Министерство образования Республики Беларусь

Учреждение образование

«Белорусский государственный технологический университет»

Кафедра программной инженерии

«Проектирование интернет систем»

**Отчет к лабораторной работе №5**:

«Объектно-ориентированное моделирование. Структурные UML-диаграммы»

Выполнил:

Студент 4 курса 5 группы ФИТ

Почиковская Ю. С.

Проверила:

Пахолко А.С.

Минск 2022

# Цель работы

Изучение методологии объектно-ориентированного моделирования средствами UML. Ознакомление с основными принципами объектно-ориентированного проектирования программного обеспечения, получение навыков проектирования структуры информационной системы с применением UML.

**Теоретические вопросы:**

1. Перечислите структурные диаграммы, которые входят в UML 2.0.

Новая версия UML 2.0, разработанная ассоциацией OMG в 2005 г., отличается от UML 1.0 большей степенью формализации в связи с необходимостью автоматизации перевода описания на UML в машинный код.

Все расширения языка основаны на получении новых модулей путем вложения базовых модулей более низкого уровня в создаваемые модули более высоких уровней.

В UML 2.0 введено 13 типов диаграмм, разделенных на два подмножества — структурных и поведенческих диаграмм.

Структурные модели, известные также как статические модели, описывают структуру сущностей или компонентов некоторой системы, включая их классы, интерфейсы, атрибуты и отношения. К структурным относятся следующие диаграммы:

* диаграммы пакетов или контейнеров (Package diagrams);
* диаграммы классов (Class or Structural diagrams);
* диаграммы объектов (Object diagrams);
* композитные диаграммы (Composite Structure);
* диаграммы компонентов (Component diagrams), показывающие образование структур из классов и интерфейсы между структурами;
* диаграммы развертывания (Deployment diagrams).

1. Укажите назначение структурных диаграммы.

Основное назначение структурных диаграмм заключается в графическом представлении состава статистических совокупностей, характеризующихся как соотношение различных частей каждой из совокупностей. Состав статистической совокупности графически может быть представлен с помощью как абсолютных, так и относительных показателей.

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

Существует несколько обозначений диаграмм классов, которые используются при рисовании диаграмм классов UML. Мы перечислили ниже наиболее распространенные нотации диаграммы классов.

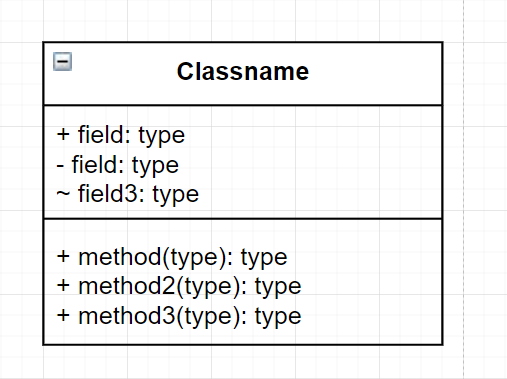


Рисунок 1 – Изображение класса

Классы представляют собой центральные объекты в системе. Он представлен прямоугольником с 3 отсеками.

Первый показывает имя класса, а средний – атрибуты класса, которые являются характеристиками объектов. В нижнем списке перечислены операции класса, которые представляют собой поведение класса.

Последние два отсека являются необязательными. Нотация класса без последних двух отделений называется простым классом и содержит только имя класса.

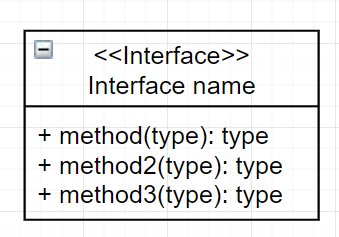


Рисунок 2 – Изображение интерфейса

Символ интерфейса на диаграммах классов обозначает набор операций, которые детализируют ответственность класса.

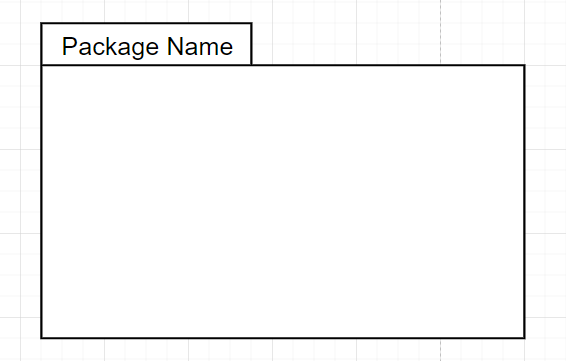


Рисунок 3 – Изображение пакета

Символ пакета используется для группировки классов или интерфейсов, которые либо похожи по своей природе, либо связаны. Группировка этих элементов дизайна с использованием символов упаковки улучшает читабельность диаграммы

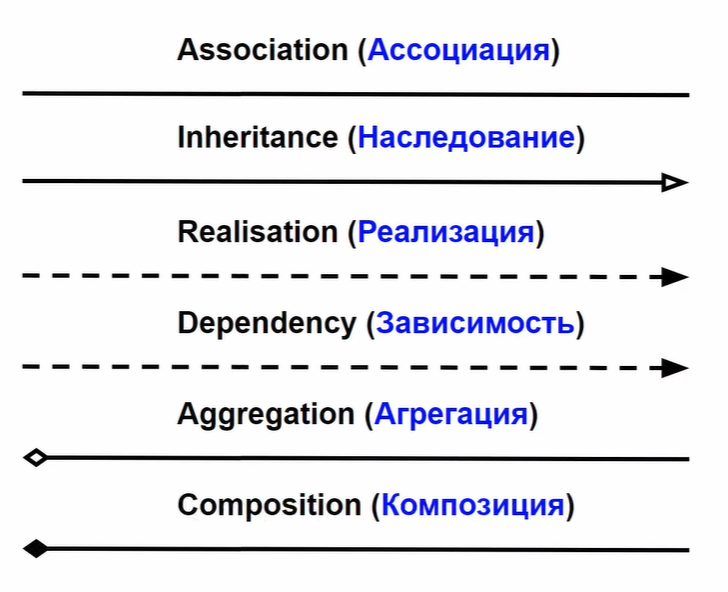


Рисунок 4 – Отношения в диаграмме классов

1. Для чего применяются расширения диаграмм UML?

Механизмы расширения UML включают: стереотипы (stereotype) - расширяют словарь UML, позволяя на основе существующих элементов языка создавать новые, ориентированные для решения конкретной проблемы; помеченные значения (tagged value) - расширяют свойства основных конструкций UML, позволяя включать дополнительную информацию в спецификацию элемента.

1. Что означают понятия «стереотип»

Стереотип- класс, связанный с "метамоделью", что означает стандартные элементы **UML**, определенные в стандарте **UML**. Таким образом, это класс классов (или класс ассоциаций, или любого другого элемента **UML**).

# Постановка задачи

Задачей данной работы выступает описание структуры взаимодействия классов в web-приложении, описанной в лабораторной работе №1.

Главными сущностями приложения выступают:

* AppContext – класс с помощью которого осуществляется доступ к базе данных данного web-приложения.
* AuthContoller – осуществляет обработку запросов, предназначенных пользователю. А именно: регистрация, авторизация, выход из аккаунта.
* ArticleController – осуществляет обработку запросов, предназначенных статье. А именно: получение всех статей, поиск статьи по id, добавление и удаление.
* RecepController – осуществляет обработку запросов, предназначенных начислению баллов. А именно: начисление баллов.

**Описание программных средств**

Построение моделей выполнялось в программной среде Draw.io.

Draw.io — инструмент для создания диаграмм, блок-схем, интеллект-карт, бизнес-макетов, отношений сущностей, программных блоков и другого. Сервис распространяется на бесплатной основе с открытым исходным кодом. Draw.io обладает богатым набором функций для визуализации большинства задач пользователя.

Адрес ресурса: https://app.diagrams.net/.

# Описание практического задания

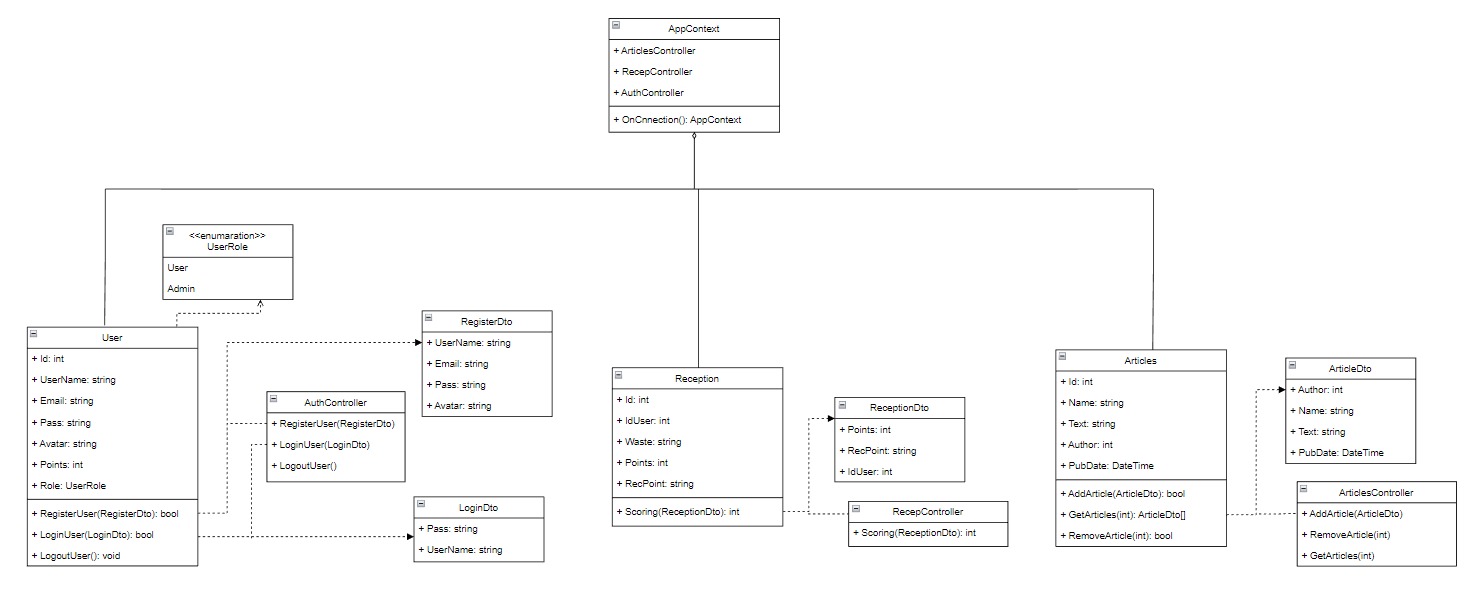


Рисунок 5 ­– Диаграмма классов

Класс AppContext реализует связь под названием “агрегация”, переменные, хранящиеся в данном классе описаны ниже:

* dbContext – приватная переменная, которая содержит экземпляр класса AppDbContext, является следствием реализации паттерна singleton;
* Users – публичная переменная, содержащая доступ к таблице users базы данных;
* Articles – публичная переменная, содержащая доступ к таблице articles базы данных;
* Reception – публичная переменная, содержащая доступ к таблице receptions базы данных;

Также этот класс имеет следующие методы:

* + OnConnect – защищенный метод, который выполняется при первом получении доступа к бд.

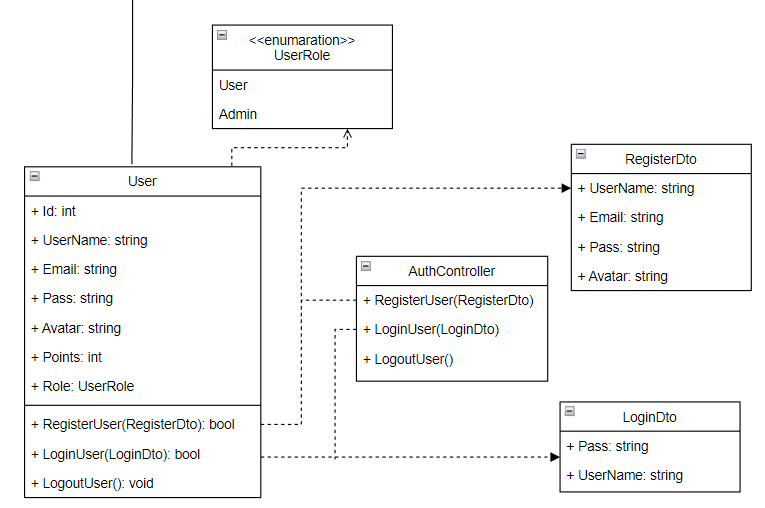


Рисунок 6 – Описание модуля “User”

UserController – осуществляет обработку запросов, предназначенных пользователю. А именно: регистрация, авторизация, выход из аккаунта, блокировка, разблокировка пользователя. Он содержит только публичные методы:

* + LoginUser – обработка запроса на авторизацию;
  + RegisterUser – обработка запроса на регистрацию;
  + LogoutUser – обработка запроса на выход из аккаунта;

Users – класс, содержащий свойства и методы для обработки запросов, которые приходят из контроллера UsersController. Далее перечислены все публичные свойства:

* + Id – id пользователя;
  + Username – ник пользователя;
  + Email – почта пользователя;
  + Pass – пароль пользователя;
  + Avatar – фото пользователя;
  + Points – баллы пользователя;
  + Role – роль пользователя, на основе которой определяются допустимые права пользователя.

Далее указаны публичные методы класса Users:

* RegisterUser – выполняет добавление данных пользователя в бд;
* LoginUser – выполняет проверку, зарегистрирован для данный пользователь;
* Logout – выполняет выход из аккаунта;

LoginDto - класс, который используется для получение данных, необходимых при авторизации. Содержит только публичные свойства:

* Username – ник пользователя;
* Password – пароль пользователя.

RegisterDto – класс, который используется для получение данных, необходимых при регистрации. Содержит только публичные свойства:

* Username – ник пользователя;
* Email – почта пользователя;
* Avatar – фото пользователя;
* Password – пароль пользователя.

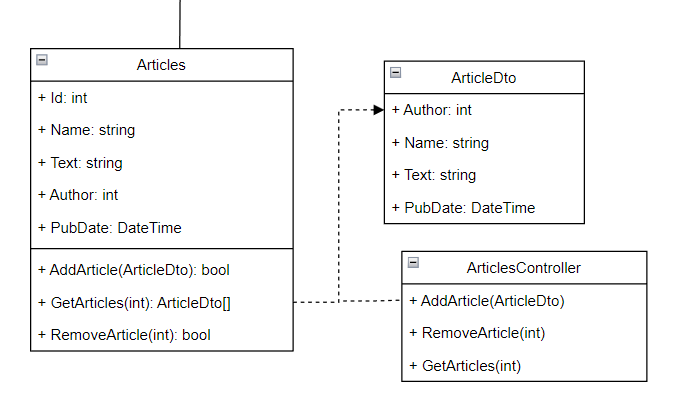


Рисунок 7 - Описание модуля “Article”

ArticleController осуществляет обработку запросов, предназначенных статье. А именно: получение всех статей, поиск статьи по id, добавление и удаление. Содержит только публичные методы:

* + GetArticles – обработка запроса на получение указанного количества статей;
  + RemoveArticle – обработка запроса на удаление одной статьи с указанным id;
  + AddArticle – обработка запроса на добавление статьи в бд;

Articles – класс, содержащий свойства и метод для обработки запросов, которые приходят из контроллера ArticlesController. Далее перечислены все публичные свойства:

* + Id – id статьи;
  + Author– id пользователя, которому принадлежит данная статья;
  + Name – название статьи;
  + TextArticle – текст статьи;
  + DatetimePublish – дата и время публикации статьи в web-приложение.

Далее указаны публичные методы класса Articles:

* AddArticle – выполняет добавление статьи в бд;
* RemoveArticle – осуществляет удаление статьи из бд;
* CheckEmpty – выполняет проверку заполненности всех полей при отправке запроса;
* FindArticleById – осуществляет поиска статьи по указанному id;
* GetArticles – выполняет получение указанного количества статей из бд.

ArticleDto – класс, который используется для получения данных, необходимых при добавлении статьи в бд. Содержит только публичные свойства:

* IdUser – id пользователя, который публикует данную статью;
* Name – название статьи;
* Text – текст статьи;
* PubDate – дата и время публикации статьи.

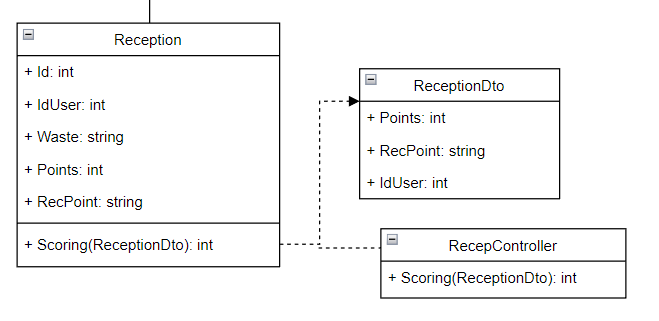


Рисунок 8 - Описание модуля “Comment”

RecepController - осуществляет обработку запросов, предназначенных для назначения баллов за сданные отходы. Содержит только публичные методы:

* + Scoring – обработка запроса на назначение баллов;

Reception – класс, содержащий свойства и метод для обработки запросов, которые приходят из контроллера CommentController. Далее перечислены все публичные свойства:

* Id – id назнкачения;
* IdUser – id юзера которому назначают баллы;
* Waste – вид отхода который сдавали;
* Points –назначенные баллы;
* RecPoint – пункт приема.

Далее указаны все публичные методы данного класса:

* + Scoring – обработка запроса на назначение баллов.

ReceptionDto – класс, который используется для получения данных, необходимых при добавлении баллов в бд. Содержит только публичные свойства:

* IdUser – id пользователя, который оставляет комментарий;
* Points –назначенные баллы;
* RecPoint – пункт приема.

Далее представлены пакеты, в которые были сгруппированы классы.

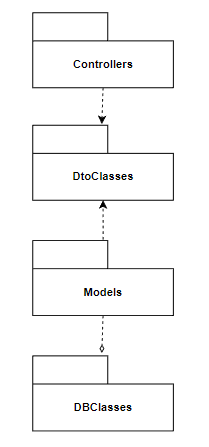


Рисунок 9 – Пакеты классов

Controllers содержит классы, отвечающие за обработку запросов на сервер. К ним относятся: AuthController, ArticleController, RecepController.

DtoClasses содержит классы, отвечающие за передаваемые в контроллерах данные. К этим классам относятся: LoginDto, RegisterDto, ArticleDto, ReceptionDto.

Models содержит классы, отвечающие за взаимодействие с базой данных. К этим классам относятся: Users, Articles, Reception.

DBClasses содержит классы, которые позволяют организовать соединение с бд. К данным классам относится AppContext.