

**Технически Университет – Варна**

Факултет по Изчислителна Техника и Автоматизация

Софтуерни и Интернет Технологии

Проект по „Технология на Софтуерното Производство“

Тема: „Таксиметрова Компания“

Изготвили:

|  |  |
| --- | --- |
| Име | Факултетен Номер |
| Никола Георгиев Дечев | 20621508 |
| Дейвид Емилов Мераков | 20621505 |

Проверил:\_\_\_\_\_\_\_\_\_\_\_\_\_\_

/доц. В.Божикова/

**Задание 1**

1. Създаване на таблици Cars (коли) и Orders (поръчки).

В таблицата Cars да се съхранява информация за: KodTaxi – уникален номер на такси; RegNomer – регистрационен номер; Mark – марка на колата; Seats - брой места за пътници; Luggage – място за голям багаж (Да/Не); Driver – име на шофьора.

В таблицата Orders да се съхранява информация за: NumOrd - номер на поръчката; KodTaxi - номер на такси; Address – адрес, от който е направена поръчката; OrdTime – време (дата и час) на поръчката; Distance – изминато разстояние (км), Fare - такса на поръчката (лв.).

При създаването на таблиците изберете подходящ тип данни и други свойства на полетата.

Ограничете таксата и изминатото разстояние да са положителни числа, а подразбиращата се стойност

на времето на поръчката да бъде текущото време. Броят на местата да бъде от 3 до 10. Изберете правилни ключови полета. Свържете таблиците с подходящи релации (връзки).

2. В таблицата Cars попълнете поне 5 записа със смислени данни на български и на кирилица, а в

Orders -10.

3. Създайте обобщена параметрична заявка qryTotalOrders, базирана на таблиците Cars и Orders, която да показва за поръчките, направени преди дата, зададена чрез параметър:

* регистрационния номер и марката на таксито;
* общата сума на поръчките за всяко такси;

4. Създайте меню лента за информационната система с три точки:

* Въвеждане - да стартира формулярите
* Справки – да стартира отчетите
* Изход – да затваря програмата

5. Създайте следните формуляри

* frmCars, базиран на таблицата Cars.
* Orders subform, базиран на таблицата Orders.
* във формуляр frmCars създайте подформуляр, базиран на Orders subform, така че когато в главния формуляр се избере такси, в подформуляра да се показват само изпълнените
* от него поръчки.

Етикетите на управляващите елементи да бъдат на български и на кирилица.

6. Създайте 2 отчета, базирани на таблицата Orders и на параметричната заявка. Надписите в отчетите да бъдат на български и на кирилица.

7. Може да използвате БД по избор.

8. Проекта трябва да бъде задължително разпечат и оформен добре, да има концептуален, физически и логически модел на БД, както и use и class диаграми.

**Проектиране**

За разработката на проекта (системата) са използвани:

* C# – за разработката на системата
* Access DB 2016 – за съхранение на данните
* SQL – за изпълнение на заявки

Използвана е База от Данни с името **TaxiManagmentApplication**, състояща се от 2 (две) таблици:

* Cars
  + KodTaxi – уникален номер на такси
  + RegNomer – регистрационен номер
  + Mark – марка на колата
  + Seats - брой места за пътници
  + Luggage – място за голям багаж (Да/Не)
  + Driver – име на шофьора
* Orders
  + NumOrd - номер на поръчката
  + KodTaxi - номер на такси
  + Address – адрес, от който е направена поръчката
  + OrdTime – време (дата и час) на поръчката
  + Distance – изминато разстояние (км)
  + Fare - такса на поръчката (лв.)

**Описание на проекта**

1. Класове
   1. *Cars.cs* – съдържа get / set за полетата

**Програмен код:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace TaxiManagmentApplication

{

internal class Cars

{

public string KodTaxi { get; set; }

public string RegNomer { get; set; }

public string Mark { get; set; }

public string Seats { get; set; }

public bool Luggage { get; set; }

public string Driver { get; set; }

}

}

* 1. *Orders.cs* – съдържа get / set за полетата

**Програмен код:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace TaxiManagmentApplication

{

internal class Orders

{

public string NumOrd { get; set; }

public string Address { get; set; }

public string OrdTime { get; set; }

public string Distance { get; set; }

public string Fare { get; set; }

public string KodTaxi { get; set; }

}

}

* 1. *Connection.cs* – съдържа логиката за добавяне на данни в Базата от Данни

**Програмен код:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.OleDb;

using TaxiManagmentApplication;

namespace TaxiCompany

{

internal class Connection

{

OleDbConnection connect;

OleDbCommand command;

private void ConnectionTo()

{

connect = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=E:\University\3-ти курс\5-ти семестър\Технология на Софтуерното Производство\Проект\vs2022 проект\TaxiManagmentApplication.accdb");

command = connect.CreateCommand();

}

public Connection()

{

ConnectionTo();

}

public void InsertCars(Cars cars)

{

try

{

command.CommandText =

"Insert into Cars(KodTaxi, RegNomer, Mark, Seats, Luggage, Driver) values (" + cars.KodTaxi + ",'"

+ cars.RegNomer + "','" + cars.Mark + "',"

+ cars.Seats + "," + cars.Luggage + ",'"

+ cars.Driver + "')";

command.CommandType = System.Data.CommandType.Text;

connect.Open();

command.ExecuteNonQuery();

System.Windows.Forms.MessageBox.Show("Успешно добавихте такси.");

}

catch (Exception e)

{

System.Windows.Forms.MessageBox.Show(e.Message);

}

finally

{

if (connect != null)

{

connect.Close();

}

}

}

public void InsertOrders(Orders orders)

{

try

{

command.CommandText =

"Insert into Orders(NumOrd, KodTaxi, Address, OrdTime, Distance, Fare) values (" + orders.NumOrd + ","

+ orders.KodTaxi + ",'" + orders.Address + "','"

+ orders.OrdTime + "'," + orders.Distance + ","

+ orders.Fare + ")";

command.CommandType = System.Data.CommandType.Text;

connect.Open();

command.ExecuteNonQuery();

System.Windows.Forms.MessageBox.Show("Успешно добавихте поръчка.");

}

catch (Exception e)

{

System.Windows.Forms.MessageBox.Show(e.Message);

}

finally

{

if (connect != null)

{

connect.Close();

}

}

}

}

}

* 1. *Program.cs* – основната входна точка за приложението

**Програмен код:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace TaxiManagmentApplication

{

internal static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Main());

}

}

}

1. Форми
   1. *Main.cs –* главната форма на приложението. Съдържа меню с осталите функционалности на приложението:
      * Меню
        + Изход
      * Въвеждане
        + Добавяне на таксита
        + Добаване на поръчки
      * Справки
        + Регистрационния номер и марката на таксито
        + Общата сума на поръчките за всяко такси

**Програмен код:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace TaxiManagmentApplication

{

public partial class Main : Form

{

public Main()

{

InitializeComponent();

}

private void изходToolStripMenuItem1\_Click(object sender, EventArgs e)

{

this.Close();

}

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

{

AddCars addCars = new AddCars();

addCars.Show();

}

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

{

AddOrders addOrders = new AddOrders();

addOrders.Show();

}

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

{

MakeNRegNumcs makeNRegNumcs = new MakeNRegNumcs();

makeNRegNumcs.Show();

}

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

{

SumByTaxiOrders sumByTaxiOrders = new SumByTaxiOrders();

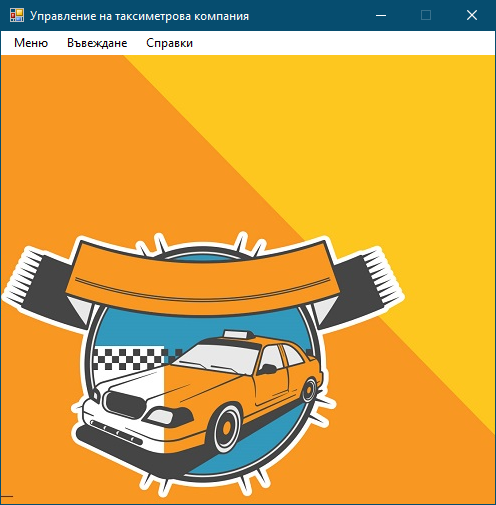
sumByTaxiOrders.Show();

}

}

}

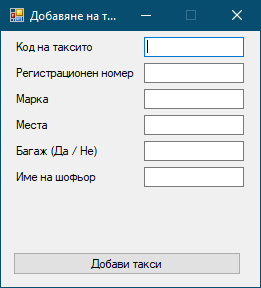
**Екранна снимка:**

****

* 1. *AddCars.cs* – добавя таксита чрез полетата във формата:
     + Код на такси – уникален код на таксито
     + Регистрационен номер
     + Марка
     + Места
     + Багаж (Да / Не) – има ли място за багаж
     + Име на шофьор

След натискането на бутона „Добави такси“, данните за таксито се запазват в таблица **Cars**.

**Екранна снимка:**

****

**Програмен код:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using TaxiCompany;

namespace TaxiManagmentApplication

{

public partial class AddCars : Form

{

Connection connection = new Connection();

public AddCars()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Cars cars = new Cars();

/\*

if (textBox1.Text == "")

System.Windows.Forms.MessageBox.Show("Това поле не може да бъде празно!");

else

cars.KodTaxi = textBox1.Text;

if (textBox2.Text == "")

System.Windows.Forms.MessageBox.Show("Това поле не може да бъде празно!");

else

cars.RegNomer = textBox2.Text;

if (textBox3.Text == "")

System.Windows.Forms.MessageBox.Show("Това поле не може да бъде празно!");

else

cars.Mark = textBox3.Text;

if (int.Parse(textBox4.Text) >= 3 && int.Parse(textBox4.Text) <= 10)

cars.Seats = textBox4.Text;

else

MessageBox.Show("Местата могат да бъдат от 3 до 7.");

if (textBox5.Text.ToLower() == "да")

cars.Luggage = true;

else

cars.Luggage = false;

if (textBox6.Text == "")

System.Windows.Forms.MessageBox.Show("Това поле не може да бъде празно!");

else

cars.Driver = textBox6.Text;

\*/

cars.KodTaxi = textBox1.Text;

cars.RegNomer = textBox2.Text;

cars.Mark = textBox3.Text;

cars.Seats = textBox4.Text;

if (textBox5.Text.ToLower() == "да")

cars.Luggage = true;

else

cars.Luggage = false;

cars.Driver = textBox6.Text;

connection.InsertCars(cars);

}

private void button2\_Click(object sender, EventArgs e)

{

}

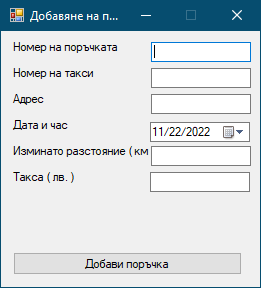
}

}

* 1. *AddOrders.cs* – добавя поръчки чрез полетата във формата:
     + Номер на поръчката
     + Номер на такси (уникален номер)
     + Адрес
     + Дата и час
     + Изминато разстояние
     + Такса

След натискане на бутона „Добави поръчка“, данните се добавят в таблица **Orders**.

**Екранна снимка:**

****

**Програмен код:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using TaxiCompany;

namespace TaxiManagmentApplication

{

public partial class AddOrders : Form

{

Connection connection = new Connection();

DateTime orderTime;

public AddOrders()

{

InitializeComponent();

//orderTime = DateTime.UtcNow("dd/mm/yyyy hh:mm:ss");

//textBox5.Text = orderTime.ToString();

}

// Insert

private void button1\_Click(object sender, EventArgs e)

{

Orders orders = new Orders();

orders.NumOrd = textBox1.Text;

orders.KodTaxi = textBox2.Text;

orders.Address = textBox3.Text;

orders.OrdTime = dateTimePicker1.Text;

orders.Distance = textBox6.Text;

orders.Fare = textBox7.Text;

connection.InsertOrders(orders);

}

private void dateTimePicker1\_ValueChanged(object sender, EventArgs e)

{

}

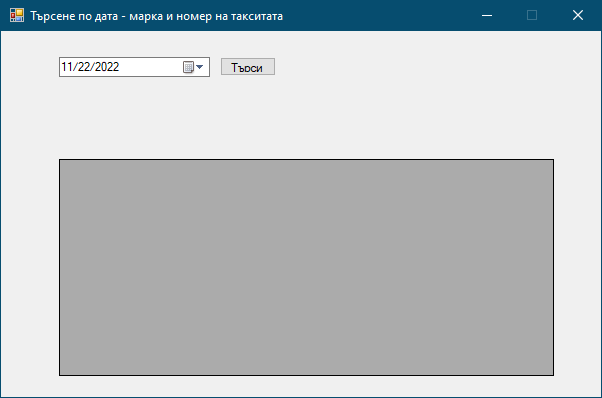
}

}

* 1. *MakeNRegNumcs.cs* – избираме дата от полето и се визуализират всички записи в таблиците, които имат дата преди зададената. Формата се състои от:
     + Форма за въвеждане на дата
     + Визуализатор на данни в табличен вид

След натискане на бутона „Търси“ се визуализират данните в табличен вид.

**Екранна снимка:**

****

**Програмен код:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using TaxiCompany;

namespace TaxiManagmentApplication

{

public partial class MakeNRegNumcs : Form

{

string conStr = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=E:\University\3-ти курс\5-ти семестър\Технология на Софтуерното Производство\Проект\vs2022 проект\TaxiManagmentApplication.accdb";

OleDbConnection dbConnect = new OleDbConnection();

public MakeNRegNumcs()

{

InitializeComponent();

}

void view(string mySelect)

{

dbConnect.ConnectionString = conStr;

dbConnect.Open();

OleDbDataAdapter adapt = new OleDbDataAdapter(mySelect, dbConnect);

DataTable dt = new DataTable();

adapt.Fill(dt);

dataGridView1.DataSource = dt;

dbConnect.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

string query =

"SELECT Cars.KodTaxi, Cars.RegNomer, Cars.Mark, Orders.OrdTime" +

" FROM Cars " +

"RIGHT JOIN Orders " +

"ON Cars.KodTaxi = Orders.KodTaxi " +

"WHERE(((Orders.OrdTime) <#" + dateTimePicker1.Text + "#))";

OleDbCommand cmd = new OleDbCommand(query, dbConnect);

dbConnect.ConnectionString = conStr;

dbConnect.Open();

cmd.ExecuteNonQuery();

dbConnect.Close();

view(query);

}

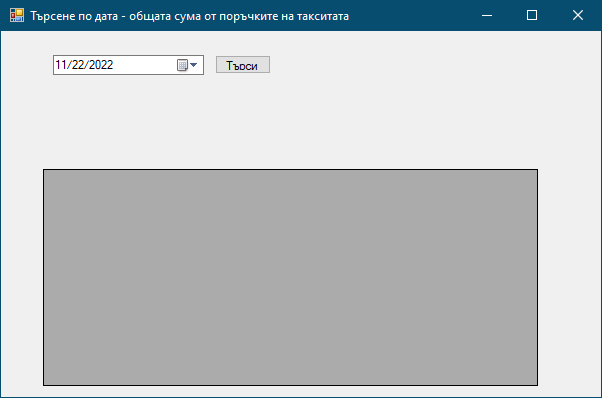
}

}

* 1. *SumByTaxiOrders.cs* – извежда общата сума за всички поръчки на всяко такси. Формата се състои от:
     + Форма за въвеждане на дата
     + Визуализатор на данни в табличен вид

След натискане на бутона „Търси“ се визуализират данните в табличен вид.

**Екранна снимка:**

****

**Програмен код:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace TaxiManagmentApplication

{

public partial class SumByTaxiOrders : Form

{

string conStr = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=E:\University\3-ти курс\5-ти семестър\Технология на Софтуерното Производство\Проект\vs2022 проект\TaxiManagmentApplication.accdb";

OleDbConnection dbConnect = new OleDbConnection();

public SumByTaxiOrders()

{

InitializeComponent();

}

void view(string mySelect)

{

dbConnect.ConnectionString = conStr;

dbConnect.Open();

OleDbDataAdapter adapt = new OleDbDataAdapter(mySelect, dbConnect);

DataTable dt = new DataTable();

adapt.Fill(dt);

dataGridView1.DataSource = dt;

dbConnect.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

string query =

"SELECT DISTINCTROW Cars.RegNomer, Sum(Orders.Fare) AS [Total] " +

"FROM Cars INNER JOIN Orders ON Cars.[KodTaxi] = Orders.[KodTaxi] " +

"GROUP BY Cars.RegNomer";

OleDbCommand cmd = new OleDbCommand(query, dbConnect);

dbConnect.ConnectionString = conStr;

dbConnect.Open();

cmd.ExecuteNonQuery();

dbConnect.Close();

view(query);

}

}

}