|  |  |  |
| --- | --- | --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики» | | |
| *Факультет социально-экономических и компьютерных наук* | | |
|  | | |
| Кривов Андрей Викторович | | |
|  | | |
| **РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПЕРСОНАЛА МАГАЗИНА КОМПЬЮТЕРНЫХ КОМПЛЕКТУЮЩИХ** | | |
| *Курсовая работа* | | |
| студента образовательной программы «Программная инженерия»  по направлению подготовки *09.03.04 Программная инженерия* | | |
|  |  | Руководитель  преподаватель кафедры информационных технологий в бизнесе  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Г. И. Рустамханова |

Пермь, 2023 год

**Аннотация**

Автор: студент НИУ ВШЭ Кривов Андрей Викторович, кафедра информационных технологий в бизнесе.

Тема работы: Разработка информационной системы для персонала магазина компьютерных комплектующих.

Актуальность работы обусловлена…

В первой главе рассматривается… Вторая глава посвящена… В третьей главе описана реализация… Четвертая глава представляет собой…

Работа состоит из N глав, включает N страниц основного текста, N иллюстраций, N таблиц и N приложений. Библиографический список состоит из N источников.

**ОГЛАВЛЕНИЕ**

[Введение 4](#_Toc129980447)

[Глава 1 Анализ задачи и разработка требований 5](#_Toc129980448)

[1.1 Постановка задачи 5](#_Toc129980449)

[1.2 Анализ существующих решений 5](#_Toc129980450)

[Глава 2 Проектирование информационной системы 8](#_Toc129980451)

[2.1 Проектирование базы данных 8](#_Toc129980452)

[2.1.1 Описание атрибутов 8](#_Toc129980453)

[2.1.2 Установление функциональных зависимостей 10](#_Toc129980454)

[2.1.3 Нормализация базы данных 11](#_Toc129980455)

[2.2 Проектирование интерфейса приложения 18](#_Toc129980456)

[2.3 Проектирование отчетов 19](#_Toc129980457)

[2.4 Результаты проектирования 19](#_Toc129980458)

[Глава 3 Реализация информационной системы 20](#_Toc129980459)

[3.1 Выбор инструментальных средств 20](#_Toc129980460)

[3.2 Создание моделей 20](#_Toc129980461)

[3.2.1 Создание модели пользователей 20](#_Toc129980462)

[3.2.2 Создание модели складов 21](#_Toc129980463)

[3.2.3 Создание модели поставщиков 22](#_Toc129980464)

[3.2.4 Создание модели категорий комплектующих 22](#_Toc129980465)

[3.2.5 Создание модели комплектующих 22](#_Toc129980466)

[3.2.6 Создание модели типов характеристик 23](#_Toc129980467)

[3.2.7 Создание модели характеристик 23](#_Toc129980468)

[3.2.8 Создание модели учета комплектующих на складах 23](#_Toc129980469)

[3.2.9 Создание модели поставок 23](#_Toc129980470)

[3.2.10 Создание модели заказов 23](#_Toc129980471)

[3.2.11 Создание модели составов заказов 23](#_Toc129980472)

[3.2.12 Создание модели конфигураций 23](#_Toc129980473)

[3.3 Реализация интерфейса 23](#_Toc129980474)

**Введение**

Здесь будет введение…

**Глава 1 Анализ задачи и разработка требований**

Данная глава обосновывает актуальность проекта, представляет обзор существующих решений информационной системы для магазина компьютерных комплектующих.

**1.1 Постановка задачи**

Целью курсовой работы является разработка информационной системы для персонала магазина компьютерных комплектующих с целью упрощения процессов управления продажами и инвентаризацией товаров, оптимизации времени и повышения эффективности работы магазина и персонала. Для достижения этой цели необходимо выполнить следующие задачи:

1. Проанализировать предметную область.
2. Создать модель данных.
3. Спроектировать приложение.
4. Разработать запросы к данным.
5. Спроектировать отчеты.
6. Реализовать информационную систему.
7. Протестировать и отладить приложение.

**1.2 Анализ существующих решений**

На сегодняшний день существует множество приложений для управления магазином и учета товаров. Но данные решения, как правило, являются универсальными. Информация система, которая должна быть разработана в рамках данной курсовой работы, является узконаправленной. Она охватывает магазины, продающие только компьютерные комплектующие.

1С:Предприятие – одна из платформ, наиболее распространенных на российском рынке. Для этой платформы существует решение «1С:Розница 8. Магазин бытовой техники и средств связи» для работы со специфическим ассортиментом (бытовая техника, сотовые телефоны, компьютеры и комплектующие) и бизнес-процессами (включая гарантийное обслуживание и обмен по схеме trade-in).

Основные функциональные возможности данного решения:

1. Управление закупками и запасами.
2. Управление складом.
3. Управление продажами.
4. Учет платежных средств.
5. Управление персоналом магазина.
6. Обмен данными.
7. Поддержка онлайн ККТ (54-ФЗ).
8. 1С:Мобильная касса.
9. Система отчетности.
10. Подключаемое торговое оборудование.

На рисунке 1.1 изображено окно оформления поставки, в котором необходимо указать поставщика, магазин, склад, организацию и другие данные. При добавлении товара заполняется артикул, номенклатура, количество товара в выбранной единице измерения, закупочная цена. НДС учитывается.



***Рис. 1.1 – Оформление поступления товара в 1C:Предприятие***

Приложение позволяет генерировать отчеты по продажам, включая визуализацию в виде столбчатой диаграммы по годам (см. рисунок 1.2).



***Рис. 1.2 – Окно продаж в 1C:Предприятие***

Текст…

**Глава 2 Проектирование информационной системы**

Здесь будет текст…

**2.1 Проектирование базы данных**

Здесь будет текст…

**2.1.1 Описание атрибутов**

Для выполнения описанных на этапе анализа операций по автоматизации учета компьютерных комплектующих необходимо, чтобы в базе данных хранилась информация, описанная в таблице 2.1.

Таблица 2.1 – Атрибуты базы данных

| **Имя атрибута** | **Тип данных** | **Ограничение на значение атрибута** | **Комментарий** |
| --- | --- | --- | --- |
| Код пользователя | Целое число | Уникально, обязательно для заполнения |  |
| Пароль | Строка символов | Обязательно для заполнения | Хранится в захэшированном (SHA256) виде |
| Дата регистрации | Целое число | Обязательно для заполнения | В формате Unix Timestamp, значение по умолчанию – текущая дата |
| Дата последнего входа | Целое число |  | В формате Unix Timestamp |
| Код группы прав | Целое число | Обязательно для заполнения | 5 групп прав (от 0 до 4) |
| Код склада | Целое число | Уникально, обязательно для заполнения |  |
| Город | Строка символов | Обязательно для заполнения, максимальная длина – 64 символа |  |
| Улица | Строка символов | Обязательно для заполнения, максимальная длина – 64 символа |  |
| Дом | Строка символов | Обязательно для заполнения, максимальная длина – 8 символов |  |
| Индекс | Целое число | Обязательно для заполнения | Маска - 000000 |
| Код поставщика | Целое число | Уникально, обязательно для заполнения |  |
| Название организации поставщика | Строка символов | Уникально, обязательно для заполнения, максимальная длина – 128 символов |  |
| Код категории комплектующих | Целое число | Уникально, обязательно для заполнения |  |
| Название категории комплектующих | Строка символов | Уникально, обязательно для заполнения, максимальная длина – 64 символа |  |
| Код комплектующего | Целое число | Уникально, обязательно для заполнения |  |
| Название модели комплектующего | Строка символов | Уникально, обязательно для заполнения, максимальная длина – 256 символов |  |
| Гарантия комплектующего | Целое число | Обязательно для заполнения, не может быть меньше 1 |  |
| Код типа характеристики | Целое число | Уникально, обязательно для заполнения |  |
| Название типа характеристики | Строка символов | Уникально, обязательно для заполнения, максимальная длина – 64 символа |  |
| Код характеристики | Целое число | Уникально, обязательно для заполнения |  |
| Значение характеристики | Строка символов | Обязательно для заполнения, максимальная длина – 128 символов |  |
| Код хранения комплектующего на складе | Целое число | Уникально, обязательно для заполнения |  |
| Цена комплектующего | Вещественное число | Обязательно для заполнения, не может быть равно или меньше 0 |  |
| Количество комплектующего | Целое число | Обязательно для заполнения, не может быть меньше 1 |  |
| Код поставки | Целое число | Уникально, обязательно для заполнения |  |
| Закупочная цена комплектующего | Вещественное число | Обязательно для заполнения, не может быть равно или меньше 0 |  |
| Дата поставки | Целое число | Обязательно для заполнения | В формате Unix Timestamp |
| Код заказа | Целое число | Уникально, обязательно для заполнения |  |
| Дата заказа | Целое число | Обязательно для заполнения | В формате Unix Timestamp |
| Статус заказа | Целое число | Обязательно для заполнения | 5 статусов (от 0 до 4) |
| Код состава заказа | Целое число | Уникально, обязательно для заполнения |  |
| Часть конфигурации | Логический тип | Обязательно для заполнения |  |
| Код конфигурации | Целое число | Уникально, обязательно для заполнения |  |
| Код части конфигурации | Целое число | Уникально, обязательно для заполнения |  |

**2.1.2 Установление функциональных зависимостей**

Рассмотрим следующие функциональные зависимости между атрибутами в каждом отношении:

1. По коду пользователя можно однозначно определить пароль, дату регистрации, дату последнего входа, группу прав.
2. По коду склада можно однозначно определить город, улицу, дом, индекс.
3. По коду поставщика можно однозначно определить название организации поставщика.
4. По коду категории комплектующих можно однозначно определить название категории комплектующих.
5. По коду комплектующего можно однозначно определить название модели комплектующего, гарантию комплектующего, код категории комплектующего.
6. По коду типа характеристики можно однозначно определить название типа характеристики.
7. По коду характеристики можно однозначно определить код типа характеристики, код комплектующего, значение характеристики.
8. По коду хранения комплектующего на складе можно однозначно определить цену комплектующего, количество комплектующего, код склада, код комплектующего.
9. По коду поставки можно однозначно определить закупочную цену комплектующего, количество комплектующего, дату поставки, код поставщика, код склада, код комплектующего.
10. По коду заказа можно однозначно определить дату заказа, статус заказа.
11. По коду состава заказа можно однозначно определить цену комплектующего, количество комплектующего, является ли комплектующее частью конфигурации, код заказа, код склада, код комплектующего.
12. По коду части конфигурации можно однозначно определить код конфигурации, количество комплектующего, код комплектующего.

**2.1.3 Нормализация базы данных**

Здесь будет текст…

**2.1.3.1 Приведение к 1НФ**

Отношение находится в первой нормальной форме, когда все его атрибуты имеют единственное значение (атомарные атрибуты) и все кортежи уникальны (наличие PK).

В данном случае исходное универсальное отношение включает следующие атрибуты:

1. Код пользователя.
2. Пароль.
3. Дата регистрации.
4. Дата последнего входа.
5. Код группы прав.
6. Код склада.
7. Город.
8. Улица.
9. Дом.
10. Индекс.
11. Код поставщика.
12. Название организации поставщика.
13. Код категории комплектующих.
14. Название категории комплектующих.
15. Код комплектующего.
16. Название модели комплектующего.
17. Гарантия комплектующего.
18. Код типа характеристики.
19. Название типа характеристики.
20. Код характеристики.
21. Значение характеристики.
22. Код хранения комплектующего на складе.
23. Цена комплектующего.
24. Количество комплектующего.
25. Код поставки.
26. Закупочная цена комплектующего.
27. Дата поставки.
28. Код заказа.
29. Дата заказа.
30. Статус заказа.
31. Код состава заказа.
32. Часть конфигурации.
33. Код конфигурации.
34. Код части конфигурации.

В соответствии с описанными выше функциональными зависимостями формируем первичный ключ отношения, который включает следующие атрибуты:

1. Код пользователя.
2. Код склада.
3. Код поставщика.
4. Код категории комплектующих.
5. Код комплектующего.
6. Код типа характеристики.
7. Код характеристики.
8. Код хранения комплектующего на складе.
9. Код поставки.
10. Код заказа.
11. Код состава заказа.
12. Код части конфигурации.

Их комбинация является уникальной для каждой строки, что позволяет однозначно идентифицировать запись.

Для того чтобы обеспечить выполнение требования атомарности, в каждой строке таблицы необходимо вводить все значения указанных атрибутов, не группируя данные. Это приводит к большой избыточности данных и следующим аномалиями при выполнении операций:

1. Аномалии ввода:

* невозможно ввести данные о заказах, если нет информации о комплектующих;
* невозможно ввести данные о поставках, если нет информации о складах;
* невозможно ввести данные о комплектующих, если нет информации о категориях комплектующих.

1. Аномалии редактирования:

* при изменении информации о комплектующем, потребуется продублировать изменения во всех строках, в которых хранится информация о данном комплектующем;
* при изменении информации о заказе, потребуется продублировать изменения во всех строках, в которых хранится информация о данном заказе.

1. Аномалии удаления:

* при удалении информации о поставках одного поставщика все данные о поставщике могут быть утеряны.

Для их устранения необходимо выполнить декомпозицию отношения в соответствии с алгоритмом нормализации.

**2.1.3.2 Приведение ко 2НФ**

Отношение находится во второй нормальной форме, если оно находится в первой нормальной форме и отсутствует частичная функциональная зависимость не ключевых атрибутов от ключа (не должно быть не ключевых полей, зависящих от части PK).

В соответствии с описанными выше зависимостями можно сделать вывод, что отношение уже находится во второй нормальной форме.

**2.1.3.3 Приведение к 3НФ**

Отношение находится в третьей нормальной форме, если оно находится во второй нормальной форме, нет функциональной зависимости между не ключевыми атрибутами (не ключевые атрибуты взаимно независимы) и каждый не ключевой атрибут не транзитивно зависит от ключа.

Текущее отношение не находится в третьей нормальной форме, так как по атрибуту «Код пользователя» можно однозначно определить следующие атрибуты:

1. Пароль.
2. Дата регистрации.
3. Дата последнего входа.
4. Группа прав.

По атрибуту «Код склада» можно однозначно определить следующие атрибуты:

1. Город.
2. Улица.
3. Дом.
4. Индекс.

По атрибуту «Код поставщика» можно однозначно определить атрибут «Название организации поставщика».

По атрибуту «Код категории» можно однозначно определить атрибут «Название категории комплектующих».

По атрибуту «Код комплектующего» можно однозначно определить следующие атрибуты:

1. Название модели комплектующего.
2. Гарантия комплектующего.
3. Код категории комплектующего.

По атрибуту «Код типа характеристики» можно однозначно определить атрибут «Название типа характеристики».

По атрибуту «Код характеристики» можно однозначно определить следующие атрибуты:

1. Код типа характеристики.
2. Код комплектующего.
3. Значение характеристики.

По атрибуту «Код хранения комплектующего на складе» можно однозначно определить следующие атрибуты:

1. Цена комплектующего.
2. Количество комплектующего.
3. Код склада.
4. Код комплектующего.

По атрибуту «Код поставки» можно однозначно определить следующие атрибуты:

1. Цена комплектующего.
2. Количество комплектующего.
3. Дата поставки.
4. Код поставщика.
5. Код склада.
6. Код комплектующего.

По атрибуту «Код заказа» можно однозначно определить следующие атрибуты:

1. Дата заказа.
2. Статус заказа.

По атрибуту «Код состава заказа» можно однозначно определить следующие атрибуты:

1. Цена комплектующего.
2. Количество комплектующего.
3. Часть конфигурации.
4. Код заказа.
5. Код склада.
6. Код комплектующего.

По атрибуту «Код части конфигурации» можно однозначно определить следующие атрибуты:

1. Код конфигурации.
2. Количество комплектующего.
3. Код комплектующего.

Таким образом, в результате приведения универсального отношения к третьей нормальной форме, должны быть выделены следующие таблицы (подчеркиванием выделены первичные ключи, штриховым подчеркиванием – внешние ключи):

1. Таблицы «Пользователи»:

* код пользователя;
* пароль;
* дата регистрации;
* дата последнего входа;
* группа прав.

1. Таблица «Склады»:

* код склада;
* город;
* улица;
* дом;
* индекс.

1. Таблица «Поставщики»:

* код поставщика;
* название организации поставщика.

1. Таблица «Категории комплектующих»:

* код категории комплектующих;
* название категории комплектующих.

1. Таблица «Комплектующие»:

* код комплектующего;
* название модели комплектующего;
* гарантия комплектующего;
* код категории комплектующего.

1. Таблица «Типы характеристик»:

* код типа характеристики;
* название типа характеристики.

1. Таблица «Характеристики»:

* код характеристики;
* код типа характеристики;
* код комплектующего;
* значение характеристики.

1. Таблица «Учет комплектующих на складах»:

* код хранения комплектующего на складе;
* цена комплектующего;
* количество комплектующего;
* код склада;
* код комплектующего.

1. Таблица «Поставки»:

* код поставки;
* закупочная цена комплектующего;
* количество комплектующего;
* дата поставки;
* код поставщика;
* код склада;
* код комплектующего.

1. Таблица «Заказы»:

* код заказа;
* дата заказа;
* статус заказа.

1. Таблица «Состав заказов»:

* код состава заказа;
* цена комплектующего;
* количество комплектующего;
* часть конфигурации;
* код заказа;
* код склада;
* код комплектующего.

1. Таблица «Конфигурации»:

* код части конфигурации;
* код конфигурации;
* количество комплектующего;
* код комплектующего.

**2.2 Проектирование интерфейса приложения**

Информация система разрабатывается для персонала, следовательно, в приложении должно быть окно авторизации, включающее в себя следующие элементы пользовательского интерфейса:

1. Поле для ввода идентификатора сотрудника.
2. Поле для ввода пароля.
3. Кнопка авторизации.

В случае ввода некорректных данных интерфейс должен сообщить об этом пользователю и предоставить возможность ввести их заново. Если данные соответствуют действительности, то должно открыться главное окно, а окно авторизации закрыться.

В главном окне должны быть расположены вкладки, соответствующие таблицам из базы данных (всего 12 таблиц). Взаимодействие (в том числе выполнение CRUD операций) будет происходить через таблицы. В случае с таблицей «Пользователи» добавление и редактирование записей должно осуществляться через диалоговое окно, так как это необходимо для ввода пароля. На каждой вкладке требуется наличие кнопки создания отчета и кнопки фильтрации, если в таблице присутствует больше 1 столбца (помимо столбца с уникальным идентификатором).

**2.3 Проектирование отчетов**

Информационная система предлагает возможность генерирования 12 отчетов, столько же, сколько таблиц в базе данных. Отчеты должны повторять структуру соответствующих таблиц. Формат отчетов – Excel (расширение .xlsx).

**2.4 Результаты проектирования**

В результате этапа проектирования были получены следующие результаты:

1. База данных состоит из 12 таблиц (см. рисунок 2.1).
2. Спроектирован интерфейс приложения (состоит из окна авторизации и главного окна, которое включает в себя 12 вкладок для взаимодействия с таблицами из базы данных).
3. Количество отчетов равно 12, что соответствует количеству таблиц из базы данных. Отчеты повторяют структуры соответствующих таблиц.

**Глава 3 Реализация информационной системы**

Здесь будет текст…

**3.1 Выбор инструментальных средств**

В качестве языка программирования выбран C#, один из самых популярных и наиболее часто применяемых в разработке прикладных приложений. Интерфейс будет построен при помощи UI фреймворка WPF.

База данных будет храниться в СУБД PostgreSQL. Выбор обусловлен низкими минимальными требованиями и простым процессом установки. Для доступа к таблицам и атрибутам из приложения будет использоваться ORM фреймворк – Entity Framework Core 7.

**3.2 Создание моделей**

В данном разделе будет описано создание классов, соответствующих таблицам из базы данных, на языке программирования C#. Это позволит выполнять запросы к базе данных без написания SQL-кода.

Каждый класс описан в соответствии с требованиями фреймворка Entity Framework Core 7 и подходом «Data Annotations».

**3.2.1 Создание модели пользователей**

Атрибуты таблицы «Пользователи»:

* код пользователя;
* пароль;
* дата регистрации;
* дата последнего входа;
* группа прав.

Для создания модели необходимо написать класс, повторяющий структуру таблицы. На рисунке 3.1 приведен код модели на C#.



***Рис. 3.1 – Класс «User»***

В класс также добавлены свойства RegistrationDateText и LastVisitDateText, которые будут использованы для вывода даты регистрации и даты последнего входа в текстовом формате.

**3.2.2 Создание модели складов**

Атрибуты таблицы «Склады»:

* код склада;
* город;
* улица;
* дом;
* индекс.

Аналогично создадим класс «Warehouse» (см. рисунок 3.2).



***Рис. 3.2 – Класс «Warehouse»***

Для модели дополнительно указывается индекс, состоящий из трех столбцов: City, Street, House. При этом индекс должен быть уникальным, что исключает возможность дублирования складов с одинаковым адресом. Устанавливается максимальная длина для City (64 символа), Street (64 символа), House (8 символов).

**3.2.3 Создание модели поставщиков**

Атрибуты таблицы «Поставщики»:

* код поставщика;
* название организации поставщика.

Представим таблицу поставщиков в виде класса «Supplier» на C# (см. рисунок 3.3).



***Рис. 3.3 – Класс «Supplier»***

На свойство «Name» наложено ограничение: максимальная длина равна 128. Кроме того, наименование организации должно быть уникальным.

**3.2.4 Создание модели категорий комплектующих**

Атрибуты таблицы «Категории комплектующих»:

* код категории комплектующих;
* название категории комплектующих.

Представим таблицу категорий комплектующих в виде класса «ComponentType» на C# (см. рисунок 3.4).



***Рис. 3.4 – Класс «ComponentType»***

На свойство «Name» наложено ограничение: максимальная длина равна 64. Кроме того, название категории должно быть уникальным.

**3.2.5 Создание модели комплектующих**

Атрибуты таблицы «Комплектующие»:

* код комплектующего;
* название модели комплектующего;
* гарантия комплектующего;
* код категории комплектующего.

Представим таблицу комплектующих в виде класса «Component» на C# (см. рисунок 3.5).



***Рис. 3.5 – Класс «Component»***

На свойство «Name» наложено ограничение на максимальное число символов – 256. Кроме того, из данного класса можно обратиться к характеристикам, заказам, конфигурациям, поставкам и информации о хранении заданного комплектующего.

**3.2.6 Создание модели типов характеристик**

Атрибуты таблицы «Типы характеристик»:

* код типа характеристики;
* название типа характеристики.

Представим таблицу типов характеристик в виде класса «DetailType» на C# (см. рисунок 3.6).



***Рис. 3.6 – Класс «DetailType»***

На «Name» наложено ограничение – значения данного свойства должны быть уникальными.

**3.2.7 Создание модели характеристик**

Атрибуты таблицы «Характеристики»:

* код характеристики;
* код типа характеристики;
* код комплектующего;
* значение характеристики.

Представим таблицу характеристик в виде класса «ComponentDetail» на C# (см. рисунок 3.7).



***Рис. 3.7 – Класс «ComponentDetail»***

На значения данного класса наложено два ограничения. Первое ограничение – пара значений «DetailTypeId» и «ComponentId» должна быть уникальной, второе – максимальная длина значения характеристики не должна превышать 128 символов.

**3.2.8 Создание модели учета комплектующих на складах**

Атрибуты таблицы «Учет комплектующих на складах»:

* код хранения комплектующего на складе;
* цена комплектующего;
* количество комплектующего;
* код склада;
* код комплектующего.

Представим таблицу учета комплектующих на складах в виде класса «ComponentStorage» на C# (см. рисунок 3.8).



***Рис 3.8 – Класс «ComponentStorage»***

Из данного класса можно обратиться к комплектующему и складу, на котором он хранится.

**3.2.9 Создание модели поставок**

Атрибуты таблицы «Поставки»:

* код поставки;
* закупочная цена комплектующего;
* количество комплектующего;
* дата поставки;
* код поставщика;
* код склада;
* код комплектующего.

Представим таблицу поставок в виде класса «Supply» на C# (см. рисунок 3.9).



***Рис. 3.9 – Класс «Supply»***

Таблица «Поставки» связана с таблицами «Поставщики», «Комплектующие», «Склады» отношением многие-к-одному. Поэтому к ним можно обратиться напрямую из класса «Supply» через свойства «Supplier», «Component» и «Warehouse».

**3.2.10 Создание модели заказов**

Атрибуты таблицы «Заказы»:

* код заказа;
* дата заказа;
* статус заказа.

Представим таблицу заказов в виде класса «Order» на C# (см. рисунок 3.10).



***Рис. 3.10 – Класс «Order»***

К составу заказа можно обратиться через свойство «OrderComponents», так как таблицы «Order» и «OrderComponents» связаны отношением один-ко-многим.

**3.2.11 Создание модели составов заказов**

Атрибуты таблицы «Состав заказов»:

* код состава заказа;
* цена комплектующего;
* количество комплектующего;
* часть конфигурации;
* код заказа;
* код склада;
* код комплектующего.

Представим таблицу состава заказов в виде класса «OrderComponent» на C# (см. рисунок 3.11).



***Рис. 3.11 – Класс «OrderComponent»***

Через данный класс можно получить информацию о комплектующем, выбранном складе, заказе через соответствующие свойства «Component», «Warehouse» и «Order».

**3.2.12 Создание модели конфигураций**

Атрибуты таблицы «Конфигурации»:

* код части конфигурации;
* код конфигурации;
* количество комплектующего;
* код комплектующего.

Представим таблицу конфигураций в виде класса «Configuration» на C# (см. рисунок 3.12).



***Рис. 3.12 – Класс «Configuration»***

В классе «Configuration» пара значений «ConfigurationId» и «ComponentId» должна быть уникальной.

**3.3 Подключение приложения к базе данных**

Взаимодействие программы с базой данных реализуется через библиотеку Entity Framework. В разделе 3.2 были рассмотрены модели, которые представлены в виде таблиц в СУБД, но для возможности работы с ними необходимо определить класс «StoreContext», который наследуется от «DbContext», коллекции моделей и данные для подключения.

Каждая коллекция имеет тип DbSet<T>, где T – название класса модели (например, User или Component). В конструкторе вызывается метод «EnsureCreated», который используется для создания базы данных, связанных таблиц и других объектов хранения. В методе «OnConfiguring» вызывается функция «UseNpgsql» с указанием данных для авторизации в СУБД. Код класса «StoreContext» приведен в приложении.

**3.4 Реализация интерфейса**

Здесь будет текст…