**Інтерфейс IProduct:**

Властивості:

* Ім’я
* Ціна
* Вага

Методи:

* Зміна ціни – void ChangePrice(double percent)

**Інтерфейс IMeat:**

Властивості:

* Категорія м’яса
* Сорт м’яса

**Інтерфейс IDairyProduct:**

Властивості:

* Строк придатності

**Інтерфейс IStorage:**

Властивості:

* Список продуктів (List<Product>);

Методи:

* Зміна ціни – void ChangePrice(double percent)
* Додавання продукту – void AddProduct(Product product)

Клас ***Product*** реалізує інтерфейс **IProduct**.

Клас ***Meat*** реалізує інтерфейс **IMeat**, клас ***DairyProduct*** реалізує інтерфейс **IDairyProduct**.

Класи ***Meat*** і ***DairyProduct*** є нащадками класу ***Product***.

Клас ***Storage*** реалізує інтерфейс **IStorage**.

Сервіс **StorageService** (реалізує інтерфейс IEnumerable) створений для пошуку товарів та виконання певних дій над двома складами. Усі методи статичні:

* Об’єднання товарів на двох складах – IEnumerable<Product> GetUnionProducts(Storage storage1, Storage storage2)
* Пошук товарів на першому складу яких немає на другому – IEnumerable<Product> GetExceptProducts(Storage storage1, Storage storage2)
* Пошук спільних товарів на двох складах – IEnumerable<Product> GetIntersectProducts(Storage storage, Storage storage2)
* Пошук по назві продукту – List<Product> SearchByName(Storage storage, string name)
* Пошук по категорії – List<Product> SearchByName(Storage storage, string name)
* Пошук по ціні – List<Product> SearchByName(Storage storage, string name)

Також є два допоміжних класи для читання продуктів з файлу та запису продуктів в файл:

* **WriterStorage** – приймає об’єкт типу Storage та записує всі товари зі складу у файл
* **ReaderStorage** – повертає об’єкт типу Storage, попередньо виконуючи парсинг рядків.