

МИНИСТЕРСТВО НА ОБРАЗОВАНИЕТО И НАУКАТА

**ПРОФЕСИОНАЛНА ГИМНАЗИЯ “ГЕН. ВЛАДИМИР ЗАИМОВ” гр. СОПОТ**

4330 гр. Сопот, ул. ”Иван Вазов” №1, тел./факс: /03134/ 83-31, 83-32, e-mail: [pgzaimov@yahoo.com](mailto:pgzaimov@yahoo.com)

**ПРОЕКТ**

**Заглавие на проекта:**

**Доставка на храна**

**Задача № 8**

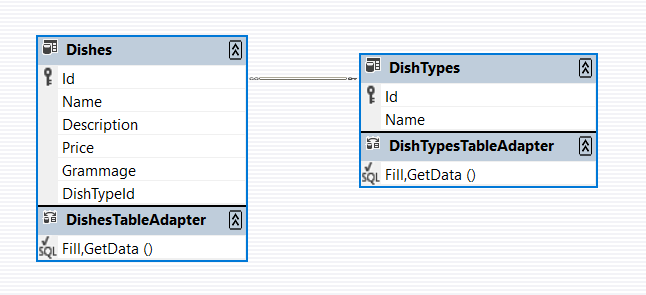
*Ученици: Кристиан Вълков ; Божидар Мутавчийски ; Василена-Мария Минева*

***Професия:*** *„Системен програмист“*

***Специалност:*** *„Системно програмиране“*

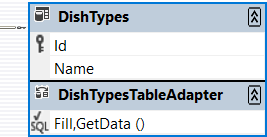
Сопот, 2023 г.

**Създаване на връзки 1:М**



*E/R diagram база данни DELIVERY.MDF*

*1.Описание на таблиците с релациите между тях*

**

public class DishType

{

public int Id { get; set; }

public string Name { get; set; }

// 1:M

public ICollection<Dish> Dishes { get; set; }

}

public class Dish

{

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

public double Price { get; set; }

public double Grammage { get; set; }

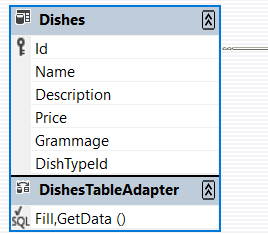
// M:1

public int DishTypeId { get; set; } // Forein key

public DishType DishTypes { get; set; } // Таблица ,

с която се осъществява връзката

}

**

*2.Създаване на DBContext*

*2.1 Описание на компонентните таблици*

public class DishesDbContext: DbContext

{

public DishesDbContext() : base("DishesDbContext")

{

}

// Създаване редовете на таблица Dishes

public DbSet<Dish> Dishes { get; set; }

// Създаване редовете на таблица DishTypes

public DbSet<DishType> DishTypes { get; set; }

}

*2.2. Инсталиране EntityFramework*

*Когато наследим DbContext се предизвиква инсталиране на пакета Entity Framework, който съдържа всички необходими инструменти за работа с база от данни: описание на таблици и релации, изпълнение на заявки, менажиране на данни и тяхната сигурност.*

*3.* *Конфигуриране на БД с описанието – добавяме секцията за конфигуриране на връзката във файла App.config от структурата на приложението // DELIVERY е името на БД, която ще се създаде в папката C:\Users\името на USER*

<connectionStrings>

<add name="DishesDbContext"

connectionString="Data Source=(localdb)\MSSQLLocalDB;

Initial Catalog=Delivery;

Integrated Security=true"

providerName="System.Data.SqlClient" />

</connectionStrings>

*4. Миграции – изпълняваме последователно командите:*

PM> Enable-Migrations

PM> Add-migration ИМЕ на миграцията

PM> Update-database

**ГЛАВА 2.**

**Функционално описание: Отделните екрани с кратко описание за работа с приложението.**

**ГЛАВА 3.**

**Анализ на задачата**

Тук опишете:

* Кратко каква технология сте използвали за връзка с БД;
* Как работи всеки метод /код с обяснения – като примерите по-долу/
* Подходът си за избор на контроли и интерфейс и/или друго /отново екрани и шорткътове/

**Описание на CRUD операциите- особености на отделните методи**

**в каре**

// GET: Orders/Details/5

public async Task<IActionResult> Details(int? id)

{

if (id == null)

{

return NotFound();

}

var order = await \_context.Order

.Include(o => o.User)

.FirstOrDefaultAsync(m => m.Id == id);

if (order == null)

{

return NotFound();

}

return View(order);

}

private void btnFind\_Click(object sender, EventArgs e)

{

int findId = 0;

if (string.IsNullOrEmpty(textBox1.Text) || !textBox1.Text.All(char.IsDigit))

{

MessageBox.Show("Въведете Id за търсене!");

textBox1.BackColor = Color.Red;

textBox1.Focus();

return;

}

else

{

findId = int.Parse(textBox1.Text);

}

Dog findedDog = dogsController.Get(findId);

if (findedDog == null)

{

MessageBox.Show("НЯМА ТАКЪВ ЗАПИС в БД! \n Въведете Id за търсене!");

textBox1.BackColor = Color.Red;

textBox1.Focus();

return;

}