МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное   
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

(Самарский университет)

Институт информатики, математики и электроники

Факультет информатики  
Кафедра суперкомпьютеров и общей информатики

**Отчет по лабораторной работе № 5**

Дисциплина: «Enterprise Systems Development (Методы проектирования и поддержки требований к программному обеспечению)»

Вариант 25

Тип предприятия: **«Music store chain»**

Выполнил: Андреев А.Ю.

Группа: 6133-010402D

Самара 2023

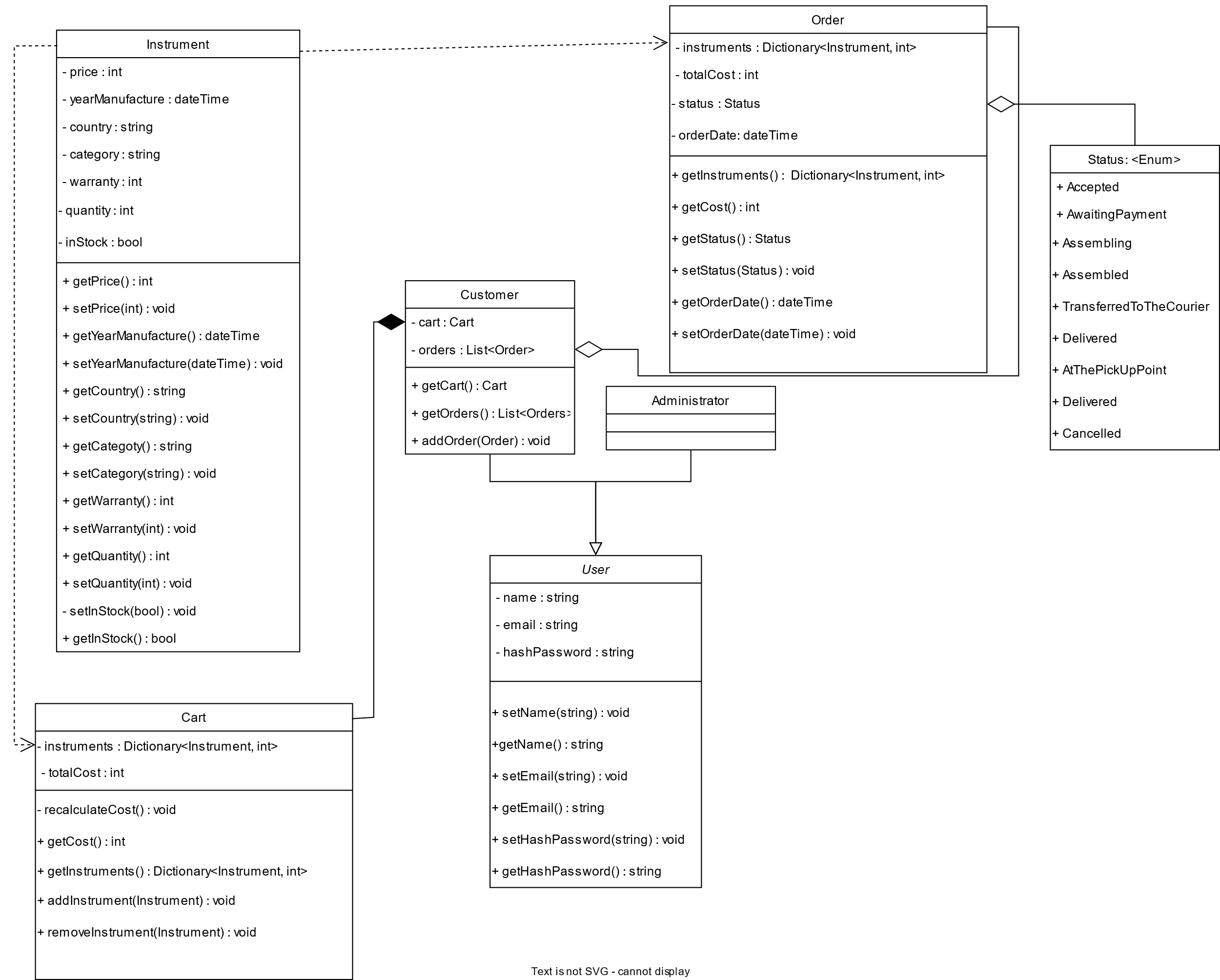


Рисунок 1 – Диаграмма классов

Таблица 1 – Пояснения атрибутов и методов

|  |  |  |
| --- | --- | --- |
| Сlass name | Attribute / Method | Data type / Comment about method assignment |
| User | name | string |
| email | string |
| hashPassword | string |
| setName(string) | Метод устанавливает имя пользователя |
| getName() | Метод считывает имя пользователя |
| setEmail(string) | Метод устанавливает Email пользователя |
| getEmail() | Метод считывает Email пользователя |
| setHashPassword(string) | Метод рассчитывает хэш пароля и устанавливает его |
| getHashPassword () | Метод считывает хэш пароля для авторизации пользователя |
| Instrument | price | int |
| yearManufacture | dateTime (пишу на C#, там есть такой тип) |
| country | string (возможно, было бы логичнее сделать Enum, но значений в теории очень много) |
| category | string (тип товара, возможно, было логичнее сделать Enum, но значений в теории очень много) |
| warranty | int (гарантия в месяцах) |
| quantity | int |
| inStock | bool (поле для быстрой проверки наличия товара, связано с quantity) |
| getPrice() | Метод считывает цену |
| setPrice(int) | Метод устанавливает цену |
| getYearManufacture() | Метод считывает год выпуска |
| setYearManufacture(dateTime) | Метод устанавливает год выпуска |
| getCountry() | Метод считывает страну производства |
| setCounty(string) | Метод устанавливает страну производства |
| getCategory() | Метод считывает категорию инструмента |
| setCategory(string) | Метод устанавливает категорию инструмента |
| getWarranty() | Метод считывает гарантию инструмента |
| setWarranty(int) | Метод устанавливает гарантию инструмента |
| getQuantity() | Метод считывает количество товара на складе |
| setQuantity(int) | Метод устанавливает количество товара на складе |
| setInStick(bool) | Метод устанавливает флаг: если товар есть на складе. Планируется использоваться только в методе setQuantuty(int) |
| getInStock() | Метод считывает флаг: если товар есть на складе. |
| Cart | instruments | Dictionary<Instrument, int> |
| totalCost | int |
| recalculateCost() | Метод для пересчета стоимости корзины, должен работать при добавлении/удалении инструментов в корзину |
| getCost() | Метод получает стоимость корзины |
| getInstruments() | Метод для получения корзины клиента |
| addInstrument(Instrument) | Метод добавляет инструмент в Dictionary<Instrument, int> или увеличивает его количество |
| removeInstrument(Instrument) | Метод удаляет инструмент из Dictionary<Instrument, int> или уменьшает его количество |
| Customer | cart | Cart |
| orders | List<Order> |
| getCart() | Метод возвращает корзину покупателя |
| getOrders() | Метод возвращает список заказов пользователя |
| addOrder(Order) | Метод добавляет заказ в список заказов пользователя |
| Order | instruments | Dictionary<Instrument, int> |
| totalCost | int |
| status | Status |
| orderDate | dateTime |
| getInstruments() | Метод возвращает список и количество купленных инструментов |
| getCost() | Метод возвращает стоимость заказа |
| getStatus() | Метод возвращает статус заказа |
| setStatus(Status) | Метод устанавливает статус заказа |
| getOrderDate() | Метод возвращает дату заказа |
| setOrderDate() | Метод устанавливает дату заказа |

Пояснение бизнес-логики.

1. Мне кажется, что роли Администратор и Покупатель лучше разделить по классам, а не хранить роль пользователя полем, потому что так будет удобнее определять пользователя (для редактирования статусов/инструментов и т.д.) и довольно странно давать возможность админу покупать товары. Для этого должна быть отдельная учетная запись.
2. Менять стоимость заказа и инструменты/количество инструментов в заказе тоже нелогично, так как весь список целиком переходит из корзины Покупателя.
3. HashPassword у юзера безопаснее хранить, чем сам пароль.
4. Стоимость корзины должна пересчитываться сразу после добавления/удаления/изменения количества товара, поэтому метод приватный, фактически это метод setCost().
5. Также не имеет никакого смысла устанавливать корзину и список покупок у покупателя, так как корзина может только наполняться, а ее содержимое переходить в заказ, который добавляется в список заказов.

P.S. Конечно, к каждому классу можно придраться и предложить различную логику, заложив различные сценарии использования, например: удаление пользователя, то есть необходимость флага активности пользователя; добавление методов setCart(Cart) и setOrders(List<Orders>), если будет необходимость «восстановить» аккаунт пользователя/создать новый на основе другого (возможно, архивного) и т.д.; добавление методов setCost(int), recalculateCost() в заказ, чтобы изменять заказ по мере наличия на складе инструментов.

Но любое проектирование зависит от конкретной бизнес-задачи, а я в своих предыдущих работах такие требования не выдвигал, поэтому такая модель хорошо подходит для моей задачи.