



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

ПРОЕКТ

Заглавие на проекта:
Управление на ястия в заведение за хранене
Задача №8

Ученик: Михаела Костадинова Костова

Професия: „Системен програмист“

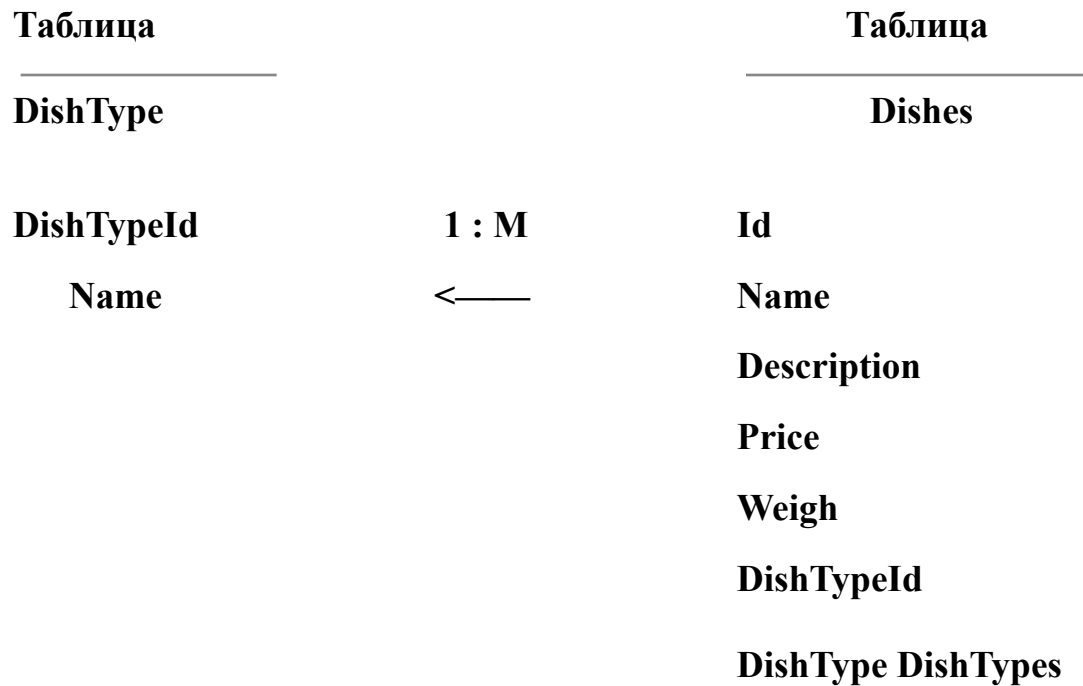
Специалност: „Системно програмиране“

Сопот, 2025 г.

Глава 1.

Описание на БД

1. Диаграма на таблиците



Таблицата DishType съдържа описание на всички типове ястия като:
“Салати”, ”Предястие”, ”Основно ястие”, ”Супа”, ”Аламинут” и “Десерт”.

Таблицата Dish съдържа информацията и описанието на всяко ястие,
име на ястието, описание, цена, грамаж и тип (супа, салата, аламинут...)

Връзката между DishType и Dish е 1 към много (1 тип с много ястия)

RestaurantContext е основният клас в Entity Framework, който управлява връзката с базата данни и достъпа до таблиците чрез свойства от тип **DbSet<Dish/DishType>**. Контекстът съдържа две таблици: **Dish** и **DishType**

ГЛАВА 2

Функционално описание

- **Форма за управление на ястия:**
 - **Падащ списък (ComboBox) за избор на тип ястие, зарежда всички типове от базата, таблицата: DishType).**
 - **Текстови полета за въвеждане на име, описание, цена и грамаж.**
 - **Бутон „Добави“ за създаване на ново ястие.**
 - **Бутон „Редактирай“ за промяна на избрано ястие.**
 - **Бутон „Изтрий“ за премахване на избрано ястие.**
 - **ListBox, който визуализира списък с всички ястия (номер и име на ястието, тип, цена и грамаж.).**

ГЛАВА 3.

Анализ на задачата

1.Технология за връзка с базата:

Проектът използва Entity Framework с подхода Code First, което позволява лесно дефиниране на класовете и автоматично генериране на базата данни и таблиците. Това намалява ръчното писане на SQL и улеснява миграциите.

2.Работа на методите:

За тип на ястие съм избрала **ComboBox**, защото позволява избор само от валидни стойности, което предотвратява въвеждането на невалидни типове.

```
private void cmbDishType_SelectedIndexChanged(object sender, EventArgs e)
{
    using (var context = new RestaurantContext())
    {
        var types = context.DishTypes.ToList();
        cmbDishType.DataSource = types;
        cmbDishType.DisplayMember = "Name";
        cmbDishType.ValueMember = "DishTypeId";
    }
}
```

За визуализация на данните съм използвала **ListBox**, който предоставя удобен начин за показване на списък с ястия, с възможност за избор и редакция.

За показване на информация за съществуващо ястие на екрана, напр. при търсене или редакция, съм използвала **LoadRecord()**.

```
private void LoadRecord(Dish dish)
{
    txtName.Text = dish.Name;
    txtDesc.Text = dish.Description.ToString();
    txtPrice.Text = dish.Price.ToString(); ;
    txtWeigh.Text = dish.Weight.ToString();
    cmbDishType.SelectedValue = dish.DishTypeId;
}
```

За да се нулира или изтрие записа на формата съм използвала **ClearScreen()**

```
private void ClearScreen()
{
    txtName.Clear();
    txtDesc.Clear();
    txtPrice.Clear();
    txtWeigh.Clear();
    cmbDishType.Text = "";
}
```

За зареждане на цяла форма съм използвала **Form1_Load()**

```
private void Form1_Load(object sender, EventArgs e)
{
    List<DishType> allDishType = dishtypeControler.GetAllDishTypes();
    cmbDishType.DataSource = allDishType;
    cmbDishType.DisplayMember = "Name";
    cmbDishType.ValueMember = "Id";

    SelectAll(sender, e, dishControler);
}
```

За да обнова списъка с ястията съм използвала SelectAll()

```
private void SelectAll(object sender, EventArgs e, DishControler
dishControler)
{
    List<Dish> allDish = dishControler.GetAll();
    listBox1.Items.Clear();
    foreach (var item in allDish)
    {
        listBox1.Items.Add($"{item.Id}. {item.Name} - {item.Description}.
{item.Price}. {item.Weight}. Тип на ястието:- { item.DishTypes.Name}");
    }
}
```

3.CRUD операция:

Create (Добавяне): Въвеждане на ново ястие с валидиране на входните данни.

```
private void btnAdd_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(txtName.Text) || txtName.Text == "")
    {
        MessageBox.Show("Въведете данни!");
        txtName.Focus();
        return;
    }
    Dish dishes = new Dish();
    dishes.Name = txtName.Text;
    dishes.Description = txtDesc.Text;
    dishes.Price = int.Parse(txtPrice.Text);
    dishes.Weight = int.Parse(txtWeigh.Text);
    dishes.DishTypeId = (int)cmbDishType.SelectedValue;
    dishControler.Create(dishes);
    MessageBox.Show("Записът е успешно добавен!");
    ClearScreen();
    SelectAll(sender, e, dishControler);
}
```

Update (Редакция): Промяна на съществуващо ястие.


```
private void btnEdit_Click(object sender, EventArgs e)
{
    int findId = 0;
    if (string.IsNullOrEmpty(txtName.Text) ||
txtName.Text.All(char.IsDigit))
    {
        MessageBox.Show("Въведете номер за търсене!");
        txtName.BackColor = Color.Red;
        txtName.Focus();
        return;
    }
    else
    {
        findId = int.Parse(txtName.Text);
    }
    if (string.IsNullOrEmpty(txtDesc.Text))
    {
        Dish findeddish = dishControler.Get(findId);
        if (findeddish == null)
        {
            MessageBox.Show("няма такова ястие!");
            txtName.Focus();
            return;
        }
        LoadRecord(findeddish);
    }
    else
    {
        Dish updatedish = new Dish();
        updatedish.Name = txtName.Text;
        updatedish.Description = txtDesc.Text;
        updatedish.Price = int.Parse(txtPrice.Text);
        updatedish.Weight = int.Parse(txtWeigh.Text);
        updatedish.DishTypeId = (int)cmbDishType.SelectedValue;
        dishControler.Update(findId, updatedish);
    }
    SelectAll(sender, e, dishControler);
}
```

Delete (Изтриване): Изтриване на избрано ястие от базата.

```
private void btnDelete_Click_1(object sender, EventArgs e)
{
    int findId = 0;
    if (string.IsNullOrEmpty(txtName.Text) || txtName.Text.All(char.IsDigit))
    {
        MessageBox.Show("Въведете номер за търсене!");
        txtName.Focus();
        return;
    }
    else
    {
        findId = int.Parse(txtName.Text);
    }
    Dish findeddish = dishControler.Get(findId);
    if (findeddish == null)
    {
        MessageBox.Show("няма такова ястие !");
        txtName.Focus();
        return;
    }
    LoadRecord(findeddish);

    DialogResult answer = MessageBox.Show("Наистина ли искате да
изтриете запис No " + findId + "?", "PROMPT",
MessageBoxButtons.YesNo);
    if (answer == DialogResult.Yes)
    {
        dishControler.Delete(findId);
    }

    SelectAll(sender, e, dishControler);
}
```

