**Задача №7 на Христо Стефанов Фънев, 11д клас, № 24**

Parcels

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **Name** | **Description** | **Price** | **Weight** | **ParcelTypeId** |
| 1 | Мишка | Компютърен аксесоар | 30 | 1,2 | 1 |
| 2 | Диван | Мебел | 250 | 50 | 5 |
| 3 | Натурален акт | Документ за притежание на имот | 25 | 0,5 | 2 |

ParcelTypes

|  |  |
| --- | --- |
| **Id** | **Name** |
| 1 | Малък пакет |
| 2 | Документи |
| 3 | Едрогабаритен |
| 4 | Писмо |
| 5 | Непалитизиран |

|  |  |
| --- | --- |
| ParcelTypes |  |
| Id | int |
| Name | string |

|  |  |
| --- | --- |
| Parcels |  |
| Id | int |
| Name | string |
| Description | string |
| Price | decimal |
| Weight | decimal |
| ParcelTypeId | int |

Променливи за работа с отделните бутони в формата:

ParcelBusiness parcelBusiness = new ParcelBusiness();

ParcelTypeBusiness parcelTypeBusiness = new ParcelTypeBusiness();

Метод за свързване на формата с бизнеса (контролера) на задачата:

private void LoadRecord(Parcel parcel)

{

idTextBox.BackColor = Color.White;

idTextBox.Text = parcel.Id.ToString();

nameTextBox.Text = parcel.Name;

descriptionTextBox.Text = parcel.Description;

priceTextBox.Text = parcel.Price.ToString();

weightTextBox.Text = parcel.Weight.ToString();

parcelTypeComboBox.Text = parcel.ParcelType.Name;

}

Метод за изчистване на формата след извършване на дадена операция:

private void ClearScreen()

{

idTextBox.BackColor = Color.White;

idTextBox.Clear();

nameTextBox.Clear();

descriptionTextBox.Clear();

priceTextBox.Clear();

weightTextBox.Clear();

parcelTypeComboBox.Text = "";

}

Начин на зареждане на формата при стартиране:

private void Form1\_Load(object sender, EventArgs e)

{

List<ParcelType> allTypes = parcelTypeBusiness.GetAllTypes();

parcelTypeComboBox.DataSource = allTypes;

parcelTypeComboBox.DisplayMember = "Name";

parcelTypeComboBox.ValueMember = "Id";

}

Метод на бутон за добавяне на елемент в базата данни:

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

{

if (nameTextBox.Text == "" || string.IsNullOrEmpty(nameTextBox.Text))

{

MessageBox.Show("Въведете данни моля!");

return;

}

Parcel parcel = new Parcel();

parcel.Name = nameTextBox.Text;

parcel.Description = descriptionTextBox.Text;

parcel.Price = decimal.Parse(priceTextBox.Text);

parcel.Weight = decimal.Parse(weightTextBox.Text);

parcel.ParcelTypeId = (int)parcelTypeComboBox.SelectedValue;

parcelBusiness.Add(parcel);

ClearScreen();

}

Метод за бутон за намиране на даден елемент в базата данни по дадено ид (Номер):

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

{

int id = 0;

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

{

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

idTextBox.BackColor = Color.Red;

return;

}

else

{

id = int.Parse(idTextBox.Text);

}

Parcel fndParcel = parcelBusiness.Get(id);

if (fndParcel == null)

{

MessageBox.Show("Няма такъв запис!");

idTextBox.BackColor = Color.Red;

return;

}

LoadRecord(fndParcel);

}

Метод за бутон за променяне на стойностите на вече създаден елемент в базата данни:

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

{

int id = 0;

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

{

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

idTextBox.BackColor = Color.Red;

return;

}

else

{

id = int.Parse(idTextBox.Text);

}

if (string.IsNullOrEmpty(nameTextBox.Text))

{

Parcel updParcel = parcelBusiness.Get(id);

if (updParcel == null)

{

MessageBox.Show("Няма такъв запис!");

idTextBox.BackColor = Color.Red;

return;

}

LoadRecord(updParcel);

}

else

{

Parcel parcel = new Parcel();

parcel.Name = nameTextBox.Text;

parcel.Description = descriptionTextBox.Text;

parcel.Price = decimal.Parse(priceTextBox.Text);

parcel.Weight = decimal.Parse(weightTextBox.Text);

parcel.ParcelTypeId = (int)parcelTypeComboBox.SelectedValue;

parcelBusiness.Update(id, parcel);

}

}

Метод за бутон за изкарване на всички вкарани елементи в базата данни:

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

{

List<Parcel> parcels = parcelBusiness.GetAll();

parcelListBox.Items.Clear();

foreach (var item in parcels)

{

parcelListBox.Items.Add($"{item.Id} {item.Name} {item.Description} {item.Price}лв. {item.Weight}кг. Вид пратка: {item.ParcelType.Name}");

}

}

Метод за бутон за изтриване на даден елемент от базата данни по дадено ид (Номер):

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

{

int id = 0;

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

{

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

idTextBox.BackColor = Color.Red;

return;

}

else

{

id = int.Parse(idTextBox.Text);

}

Parcel delParcel = parcelBusiness.Get(id);

if (delParcel == null)

{

MessageBox.Show("Няма такъв запис!");

idTextBox.BackColor = Color.Red;

return;

}

LoadRecord(delParcel);

DialogResult question = MessageBox.Show("Искате ли да изтриете записа?", "Изтриване", MessageBoxButtons.YesNo);

if (question == DialogResult.Yes)

{

parcelBusiness.Delete(id);

}

}

В изпълнението на Задача №7 съм използвал CRUD (Create, Read, Update, Delete) операции, както и технологията ORM (Code First). Също се използва и пакета Entity Framework и миграции за работа с базата данни и поддръжката им.