**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

ФАИС

Кафедра «Информатика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

по дисциплине **«Языки программирования высокого уровня»**

на тему: **«Работа с файлами и каталогами»**

Выполнил: студент гр. ИП-22

Коваленко А. И.

Принял: ст. преподаватель Романькова Т. Л.

Дата сдачи отчета: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата допуска к защите: ­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Гомель 2022

**Коллекции структур данных. Классы прототипы**

**Цель**: *получить навыки разработки программ, использующих классы File, FileInfo, DirectoryInfo, FileSystemWatcher, FileStream, BinaryReader и BinaryWriter, научиться осуществлять сериализацию и десериализацию объектов.*

**Вариант 7.**

Разработать приложение, выполняющее следующие функции:

* Запись внесенной в таблицу информации о поездах в файл с применением сериализации в **двоичном** формате.

Структура записи в файл:

- *номер поезда*

*- пункт отправления*

*- пункт назначения*

*- цена плацкарты*

*- количество свободных мест.*

* Считывание информации из файла с применением десериализации и вывод в виде таблицы.
* Поиск поезда с максимальным количеством свободных мест, увеличение этой цены плацкарты на 20% и сохранение изменений в файле.
* Предоставление возможности смены рабочего каталога. ()

**Листинг программы:**

**Collection.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace llllrrrr4

{

class Collection

{

public List<Train> trains = new List<Train>();

public void Add(Train train)

{

trains.Add(train);

}

}

}

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace llllrrrr4

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Train2());

}

}

}

**Train.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace llllrrrr4

{

[Serializable]

class Train

{

private string numberOfTrain;

private string fromDestination;

private string toDestination;

private double price;

private int amountOfFreePlaces;

public Train()

{ }

public Train(string numberOfTrain, string fromDestination, string toDestination, double price, int countOfFreePlaces)

{

this.numberOfTrain = numberOfTrain;

this.fromDestination = fromDestination;

this.toDestination = toDestination;

Price = price;

this.amountOfFreePlaces = countOfFreePlaces;

}

public string NumberOfTrain { get => numberOfTrain; }

public string FromDestination { get => fromDestination; }

public string ToDestination { get => toDestination; }

public double Price { get => price; set { price = value; } }

public int AmountOfFreePlaces { get => amountOfFreePlaces; set { amountOfFreePlaces = value; } }

}

}

Листинг формы

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Runtime.Serialization.Formatters.Binary;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace llllrrrr4

{

public partial class Train2 : Form

{

public Train2()

{

InitializeComponent();

}

public DataGridViewCell this[int row, int column] { get => dataGridView1[row, column]; set { dataGridView1[row, column] = value; } }

private void Train2\_Load(object sender, EventArgs e)

{

numberTextBox.Text = "Номер поезда";

fromDestTextBox.Text = "Пункт отправления";

toDestTextBox.Text = "Пункт назначения";

priceTextBox.Text = "Цена за место (BYN)";

amountOfPlacesTextBox.Text = "Количество свободных мест";

}

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

{

dataGridView1.Rows.Clear();

}

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

{

Collection collection = new Collection();

BinaryFormatter formatter = new BinaryFormatter();

using (FileStream fs = new FileStream("train.dat", FileMode.OpenOrCreate))

{

collection.trains = (List<Train>)formatter.Deserialize(fs);

}

foreach (Train train in collection.trains)

{

dataGridView1.Rows.Add(train.NumberOfTrain, train.FromDestination, train.ToDestination, train.Price, train.AmountOfFreePlaces);

}

}

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

{

Collection collection = new Collection();

for (int i = 0; i < dataGridView1.RowCount - 1; i++)

{

collection.Add(new Train(

dataGridView1.Rows[i].Cells[0].Value.ToString(),

dataGridView1.Rows[i].Cells[1].Value.ToString(),

dataGridView1.Rows[i].Cells[2].Value.ToString(),

Convert.ToDouble(dataGridView1.Rows[i].Cells[3].Value.ToString()),

Convert.ToInt32(dataGridView1.Rows[i].Cells[4].Value.ToString())));

}

dataGridView1.Rows.Clear();

BinaryFormatter formatter = new BinaryFormatter();

using (FileStream fs = new FileStream("train.dat", FileMode.OpenOrCreate))

{

formatter.Serialize(fs, collection.trains);

}

}

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

{

Collection collection = new Collection();

for (int i = 0; i < dataGridView1.RowCount - 1; i++)

{

collection.Add(new Train(

dataGridView1.Rows[i].Cells[0].Value.ToString(),

dataGridView1.Rows[i].Cells[1].Value.ToString(),

dataGridView1.Rows[i].Cells[2].Value.ToString(),

Convert.ToDouble(dataGridView1.Rows[i].Cells[3].Value.ToString()),

Convert.ToInt32(dataGridView1.Rows[i].Cells[4].Value.ToString())));

}

dataGridView1.Rows.Clear();

Train tempTrain = collection.trains[0];

foreach (Train train1 in collection.trains)

{

if (train1.AmountOfFreePlaces > tempTrain.AmountOfFreePlaces)

{

tempTrain = train1;

}

}

foreach (Train train in collection.trains)

{

if (train == tempTrain)

{

train.Price = Math.Round((train.Price \* 0.8), 2);

}

dataGridView1.Rows.Add(train.NumberOfTrain, train.FromDestination, train.ToDestination, train.Price, train.AmountOfFreePlaces);

}

}

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

{

SaveFileDialog saveFileDialog = new SaveFileDialog();

saveFileDialog.Filter = "Data files(\*.dat)|\*.dat|All files(\*.\*)|\*.\*";

if (saveFileDialog.ShowDialog() == DialogResult.Cancel)

return;

string filename = saveFileDialog.FileName;

Collection collection = new Collection();

for (int i = 0; i < dataGridView1.RowCount - 1; i++)

{

collection.Add(new Train(

dataGridView1.Rows[i].Cells[0].Value.ToString(),

dataGridView1.Rows[i].Cells[1].Value.ToString(),

dataGridView1.Rows[i].Cells[2].Value.ToString(),

Convert.ToDouble(dataGridView1.Rows[i].Cells[3].Value.ToString()),

Convert.ToInt32(dataGridView1.Rows[i].Cells[4].Value.ToString())));

}

dataGridView1.Rows.Clear();

BinaryFormatter formatter = new BinaryFormatter();

using (FileStream fs = new FileStream(filename, FileMode.OpenOrCreate))

{

formatter.Serialize(fs, collection.trains);

}

}

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

{

OpenFileDialog openFileDialog = new OpenFileDialog();

openFileDialog.Filter = "Data files(\*.dat)|\*.dat|All files(\*.\*)|\*.\*";

if (openFileDialog.ShowDialog() == DialogResult.Cancel)

{

return;

}

string fileName = openFileDialog.FileName;

Collection collection = new Collection();

BinaryFormatter formatter = new BinaryFormatter();

using (FileStream fs = new FileStream(fileName, FileMode.OpenOrCreate))

{

collection.trains = (List<Train>)formatter.Deserialize(fs);

}

foreach (Train train in collection.trains)

{

dataGridView1.Rows.Add(train.NumberOfTrain, train.FromDestination, train.ToDestination, train.Price, train.AmountOfFreePlaces);

}

}

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

{

string numberOfTrain = number.Text;

string fromDestination = fromDest.Text;

string toDestination = toDest.Text;

double priceOf = Math.Round((Convert.ToDouble(price.Text)), 2);

int places = Convert.ToInt32(amountOfPlaces.Text);

dataGridView1.Rows.Add(numberOfTrain, fromDestination, toDestination,

priceOf, places);

number.Clear();

fromDest.Clear();

toDest.Clear();

price.Clear();

amountOfPlaces.Clear();

Collection collection = new Collection();

Train train = new Train(numberOfTrain, fromDestination, toDestination,

priceOf, places);

collection.Add(train);

}

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

{

int index = dataGridView1.SelectedCells[0].RowIndex;

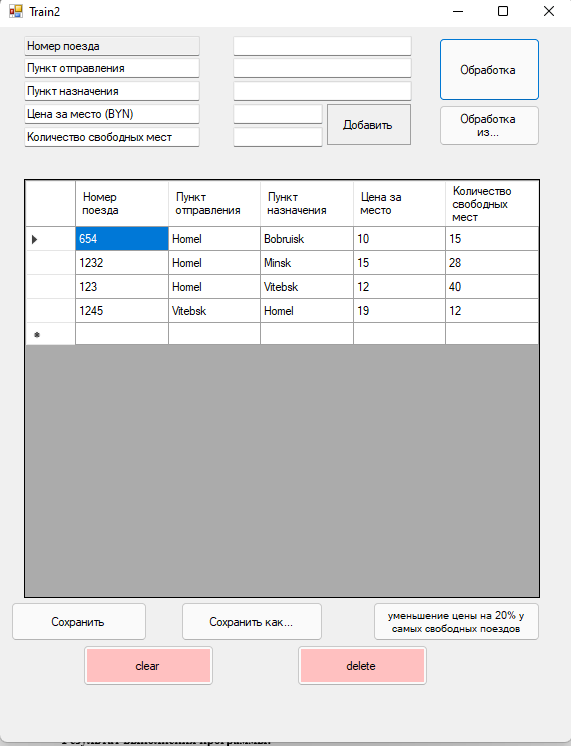
dataGridView1.Rows.RemoveAt(index);

}

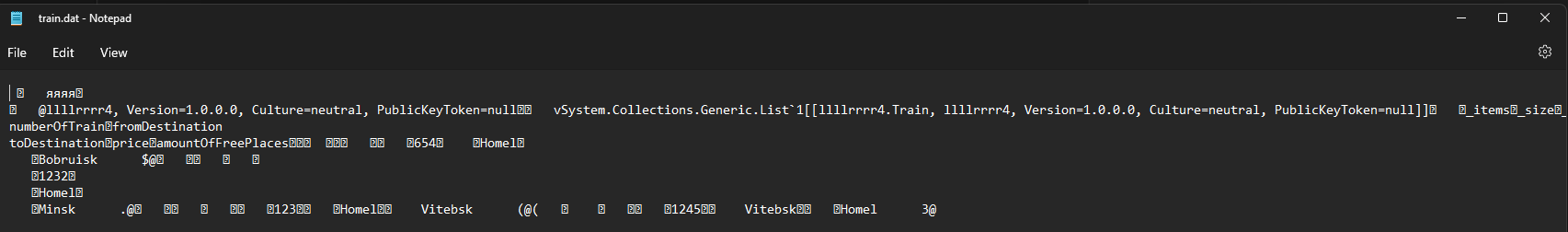
}

}

**Результат выполнения программы:**

****

Сериализованный файл:

****

**Вывод:** *В ходе лабораторной работы были получены навыки разработки программ, использующих классы File, FileInfo, DirectoryInfo, FileSystemWatcher, FileStream, BinaryReader и BinaryWriter, научиться осуществлять сериализацию и десериализацию объектов.*