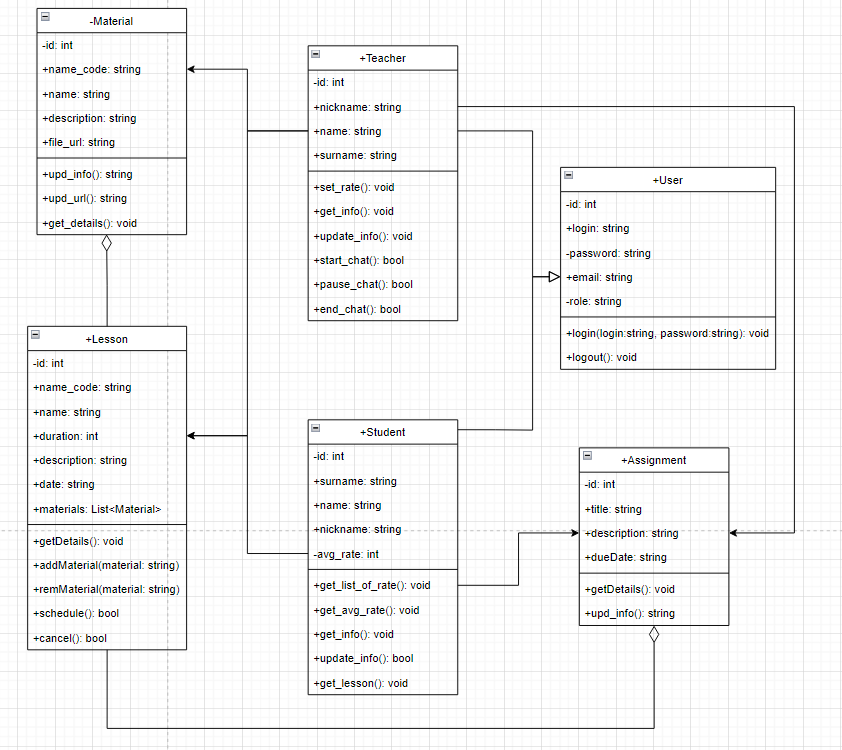


Рисунок 5.1 – Диаграмма классов

Это диаграмма пакетов, которая представляет структуру и взаимосвязи основных компонентов вашего приложения. Давайте рассмотрим ее более подробно:

Additional: пакет содержит дополнительные элементы, которые расширяют функциональность приложения. Внутри него находятся подпакеты для Animations, Media, Utilities и Logging.

Data: пакет отвечает за работу с данными в вашем приложении. Он включает в себя Data Repositories, DB Entities и компонент для DB Connection. Entities: пакет содержит основные сущности вашего приложения, такие как Teacher, Student, Lesson и Material.

DB Connection: компонент отвечает за подключение к базе данных и управление соединениями. DB Entities: пакет, содержащий классы-сущности, соответствующие таблицам в базе данных. Data Repositories: здесь находятся абстрактные классы и реализации репозиториев для работы с данными.

Такая структура пакетов позволяет разделить приложение на логические модули, обеспечивая модульность, масштабируемость и облегчая дальнейшее развитие и поддержку системы. Ключевые сущности вашей онлайн-школы, такие как Учитель, Студент, Урок и Материал, сгруппированы в отдельном пакете "Entities", что способствует их четкому разграничению и независимому тестированию.

Пакеты, связанные с работой с данными (Data, DB Connection, DB Entities, Data Repositories), изолируют логику доступа к базе данных, упрощая интеграцию с различными источниками данных в будущем. Дополнительные элементы, такие как анимации, медиа и утилиты, сосредоточены в пакете "Additional", который можно расширять по мере необходимости.

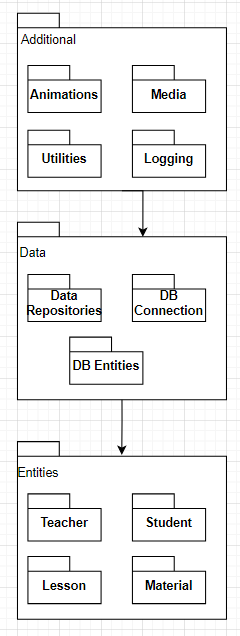


Рисунок 5.2 – Диаграмма пакетов

Общая структура диаграммы пакетов демонстрирует хорошо продуманную архитектуру вашего приложения, ориентированную на модульность, масштабируемость и расширяемость.