Министерство образования Российской Федерации Пензенский государственный университет

Кафедра «Математическое обеспечение и применение ЭВМ»

**ОТЧЕТ**

по лабораторной работе №2

по курсу «ООП» на тему «Использование коллекций. Делегаты и события»

Выполнили:

студенты группы 23ВП2 Соколов О.А.

Муравьев В.С.

Принял:

к.т.н. Афонин А.Ю.

Пенза 2025

### Название

Использование коллекций. Делегаты и события

### Цель работы

Научиться работать с коллекциями элементов платформы .Net. Научиться использовать делегаты и события.

### Лабораторное задание

В лабораторной работе требуется определить класс, содержащий типизированную коллекцию, который с помощью событий извещает об изменениях в коллекции. Для событий, извещающих об изменениях в коллекции, определяется пользовательский тип делегат. События регистрируются в специальных классах-слушателях.

Сформировать коллекцию элементов в соответствии с заданием. Организовать добавление, удаление элементов в коллекцию. Вывести содержимое коллекции элементов на экран пользователю.

Придерживаться предметной области предыдущих лабораторных работ. Реализовать события на добавление/удаление элементов в коллекцию. Зарегистрировать делегаты для этих событий.

Привести рабочий пример делегата с использованием синтаксиса лямбда выражений. Вариант коллекции, который необходимо реализовать – Stack. Коллекция должна состоять из экземпляров класса, созданного в рамках предыдущих лабораторных работ. Организовать вывод элементов коллекции с использованием оператора foreach.

Осуществить замер производительности на коллекции из своего варианта и Array. Предусмотреть автоматическое генерирование 100 000 элементов на вставку в коллекцию и осуществить тестовую выборку 100 000 элементов последовательно и в случайном порядке. Результаты замеров отобразить в виде ListView таблицы.

Предметная область – фирма грузоперевозок.

### Листинг

### Файл TransportCompany.cs:

using System;

using System.Collections.Generic;

using System.Drawing.Printing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab2

{

internal class TransportCompany

{

public int price { get; set; }

public float transportedMass { get; set; }

public string name { get; set; }

public float rating { get; set; }

public int completedOrders { get; set; }

public string phoneNumber { get; set; }

public string email { get; set; }

public static int countObj = 0;

public TransportCompany()

{

price = 0;

transportedMass = 0;

name = "";

rating = 0;

completedOrders = 0;

phoneNumber = "";

email = "";

countObj++;

}

public TransportCompany(int price) : this()

{

this.price = price;

}

public TransportCompany(int price, string name) : this(price)

{

this.name = name;

}

public TransportCompany(int price, float transportedMass, string name, float rating, int completedOrders, string phoneNumber, string email) : this(price, name)

{

this.transportedMass = transportedMass;

this.rating = rating;

this.completedOrders = completedOrders;

this.phoneNumber = phoneNumber;

this.email = email;

}

public override string ToString()

{

return "\tТрансопртная компания" + "\nНазвание: " + name + "\nЦена грузоперевозки: " + price + "\nМасса перевезенных грузов: " + transportedMass + "\nРейтинг: " + rating + "\nКоличество выполненных заказов: " + completedOrders + "\nНомер компании: " + phoneNumber + "\nПочта компании: " + email;

}

public string PrintName()

{

if (string.IsNullOrEmpty(name))

return "У фирмы нет названия";

return "Название фирмы: " + name;

}

public string PriceToHex()

{

return Convert.ToString(price, 16);

}

}

}

Файл MyException.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab1

{

internal class MyException : Exception

{

public MyException(string message) : base(message) { }

}

}

Файл StackTransportCompany.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab2

{

internal class StackTransportCompany

{

private Stack<TransportCompany> transportCompanies;

public event Action<TransportCompany> StackAdded;

public event Action StackRemoved;

public StackTransportCompany()

{

this.transportCompanies = new Stack<TransportCompany>();

}

public Stack<TransportCompany> GetTransportCompanies()

{

return transportCompanies;

}

public void AddCompany(TransportCompany company)

{

transportCompanies.Push(company);

StackAdded?.Invoke(company);

}

public void DeleteCompany()

{

if (transportCompanies.Count == 0)

throw new MyException("Стек пуст");

TransportCompany top = transportCompanies.Pop();

StackRemoved?.Invoke();

}

}

}

Файл StackListener.cs:

using System;

using System.Windows.Forms;

namespace Lab2

{

internal class StackListener

{

private ListView listView;

private TextBox objCount;

public StackListener(StackTransportCompany stack, ListView listView, TextBox objCount)

{

this.listView = listView;

this.objCount = objCount;

stack.StackAdded += OnCompanyAdded;

stack.StackRemoved += OnCompanyRemoved;

}

private void OnCompanyAdded(TransportCompany company)

{

var listItem = new ListViewItem(company.name);

listItem.SubItems.Add(company.price.ToString());

listItem.SubItems.Add(company.transportedMass.ToString());

listItem.SubItems.Add(company.rating.ToString());

listItem.SubItems.Add(company.completedOrders.ToString());

listItem.SubItems.Add(company.phoneNumber);

listItem.SubItems.Add(company.email);

listView.Items.Add(listItem);

objCount.Text = TransportCompany.countObj.ToString();

}

private void OnCompanyRemoved()

{

if (listView.Items.Count > 0)

listView.Items.RemoveAt(0);

objCount.Text = TransportCompany.countObj.ToString();

}

}

}

Файл Form1.cs:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Security.Policy;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Lab2

{

public partial class Form1 : Form

{

private StackTransportCompany companies;

private StackListener stackListener;

public Form1()

{

InitializeComponent();

companies = new StackTransportCompany();

InitializeListView();

//companies.StackAdded += OnCompanyAdded;

//companies.StackRemoved += () => objCount.Text = (--TransportCompany.countObj).ToString();

stackListener = new StackListener(companies, listView1, objCount);

}

private void InitializeListView()

{

listView1.Columns.Add("Название компании");

listView1.Columns.Add("Цена");

listView1.Columns.Add("Масса");

listView1.Columns.Add("Рейтинг");

listView1.Columns.Add("Количество заказов");

listView1.Columns.Add("Номер телефона");

listView1.Columns.Add("Почта");

listView1.View = View.Details;

}

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

{

try

{

if (string.IsNullOrWhiteSpace(name.Text))

throw new MyException("Фирма должна иметь название");

if (string.IsNullOrWhiteSpace(phoneNumber.Text))

throw new MyException("Фирма должна иметь номер");

if (!Regex.IsMatch(phoneNumber.Text.Trim(), @"^\d{11}$"))

throw new MyException("Номер должен состоять из 11 цифр и не содержать буквы или символы");

if (!Regex.IsMatch(email.Text.Trim(), @"^[a-zA-Z0-9\_]+@mail\.ru$"))

throw new MyException("Неверный формат почты");

TransportCompany firm = new TransportCompany((int)price.Value,

(float)transportedMass.Value,

name.Text,

(float)rating.Value,

(int)completedOrders.Value,

phoneNumber.Text,

email.Text);

objCount.Text = TransportCompany.countObj.ToString();

companies.AddCompany(firm);

}

catch (MyException ex)

{

MessageBox.Show(ex.Message, "Ошибка");

}

}

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

{

try

{

TransportCompany.countObj--;

companies.DeleteCompany();

}

catch (MyException ex)

{

MessageBox.Show(ex.Message, "Ошибка");

}

}

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

{

listView1.Items.Clear();

foreach(var company in companies.GetTransportCompanies())

{

var listItem = new ListViewItem(company.name);

listItem.SubItems.Add(company.price.ToString());

listItem.SubItems.Add(company.transportedMass.ToString());

listItem.SubItems.Add(company.rating.ToString());

listItem.SubItems.Add(company.completedOrders.ToString());

listItem.SubItems.Add(company.phoneNumber);

listItem.SubItems.Add(company.email);

listView1.Items.Insert(0, listItem);

}

}

}

}

### Пояснительный текст к программе

Класс TransportCompany представляет собой модель транспортной компании и содержит поля name (наименование компании), price (цена за грузоперевозку), rating (рейтинг), completedOrders (количество выполненных заказов), phoneNumber (номер телефона компании), email (электронная почта компании), transported mass (масса перевезенных грузов).

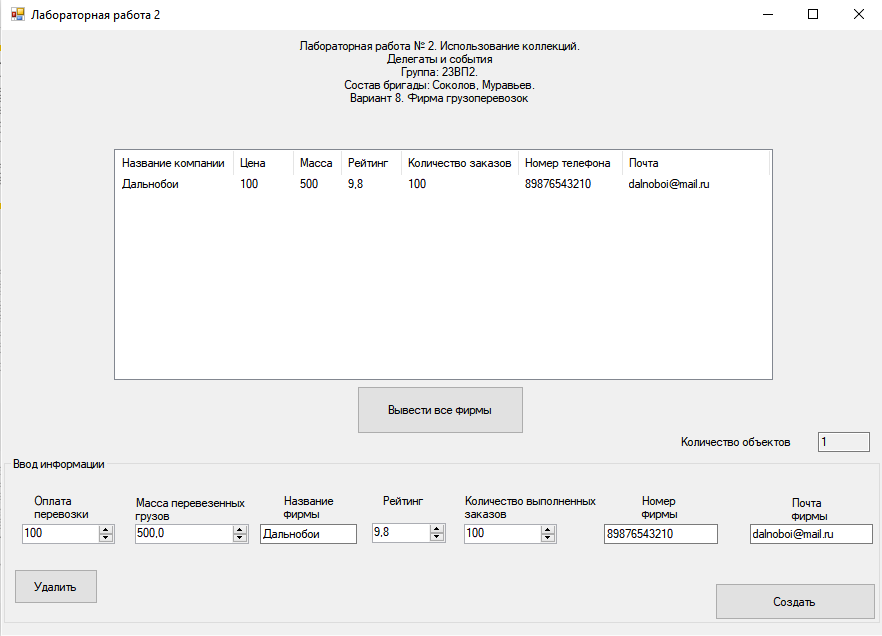
Класс MyException – это класс пользовательских исключений. Является наследником базового класса Exception и содержит информацию об ошибке, сгенерированной во время выполнения программы.

Класс StackTransportCompany является классом, хранящим стек из транспортных компаний. Осуществляет добавление элемента в стек и удаление элемента из стека.

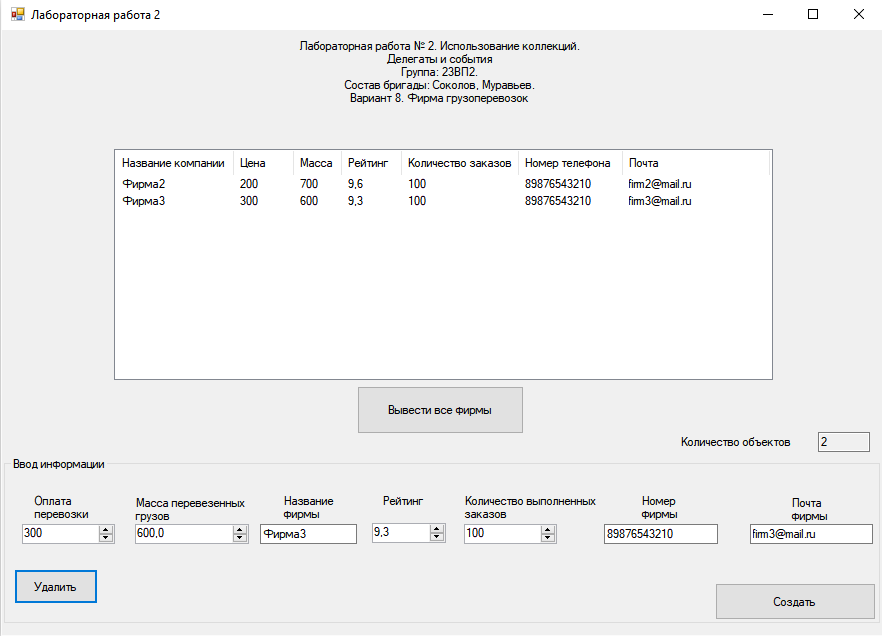
Класс StackListener – класс-слушатель, который обрабатывает события.

### Результаты работы программы

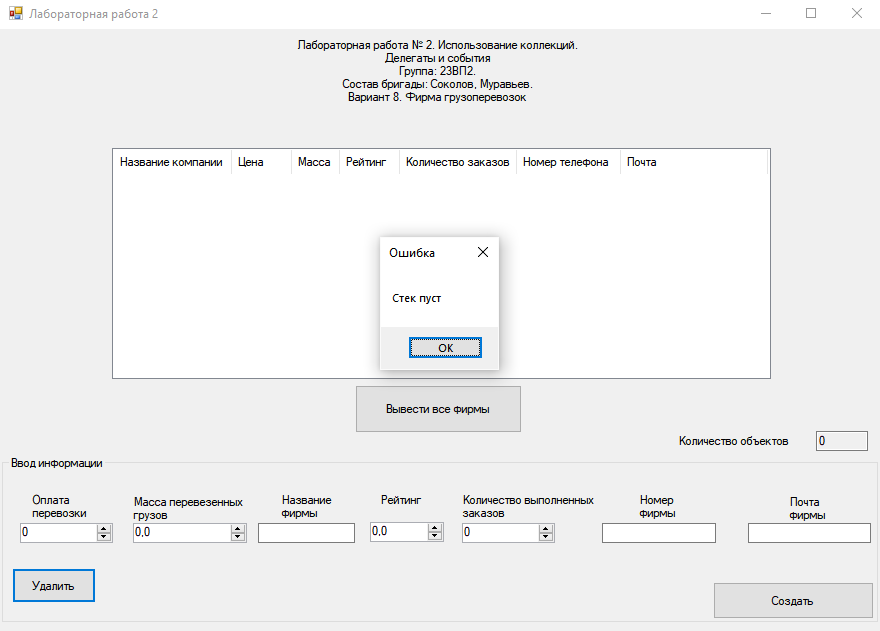
Результаты работы программы показаны на рисунках 1-3.



**Рисунок 1 – Создание объекта, добавление в стек и вывод информации**



**Рисунок 2 – Удаление фирмы из стека и вывод информации**



**Рисунок 3 – Попытка удаления элемента из пустого стека**

### Диаграмма вариантов использования

### C:\Users\Влад\Downloads\2025-02-25_01-28-01.png

**Рисунок 4 - Диаграмма вариантов использования**

### Выводы

В ходе выполнения лабораторной работы была реализована типизированная коллекция на основе структуры данных Stack, содержащая экземпляры класса, соответствующего предметной области фирмы грузоперевозок. В коллекции предусмотрены события на добавление и удаление элементов, а также разработаны классы-слушатели, регистрирующие данные события.

Были созданы делегаты для обработки событий, а также приведен пример использования лямбда-выражений для подписки на события. Реализованы методы добавления и удаления элементов, организован их вывод на экран с использованием оператора foreach.

Таким образом, лабораторная работа позволила закрепить навыки работы с коллекциями, делегатами, событиями и производительностью структур данных в C#.