Data

FileOrderRepository: IOrderRepository

- _filepath: string
- _filepathWithoutExtension: string
- _header: string

+<<constructor>>FileOrderRepository(filepath: string)

+ LoadOrders(orderDate: DateTime): List<Order>
+ LoadOrder(orderNumber: int, orderDate: DateTime)
+ AddOrder(order: Order, orderDate: DateTime): Order
+ EditOrder(order: Order, orderDate: DateTime)
+ DeleteOrder(orderNumber: int, orderDate: DateTime)

- CreateFilepath(orderDate: DateTime): string
- CreateCsvForOrder(order: Order): string
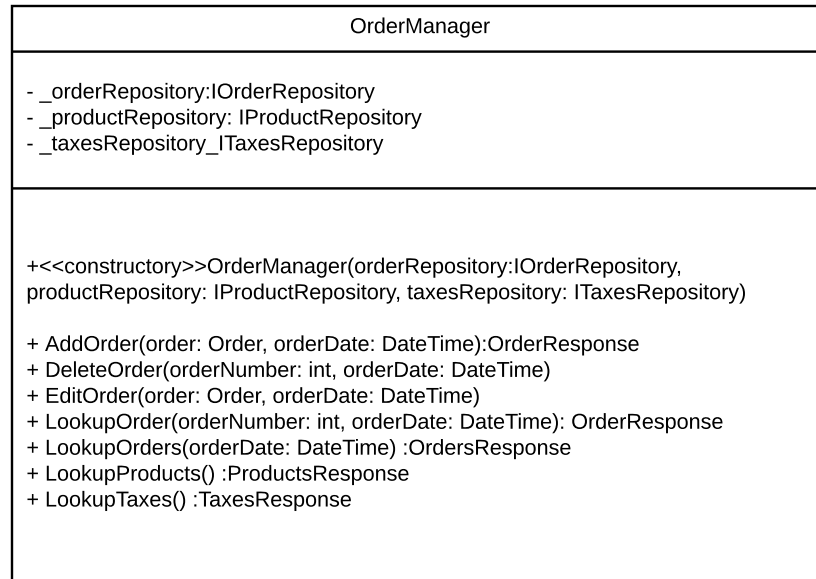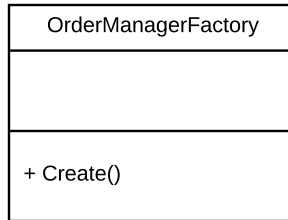- CreateOrderFile(orders: List<Order>)

OrderTestRepository: IOrderRepository

- _order: Order
- _order2: Order
- _orderDate: DateTime

+ LoadOrders(orderDate: DateTime): List<Order>
+ LoadOrder(orderNumber: int, orderDate: DateTime): Order
+ AddOrder(order: Order, orderDate: DateTime): Order
+ EditOrder(order: Order, orderDate: DateTime)
+ DeleteOrder(orderNumber: int, orderDate: DateTime)

FileTaxesRepository: ITaxesRepository

- _filepath:string

+<<constructor>>FileTaxesRepository(filepath: string)

+ LoadTaxes(): List<Tax>
+ LoadTax(stateAbbreviation: string): Tax

TaxesTestRepository: ITaxesRepository

- _tax: string
- _tax2: string
- _tax3: string

+ LoadTaxes(): List<Tax>
+ LoadTax(stateAbbreviation: string): Tax

FileProductRepository: IProductRepository

- _filepath:string

+<<constructor>>FileProductRepository(filepath: string)

+ LoadProducts(): List<Product>

ProductTestRepository: IProductRepository

- _product:string
- _product2: string
- _product3: string

+ LoadProducts(): List<Product>

```
┌─────────────────────────────────────┐
│          OrderManagerFactory         │
├─────────────────────────────────────┤
│                                      │
│                                      │
├─────────────────────────────────────┤
│                                      │
│  + Create()                          │
│                                      │
└─────────────────────────────────────┘
```

```
┌──────────────────────────────────────────────────────────────────────┐
│                            OrderManager                                │
├──────────────────────────────────────────────────────────────────────┤
│  - _orderRepository:IOrderRepository                                   │
│  - _productRepository: IProductRepository                              │
│  - _taxesRepository_ITaxesRepository                                   │
│                                                                        │
├──────────────────────────────────────────────────────────────────────┤
│                                                                        │
│                                                                        │
│  +<<constructory>>OrderManager(orderRepository:IOrderRepository,       │
│  productRepository: IProductRepository, taxesRepository: ITaxesRepository) │
│                                                                        │
│  + AddOrder(order: Order, orderDate: DateTime):OrderResponse           │
│  + DeleteOrder(orderNumber: int, orderDate: DateTime)                  │
│  + EditOrder(order: Order, orderDate: DateTime)                        │
│  + LookupOrder(orderNumber: int, orderDate: DateTime): OrderResponse   │
│  + LookupOrders(orderDate: DateTime) :OrdersResponse                   │
│  + LookupProducts() :ProductsResponse                                  │
│  + LookupTaxes() :TaxesResponse                                        │
│                                                                        │
│                                                                        │
└──────────────────────────────────────────────────────────────────────┘
```

UI

Menu

+ Start()

OrderAddWorkflow

+ Execute()

OrderDeleteWorkflow

+ Execute()

OrderEditWorkflow

+ Execute()

OrderLookupWorkflow

+ Execute()

ConsoleIO

+ SeparatorBar:string
+ OrderDisplayFormat:string

+ DisplayOrders(orders: List<Order>)
+ GetOptionalStringFromUser(prompt: string, customerName: string): string
+ GetRequiredIntFromUser(prompt: string: int
+ DisplayOrder(order: Order)
+ DisplayProducts(products: List<Product>)
+ DisplayTaxRates(taxes: List<Tax>)
+ GetRequiredAreaFromUser(prompt: string): decimal
+ GetOptionalAreaFromUser(prompt: string, area: decimal): decimal
+ GetRequiredStringFromUser(prompt: string): string
+ GetRequiredDecimalFromUser(prompt: string): decimal
+ GetRequiredFutureDateFromUser(prompt: string): DateTime
+ GetRequiredDateFromUser(prompt: string): DateTime
+ GetYesNoAnswerFromUser(prompt: string): string
+ GetOptionalIntFromUser(prompt: string): int

OrderFileTest

- _filepath: string
- _originalData: string
- _filepathWithoutExtension: string

+ Setup()
+ CanAddOrderFileData
+ CanEditOrderFileData
+ CanLoadOrderFromList
+ CanLoadOrdersFileData
+ CanDeleteOrderFileData
+ CannotLoadOrderNumberNotFound
+ CannotLoadOrdersByDateNotFound

OrderTest

+ CanAddOrderToListTest
+ CanEditOrder
+ CannotLoadDateNotFoundTest
+ CanLoadOrderTestData
+ OrderCalculationsTest

ProductFileTest

+ CanLoadProductFileData

ProductTest

+ CanLoadProductTestData

TaxFileTest

+ CanLoadTaxFileData

TaxesTest

+ CanLoadTaxTestData

Order

Key	OrderNumber	int
	CustomerName	string
	State	string
	TaxRate	decimal
	ProductType	string
	Area	decimal
	CostPerSquareFoot	decimal
	LaborCostPerSquareFoot	decimal
	MaterialCost	decimal(Area * CostPerSquareFoot)
	LaborCost	decimal(Area * LaborCostPerSquareFoot)
	Tax	decimal((MaterialCost + LaborCost) * (TaxRate/100))
	Total	decimal(MaterialCost + LaborCost + Tax)
Key	OrderDate	DateTime

Product

Key	ProductType	String
	CostPerSquareFoot	Decimal
	LaborCostPerSquareFoot	Decimal

Tax

Key	StateAbbreviation	String
	StateName	String
	TaxRate	Decimal

Response

+ Success: bool
+ Message: string

OrderResponse

+ Success: bool
+ Message: string
+ order: Order

OrdersResponse

+ Success: bool
+ Message: string
+ Orders: List<Order>

ProductsResponse

+ Success: bool
+ Message: string
+ Products: List<Product>

TaxesResponse

+ Success: bool
+ Message: string
+ Taxes: List<Tax>

ITaxesRepository

LoadTax(stateAbbreviation: string): Tax
LoadTaxes(): List<Tax>

IOrderRepository

LoadOrders(orderDate: DateTime): List<Order>
LoadOrder(orderNumber: int, orderDate: DateTime)
AddOrder(order: Order, orderDate: DateTime): Order
EditOrder(order: Order, orderDate: DateTime)
DeleteOrder(orderNumber: int, orderDate: DateTime)

IProductRepository

LoadProducts(orderDate: DateTime): List<Product>