**Диаграмма кооперации**

**Описание диаграммы кооперации**

Диаграмма кооперации описывает взаимодействие между участниками (объектами, компонентами или ролями) в процессе создания заказа пиццы. Она показывает последовательность действий и обмен сообщениями между элементами системы.

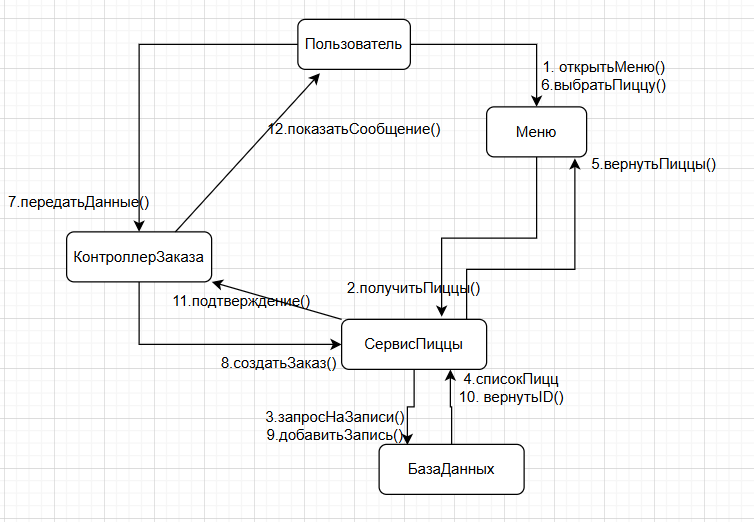


Рисунок 1.

**Участники (Компоненты)**

1. Пользователь
   1. Инициатор процесса заказа.
   2. Взаимодействует с меню и контроллером заказа.
2. Меню
   1. Отображает доступные варианты пицц.
   2. Обеспечивает выбор пиццы пользователем.
3. Контроллер заказа
   1. Управляет процессом оформления заказа.
   2. Взаимодействует с сервисом пиццы и базой данных.
4. Сервис пиццы
   1. Обрабатывает логику, связанную с пиццами (создание заказа, добавление записей).
5. База данных
   1. Хранит информацию о заказах и доступных пиццах.

**Функционал и последовательность действий**

1. открытьМеню() - Пользователь инициирует процесс, открывая меню пиццерии.
2. получитьПиццы() - Система отображает список доступных пицц.
3. запросНаЗаписи() - Контроллер заказа запрашивает актуальные данные о пиццах из базы данных.
4. списокПицц() - База данных возвращает информацию о пиццах
5. вернутьПиццы() - Система возвращает список доступных пицц пользователю.
6. выбратьПиццу() - Пользователь делает выбор конкретной пиццы из меню.
7. передатьДанные() - Данные о выбранной пицце передаются в контроллер заказа.
8. создатьЗаказ() - Сервис пиццы создает новый заказ на основе полученных данных.
9. добавитьЗапись() - Информация о заказе сохраняется в базе данных.
10. вернутьID() – База данных передает системе ID нового заказа
11. подтверждение() - Контроллер заказа подтверждает успешное создание заказа пользователю.

**Ключевые взаимодействия**

1. Пользователь < > Меню: Выбор пиццы.
2. Пользователь < >КонтроллерЗаказа: Передача данных и подтверждение.
3. КонтроллерЗаказа < >СервисПиццы: Создание заказа и управление записями.
4. СервисПиццы < >БазаДанных: Сохранение и извлечение данных.

Диаграмма отражает процесс от выбора пиццы до сохранения заказа в базе данных.

**Диаграмма классов**

**Описание диаграммы классов системы заказа пиццы**

Диаграмма представляет полную модель данных для системы онлайн-заказа пиццы с детализацией всех ключевых сущностей и их взаимосвязей.

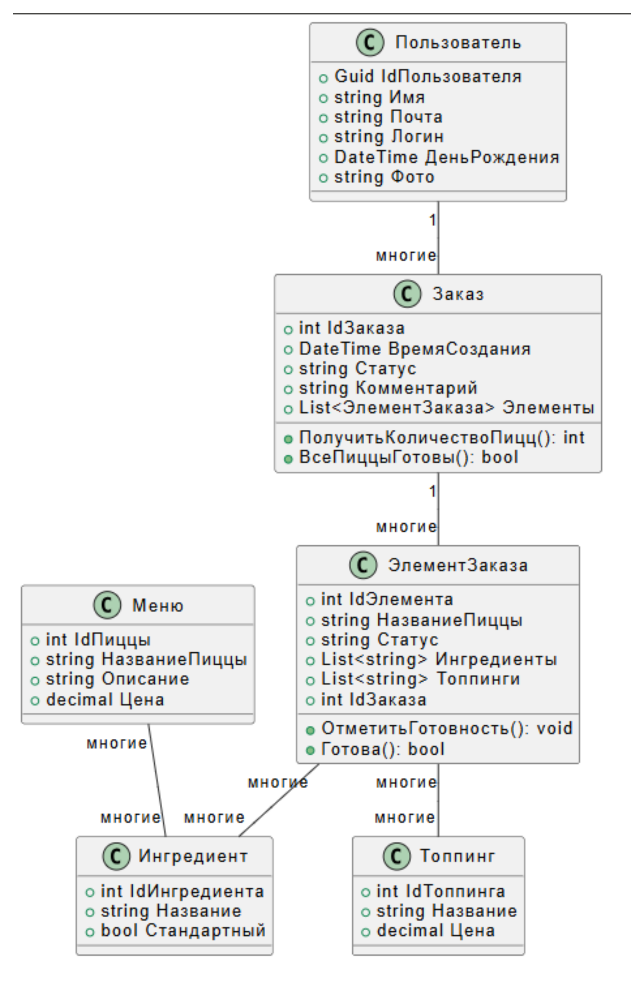


Рисунок. 2

**Основные компоненты системы**

Описание диаграммы классов системы заказа пиццы

1. Пользователь (User)

Атрибуты:

1. «IdПользователя» (Guid) - уникальный идентификатор
2. «Имя» (string) - имя пользователя
3. «Почта» (string) - электронная почта
4. «Логин» (string) - учетные данные
5. «ДеньРождения» (DateTime) - дата рождения
6. «Фото» (string) - ссылка на аватар

Назначение: сущность представляет клиента системы. Хранит персональные данные и учетную информацию.

2. Заказ (Order)

Атрибуты:

1. «IdЗаказа» (int) - уникальный идентификатор
2. «ВремяСоздания» (DateTime) - дата/время оформления
3. «Статус» (string) - текущее состояние заказа
4. «Комментарий» (string) - доп. информация от клиента
5. «Элементы» (List<ЭлементЗаказа>) - состав заказа

Методы:

1. «ПолучитьКоличествоПицц(): int» - возвращает общее количество пицц в заказе
2. «ВсеПиццыГотовы(): bool» - проверяет готовность всех позиций

Назначение: агрегирует информацию о заказе. Управляет коллекцией элементов заказа и их статусами.

3. Элемент заказа (OrderItem)

Атрибуты:

1. «IdЭлемента» (int) - уникальный идентификатор
2. «НазваниеПиццы» (string) - наименование позиции
3. «Статус» (string) - состояние приготовления
4. «Ингредиенты» (List<string>) - состав пиццы
5. «Топпинги» (List<string>) - дополнительные ингредиенты
6. «IdЗаказа» (int) - ссылка на родительский заказ

Методы:

1. «ОтметитьГотовность(): void» - изменяет статус на "готово"
2. «Готово (): bool» - проверяет состояние готовности

Назначение: представляет отдельную позицию в заказе с детализацией состава и статусом приготовления.

4. Меню (Menu)

Атрибуты:

1. «IdПиццы» (int) - уникальный идентификатор
2. «НазваниеПиццы» (string) - наименование
3. «Описание» (string) - детали рецепта
4. «Цена» (decimal) - стоимость

Назначение: каталог доступных для заказа пицц с описанием и ценами.

5. Ингредиент (Ingredient)

Атрибуты:

1. «IdИнгредиента» (int) - уникальный идентификатор
2. «Название» (string) - наименование
3. «Стандартный» (bool) - флаг базового ингредиента

Назначение: справочник ингредиентов, используемых при приготовлении пицц.

6. Дополнение (Topping)

Атрибуты:

1. «IdТоппинга» (int) - уникальный идентификатор
2. «Название» (string) - наименование
3. «Цена» (decimal) - дополнительная стоимость

Назначение: дополнительные ингредиенты, которые можно добавить к пицце за отдельную плату.

**Связи между классами**

1. Пользователь → Заказ (1 → много)

- Один пользователь может создавать множество заказов

2. Заказ → ЭлементЗаказа (1 → много)

- Заказ содержит несколько позиций (пицц)

3. ЭлементЗаказа → Меню (ассоциация)

- Каждая позиция заказа ссылается на пиццу из меню

4. ЭлементЗаказа ↔ Ингредиент/Топпинг

- Состав пиццы формируется из ингредиентов и топпингов

**Функциональность системы**

1. Оформление заказа:
   1. Пользователь выбирает пиццы из меню
   2. Система создает заказ с элементами
   3. Учитываются стандартные и дополнительные ингредиенты
2. Отслеживание статусов:
   1. Контроль готовности отдельных позиций
   2. Автоматическая проверка полной готовности заказа
3. Управление составом:
   1. Возможность расширения списка ингредиентов
   2. Гибкая система топпингов с индивидуальным ценообразованием

**Диаграмма состояний заказа**

После того как сотрудник оформил заказ он попадает в состояние «Оформлен», далее идет его обработка и он заливается в базу под статусом «В обработке», после он отображается в разделе, где повара могут его выполнить нажимают начать, и он попадает в текущий и когда готово, заказ переходит в выполнен.

Основные состояния:

1. Новый – начальное состояние заказа.
2. Оформлен – сотрудник подтвердил заказ.
3. В обработке – заказ внесен в базу и ожидает выполнения.
4. Текущий – повар начал готовить заказ.
5. Выполнен – заказ готов и передан клиенту.
6. Отменен – заказ отменен на любом этапе.

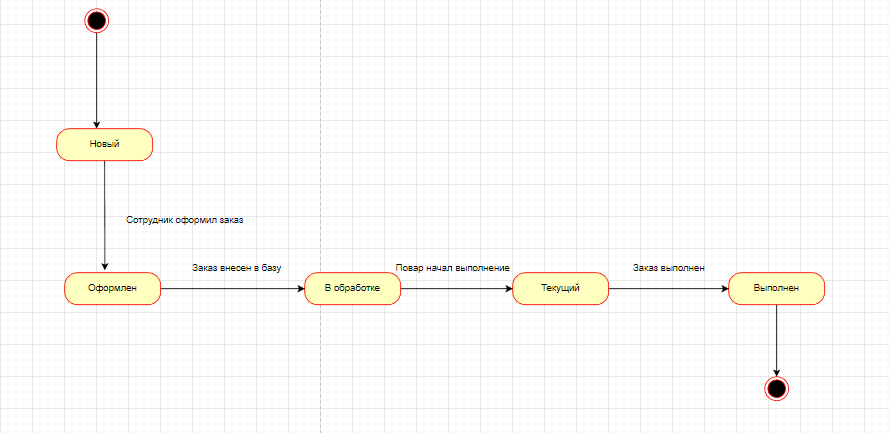


Рисунок 1 – диаграмма состояний

**Диаграмма компонентов**

Компонент Заказ является основным он связан со многими другими компонентами в системе.

Компонент Заказ требует проведения платежа от оплаты, а оплата предоставляет оплату товаров заказу.

Компонент Чек предоставляет генерацию чека для заказа, а заказ ее требует для корректной работы.

Компонент Комментарии предоставляет поле для добавлений заметок, а заказ требует это от компонента.

Чтобы выдать чек, этот компонент требует оплату от компонента оплаты.

Корзина передает заказу данные о выбранных товаров и требует создать заказ на этой основе, а заказ это предоставляет.

Меню требует список товаров, а товары это предоставляют.

Меню требует от корзины методы управления, чтобы добавлять удалять и убирать товары.

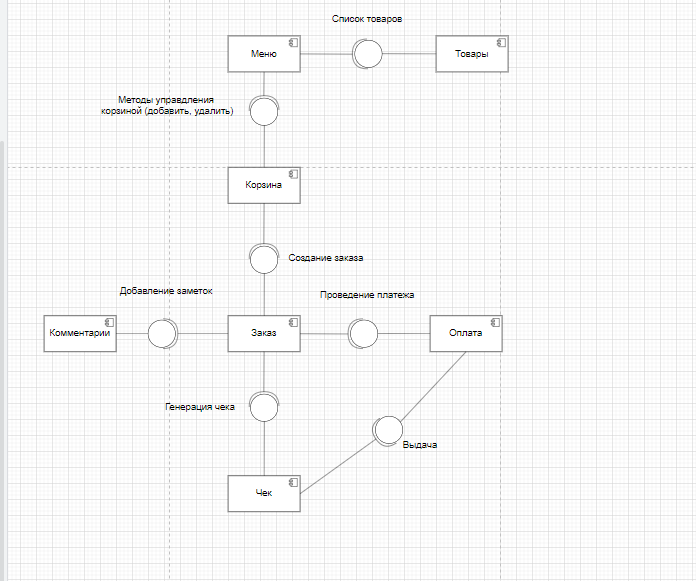


Рисунок 2 – диаграмма компонентов