

## Техническое задание на разработку (тестовое задание на должность C#-разработчик)

### 1. Разработать модели данных основных сущностей:

1.1. **Product** – модель товара, состоящая из следующих обязательных свойств:

- **Id** – Идентификатор
- **Name** – Название
- **Price** – стоимость
- Расширение модели другими свойствами (необходимыми с точки зрения разработчика) – на усмотрение разработчика

1.2. **SalesPoint** – точка продажи товаров, состоящая из следующих обязательных свойств:

- **Id** – Идентификатор
- **Name** – Название
- **ProvidedProducts** – список сущностей (ProvidedProduct) доступных к продаже товаров, с текущим доступным количеством по каждому товару. Сущность ProvidedProduct имеет следующие обязательные свойства:
  - **ProductId** – идентификатор продукта
  - **ProductQuantity** – количество
- Расширение модели другими свойствами (необходимыми с точки зрения разработчика) – на усмотрение разработчика

1.3. **Buyer** – покупатель, лицо, осуществляющее покупку товара или услуги в одной из точек продаж

- **Id** – Идентификатор
- **Name** – Имя
- **SalesIds** – коллекция всех идентификаторов покупок, когда-либо осуществляемых данным покупателем
- Расширение модели другими свойствами (необходимыми с точки зрения разработчика) – на усмотрение разработчика

1.4. **Sale** – Акт продажи, состоящий из следующих обязательных свойств

- **Id** – Идентификатор
- **Date** – дата осуществления продажи
- **Time** – время осуществления продажи
- **SalesPointId** – идентификатор точки продажи
- **BuyerId** – идентификатор покупателя (**Can be null**)
- **SalesData** – список сущностей SaleData, содержащей в себе следующие свойства:
  - **ProductId** – идентификатор купленного продукта
  - **ProductQuantity** – количество штук купленных продуктов данного ProductId
  - **ProductIdAmount** – общая стоимость купленного количества товаров данного ProductId
- **TotalAmount** – общая сумма всей покупки

2. Разработать web API с реализацией CRUD операций с базой данных над всеми моделями описанными в п.1
3. Разработать web API с реализацией следующей бизнес-логики:
  - 3.1. **Sale** – продажа товара или услуги. Товар можно приобрести в любой точке продажи, при условии наличия в ней необходимого количества товара. Покупка товаров доступна как для авторизованных пользователей (имеющих Id), так и для неавторизованных пользователей.
    - При осуществлении операции продажи товаров для авторизованных пользователей, выполняются соответствующие записи в базу данных:
      - I. Изменяется количество доступных товаров в точке продажи, согласно количеству проданных товаров
      - II. Формируется экземпляр сущности **Sale** (п. 1.4), и записывается в базу данных
      - III. Идентификатор сущности (сформированной в ч. II.) Sale записывается в коллекцию покупок в таблице покупателя
    - Для неавторизованных пользователей действия аналогичны, за исключением ч. III. – в этом случае никаких записей в таблицы покупателя не производится, а свойство BuyerId в сущности Sale (п. 1.4) имеет значение null
4. Реализовать функцию автоматического наполнения базы данных тестовыми значениями при запуске приложения.
5. Реализовать функционал OpenAPI (swagger) для всех API
6. Функция логирования – не является обязательным требованием, но будет плюсом при рассмотрении задания
7. Используемый стек:
  - 7.1. **.NET 5**
  - 7.2. **Entity Framework Core** code first (InMemory db)
8. При выполнении задания следует ориентироваться на принципы **DRY, SOLID, IoC DI**. Наличие **Unit-тестов** в каждом проекте не является обязательным, но будет **жирным плюсом** при рассмотрении задания.
9. Выполненные работы принимаются в виде ссылки на публичный репозиторий в GitHub. При необходимости, в репозиторий необходимо поместить файл run.txt, с описанием необходимых действий для сборки и запуска решения.