

Bookshop: | Product Admin, | Product Module  
administration on fragen

Bookshop (| Persistence)  
wie soll abgespeichert werden  
JSON, Datenbank?

administration:

Insert Book(Book): void

Delete Books(): void

List Books(): List

Delete Book(string isbn): void

Add Author(string isbn, string Author): void

by: | Product Admin, | Product Module

by (| Persistence persistence)

bei uns 1 Persistence EF (Datenbank)

administration:

Mitglied registrieren(Member, int VertragID)

Mitglied löschen (int MemberID)

List Mitglieder()

Delete Mitglieder() Vlt auch nicht

change Vertrag von Mitglied (int MemberID, int VertragID)

Vertrag Erstellen (Vertrag)

Vertrag Löschen (int VertragID)

List Verträge()

Delete Verträge() vlt auch nicht

Mitarbeiter Registrieren (Mitarbeiter)

Mitarbeiter Löschen (int MitarbeiterID)

List Mitarbeiter()

Delete Mitarbeiter() vlt auch nicht

Artikel Einfügen Automat (Artikel)

Artikel Kaufen Automat (int ArtikelID, int MemberID) → Order Einfügen

List Artikel Automat()

Delete All Artikel Automat() vlt auch nicht

Artikel Einfügen Liste (Artikel)

Artikel Löschen Liste (int ArtikelID)

List Artikel Liste()

Delete All Artikel Liste()

Order Einfügen (Order)

Order Löschen (int OrderID): void

List Orders()

Order Löschen()

Add Mitglied To Order (int OrderID, int MitgliedID)

Get All Orders From (int MitgliedID): List<Orders>

anfragen:

Search Mitglied (string searchTerm): IDictionary<Mitglied, int>

Get Mitglied Details (int MitgliedID): Mitglied

Search Mitarbeiter (string searchTerm): IDictionary<Mitarbeiter, int>

Get Mitarbeiter Details (int MitarbeiterID): Mitarbeiter

Search Vertrag (string searchTerm): IDictionary<Vertrag, int>

Get Vertrag Details (int VertragID): Vertrag

Search Artikel Automat (string searchTerm): IDictionary<Artikel, int>

Get Artikel Automat Quantity (int ArtikelID): int

anfragen:

Search Book (string searchTerm): IDictionary<Book, int>

Get Book Details (int BuchID): Book

Get Quantity (int isbn): int

`Search ArtikelListe(String searchTerm): IDictionary<Artikel, int>` } Einkaufsliste  
`Search Order(String searchTerm): IDictionary<Order, int>`  
`Get OrderDetails(int orderID): Order`

`Get ArtikelDetails(int ArtikelID): Artikel` → Allgemein

Weiterleitung an Persistence! Da ist der richtige Code! Aber bei Search Lokale Speicherung und Suche

Hier in DD Speichern!

Book Persistence EF: | Book Persistence  
was sollte Speicherstelle können

`UpdateBook(Book, string author): void`

`CreateBook(Book): void`

`DeleteBook(int isbn): void`

`DeleteBooks(): void`

`FindBook(int isbn): Book`

`FindAllBooks(): IList<Book>`

Gym Persistence EF: | Gym Persistence

`UpdateMitglied(Mitglied, int VertragID): void`

`CreateMitglied(Mitglied): void`

`DeleteMitglied(Mitglied): void`

`DeleteMitglieder(): void`

`FindMitglied(int MitgliedID): Mitglied`

`FindMitglieder(): IList<Mitglied>`

`CreateVertrag(Vertrag): void`

`DeleteVertrag(int VertragID): void` überprüft Mitglieder ob wer Vertrag hat  
↳ sonst exception  
↳ damit gehen

`DeleteVerträge(): void`

`FindVertrag(int VertragID): Vertrag`

`FindAllVerträge(): IList<Vertrag>`

`CreateMitarbeiter(Mitarbeiter): void`

`DeleteMitarbeiter(int MitarbeiterID): void`

`DeleteAllMitarbeiter(): void`

`FindMitarbeiter(int MitarbeiterID): Mitarbeiter`

`FindAllMitarbeiter(): IList<Mitarbeiter>`

`CreateArtikelAutomat(Artikel): void`

`DeleteArtikelAutomat(int ArtikelID): void`

`DeleteArtikelAutomat(): void`

`FindArtikelAutomat(int ArtikelID): Artikel`

`FindAllArtikelAutomat(): IList<Artikel>`

`CreateArtikelListe(Artikel): void`

`DeleteArtikelListe(int ArtikelID): void`

un 1 Artikel Anzahl runter bis löschen  
} Automat

? Liste

Delete ArtikelListe(): void  
 Find ArtikelListe( int ArtikelID): Artikel  
 Find All ArtikelListe(): /List<Artikel>

} Liste

Create Order( Order): void  
 Delete Order( int OrderID): void  
 Delete All Orders(): void  
 Find Order( int OrderID): Order  
 Find All Orders(): /List<Order>

Update OrderMitglied( Order, int MitgliedID): void

→ Mitglied speichert Vertrag bei seiner Tabelle

→ Order speichert MitgliedID bei sich!

→ ArtikelAutomat ist eigene Tabelle mit ArtikelID und  
 Menge (target stock, actual stock), preis  
 ↑  
 muss in Liste enthalten sein

→ ArtikelListe ist die Liste mit allen Artikel

