

CRUD приложение без ORM



Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>





CREATE



READ



UPDATE



DELETE

C

R

U

D

CRUD операции

Какво са CRUD операциите?

- **CRUD** – Създаване **/Create/** - пример: добавяне на нов запис в таблицата
- **CRUD** – Прочитане **/Read/** - пример: прочитане на запис от таблицата
- **CRUD** – Променяне **/Update/** - пример: промяна на един или няколко елемента от запис в таблицата
- **CRUD** – Изтриване **/Delete/** - пример: изтриване на запис в таблицата

Архитектура на приложението

- Ще създадем **CRUD** приложение, което ще управлява **продукти**
- Трислойна архитектура /**3-tier architecture**/
 - Слой за данни /**Data access layer**/
 - Бизнес слой /**Business layer**/
 - Презентационен слой /**Presentation layer**/

Структура на проекта

- Business
 - **ProductBusiness.cs** – описва бизнес логиката свързана с продуктите
- Common – общ (споделен) слой между всички останали
 - **Product.cs** – описва класа данни за продукт
- Data
 - **Database.cs** – статичен клас за връзка с базата данни
 - **ProductData.cs** – клас, в който ще изпълняваме заявки към БД
- Presentation
 - **Display.cs** – клас, в който ще реализираме конзолно управление на приложението
 - **Program.cs** – Място, където ще създадем обект от класа Display

База данни

- Създайте локална БД shop в SQL Server и таблица **product**:

```
create table product(  
    Id int identity(100, 1) primary key,  
    Name varchar(100),  
    Price decimal(10,2),  
    Stock Int  
);
```

Data/Database.cs

- Създайте статичен клас **Database.cs**
 - Създайте частно статично свойство за низа за връзката
 - Създайте статичен метод **GetConnection()**, който връща **SqlConnection** обект

Data/ProductData.cs

- Създайте следните методи:
 - `public List<Product> GetAll()` – този метод връща списък с всички продукти в таблицата
 - `public Product Get(int id)` – този метод връща запис от таблицата според подадено id
 - `public void Add(Product product)` – този метод добавя запис в таблицата
 - `public void Update(Product product)` – този метод редактира запис в таблицата
 - `public void Delete(int id)` – този метод изтрива запис в таблицата по подадено id

Data/ProductData.cs (2)

- Във всеки от методите трябва да изпълните SQL команди:
 - `public List<Product> GetAll()`
 - `INSERT INTO product (Name, Price, Stock) VALUES (@name, @price, @stock)`
 - `public Product Get(int id)`
 - `SELECT * FROM product where Id = @id`
 - `public void Add(Product product)`
 - `INSERT INTO product (Name, Price, Stock) VALUES (@name, @price, @stock)`
 - `public void Update(Product product)`
 - `UPDATE product SET Name = @name, Price = @price, Stock = @stock WHERE Id = @id`
 - `public void Delete(int id)`
 - `DELETE product WHERE Id = @id`

Common/Product.cs

- В споделения клас Product описваме свойства, които съответстват на колоните от таблицата в БД. Добавяме и конструктори:

```
public int Id { get; set; }
public string Name { get; set; }
public decimal Price { get; set; }
public int Stock { get; set; }
public Product() { } //конструктор по подразбиране

public Product(int id, string name, decimal price, int stock){
    this.Id = id; //TODO: Добавете и останалите
}
```

Business/ProductBusiness.cs

- Създаваме управляващо поле от клас ProductData:

```
private ProductData manager = new ProductData();
```

- Създаваме методи, които извършват желаните операции, извиквайки методи от ProductData:

```
public List<Product> GetAll()  
{  
    return manager.GetAll();  
}
```

```
public Product Get(int id)  
{  
    return manager.Get(id);  
}
```

- Реализирайте по сходен начин останалите методи в този клас.

Реализиране на презентационен слой

- В рамките на класа Display.cs реализирайте презентационен слой, който да предлага различни операции на потребителя:

```
-----  
MENU  
-----  
1. List all entries  
2. Add new entry  
3. Update entry  
4. Fetch entry by ID  
5. Delete entry by ID  
6. Exit
```

Реализиране на презентационен слой (2)

- `private ProductBusiness productBusiness = new ProductBusiness()` – създава се обект от бизнес класа, който се използва за извикване на съответната бизнес логика
- `private void ShowMenu()` – метод, който визуализира възможните операции за избор
- `private void Input()` – метод, който въвежда желаната операция от потребителя и извиква някой от останалите методи, реализиращи отделните операции

Реализиране на презентационен слой (3)

- `void ListAll()` – визуализира всички данни
- `void Add()` – въвежда информация за продукт и я предава за съхранение
- `void Update()` – приема id, въвежда информация за него и предава за съхранение променената информация
- `void Fetch()` – извлича информация за продукт по id
- `void Delete()` – изтрива информация за продукт по id

CRUD приложение без ORM



Въпроси?

Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

