Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ»

«Проектирование программного обеспечения»

Отчёт по лабораторной работе №5

Объектно-ориентированное моделирование. СТРУКТУРНЫЕ ДИАГРАММЫ UML

Выполнил: Ксюф Н. Д.

ФИТ 3 курс 3 группа

Преподаватель: Гончар Е. А

Цель: 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

1. **Постановка задачи**

Задачей данной работы выступает описание структуры взаимодействия классов в web-приложении, описанной в лабораторной работе №1. Необходимо разработать диаграмму классов для шашлычной, а также сгруппировать их в пакеты.

1. **Описание программных средств**

Для построения моделей использовалось программное средство 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.

Особенности Draw.io:

1. Более 500 шаблонов элементов и фигур;
2. Облегчённый интерфейс, в котором за короткий промежуток времени можно создать готовый проект;
3. Поддержка горячих клавиш, задействованных в большинстве графических редакторов;
4. Экспорт в форматы: JPG, PNG, SVG, VDSX;
5. Возможность совместной работы;
6. Наличие различных фоновых тем;
7. Мультиязычный интерфейс.

Программное средство draw.io также поддерживает импорт файлов в различных форматах, включая .vsdx (формат Microsoft Visio), Gliffy™ и Lucidchart™. Это позволяет вам работать с существующими диаграммами, созданными в других инструментах, и продолжать их редактирование и доработку в draw.io.

1. **Практическое задание**

Диаграмма классов для нашего проекта будет выглядеть следующим образом – Рисунок 1.

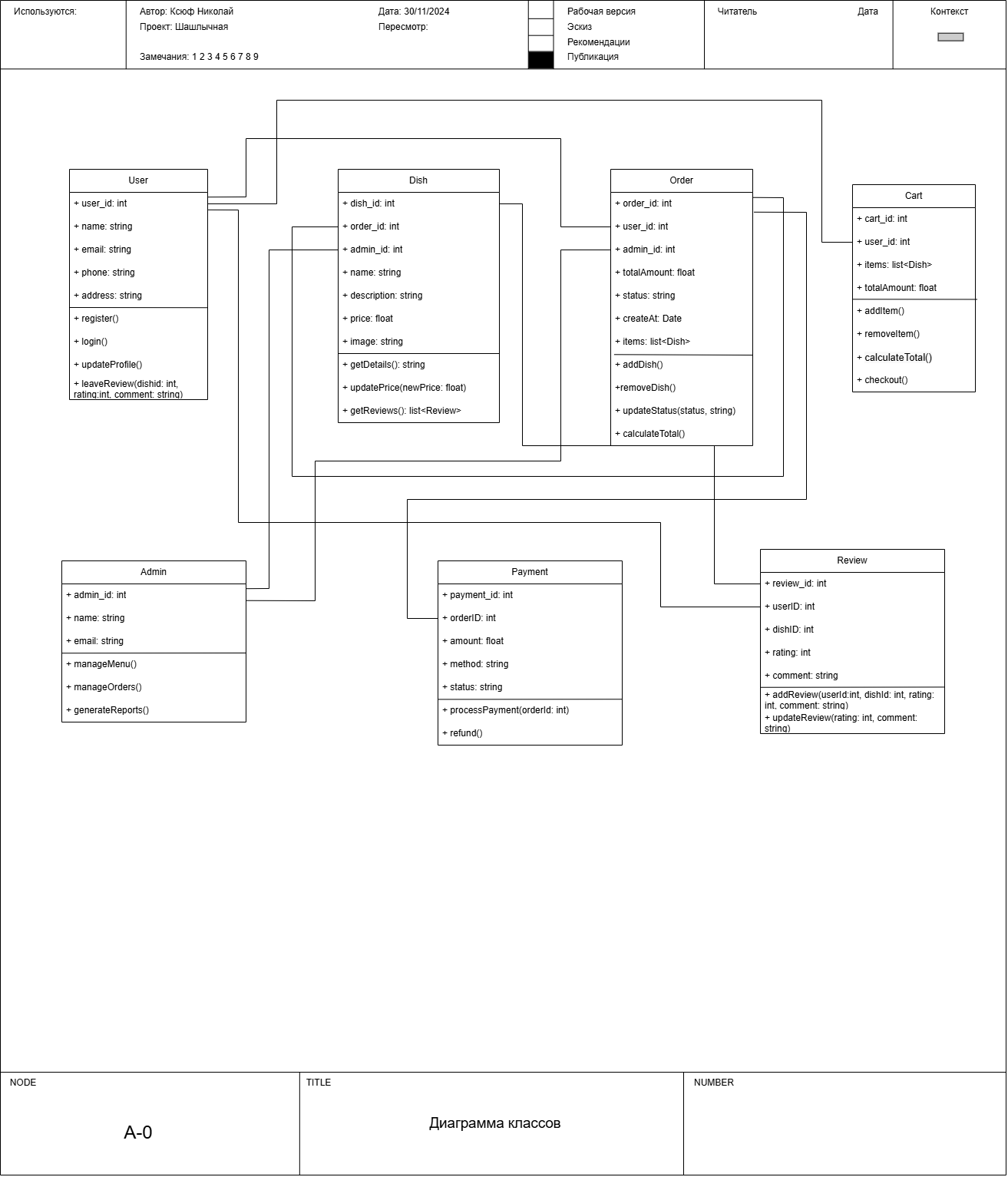


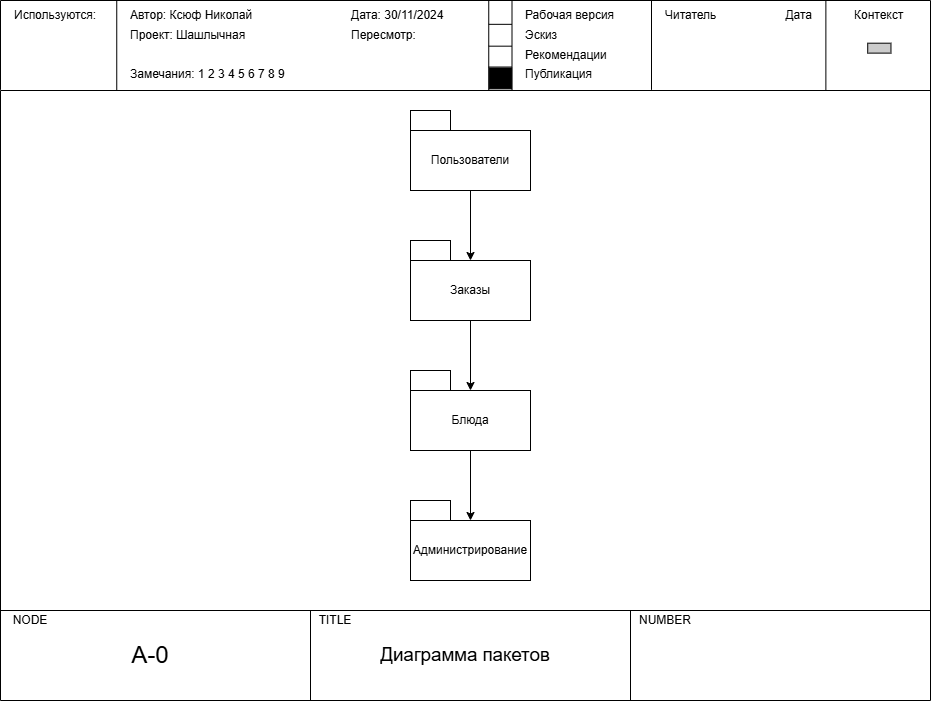
Рисунок 1 – Диаграмма классов

Следует описать их поподробнее.

Диаграмма классов представляет собой модель системы управления заказами в ресторане, включающую ключевые классы: User (Пользователь), Order (Заказ), Cart (Корзина), Dish (Блюдо), Admin (Администратор), Payment (Платеж) и Review (Отзыв). Каждый класс имеет уникальные атрибуты, такие как идентификаторы, названия и методы для выполнения различных операций, например, регистрации пользователей, обработки заказов и управления меню. Это позволяет четко структурировать взаимодействие между сущностями системы.

Связи между классами отображают их взаимозависимости: один пользователь может иметь множество заказов и отзывов, каждый заказ включает несколько блюд и соответствует одному платежу. Администраторы могут управлять как заказами, так и блюдами, обеспечивая возможность эффективного администрирования системы. Эта диаграмма служит основой для разработки и реализации функциональности системы, упрощая понимание архитектуры и взаимодействия компонентов. Если сгруппировать всё вышеперечисленное в пакеты, то получится следующее – Рисунок 2.

Рисунок 2 – Диаграмма пакетов



На данном рисунке пакет "Пользователи" включает элементы, связанные с регистрацией и отзывами клиентов. Пакет "Заказы" отвечает за управление процессами оформления заказов, корзин и платежей. Пакет "Блюда" охватывает все аспекты, связанные с меню и доступными блюдами, в то время как пакет "Администрирование" управляет всеми процессами и взаимодействиями в системе. Взаимозависимости между пакетами подчеркивают их взаимодействие: пользователи создают заказы, которые требуют блюд и обработки платежей, а администраторы контролируют и координируют работу всех компонентов системы.

**Вывод**: В результате данной лабораторной работы были получены навыки проектирования структуры информационной системы с применением UML, получены теоретические знания об основных принципах объектно-ориентированного проектирования программного обеспечения в рамках методологии объектно-ориентированного моделирования средствами UML.

**Ответы на 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.

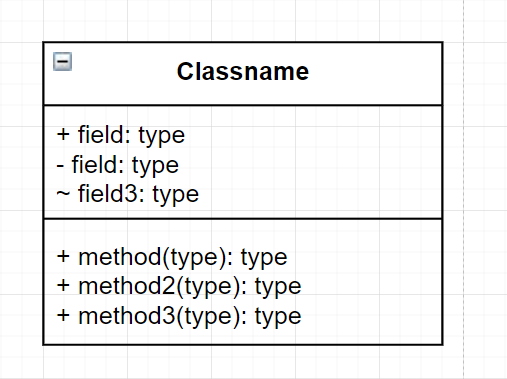


Рисунок 3 – Изображение класса

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.

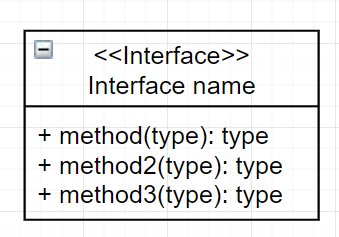


Рисунок 4 – Изображение интерфейса

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

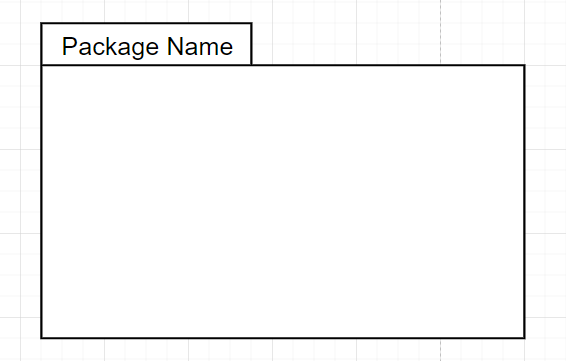


Рисунок 5 – Изображение пакета

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

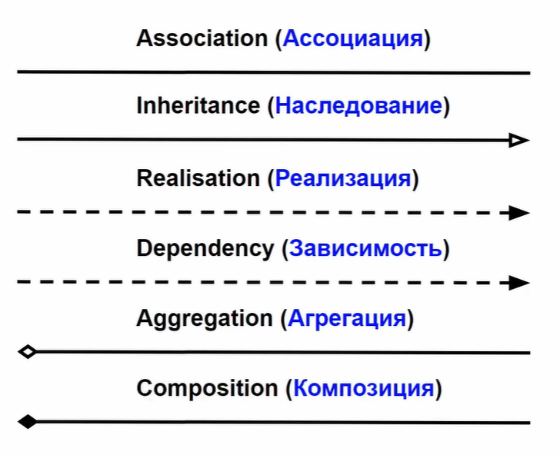


Рисунок 6 – Отношения в диаграмме классов

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).