**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

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

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

**ЛАБОРАТОРНАЯ РАБОТА №8**

по дисциплине: «Объектно-ориентированное программирование»

на тему: «Интерфейсы и абстрактные классы»

Выполнил: студент гр. ИТП-21

Король В.Н.

Принял: преподаватель-стажер

Гуменников Е.Д.

Гомель 2022

**Цель работы:** Изучить что такое абстрактные классы и интерфейсы. Изучить основные принципы наследования в *C*#. Реализовать оконное приложение при помощи библиотеки классов *WPF*.

**Задание**

1. Необходимо решить задачу, согласно варианта (таблица 8)

2. При создании классов руководствоваться Code Convention

3. Весь код должен содержать элементы документирования

4. Разработать модульные тесты для верификации созданных классов

5. Классы должен быть размещены в библиотеке классов

6. Модульные тесты – в отдельном проекте

7. В отдельном проекте реализовать графический интерфейс: для чётных вариантов – WPF, для нечётных – WFA.

8. Приложение должно обеспечивать ввод, редактирование и просмотр данных

**Вариант 14**

1.1 Создать иерархию классов «Наземного транспорта» («Поезд», «Автобус») и класс «Самолетов», реализующих общий класс «Транспорт»

1.2 Класс «Транспорт» должен быть абстрактным и содержать следующие элементы: абстрактный метод, возвращающий количество свободных мест; поля - пункт отправления и пункт назначения, вид транспорта; метод вывода информации.

1.3 Класс наземного транспорта должен содержать следующие элементы: поле – номер рейса; поле – массив, содержащий стоимость билета.

1.4 Класс автобусов должен содержать поле с количеством свободных мест; индексатор, в котором индекс может принимать значения мягкий, жесткий; переопределять метод, возвращающее количество свободных мест.

1.5 Класс поездов должен содержать поле-массив с количеством свободных мест в каждом вагоне; индексатор, в котором индекс может принимать значения люкс, купейный, плацкартный, общий; переопределять метод, возвращающий количество свободных мест.

1.6 Класс самолетов должен содержать поле с количеством свободных мест; поле – номер рейса; поле – массив, содержащий стоимость билета; индексатор, в котором индекс может принимать значения эконом, бизнес, первый.

1.7 Разработать программу, которая выполняет следующие действия:

 считывает информацию из текстового файла, каждая строка которого

содержит вид транспорта, номер рейса, пункт отправления, пункт

назначения, стоимость билетов, например:

самолет 253 Москва Париж 250 280 350;

 формирует на основании этой информации массив объектов типа

«Транспорт»; вводит всю недостающую информацию; выводит информацию, выделяя красным цветом строки с информацией о самолетах и голубым – о поездах. выводит цену билета заданного вида на заданный вид транспорта.

**Ход Работы**

В ходе выполнения лабораторной работы был создан абстрактный класс *Transport* который является родительским для абстрактного класса *GroundTransport* и для класса *Plain.* От класса *GroundTransport* наследуется два класса *Train* и *Bus.*

После создания иерархии классов был разработан графический интерфейс по управлением *WPF.* Была создано главное окно, где есть поля для ввода ссылки на файл и кнопка подтвердить. Пример главного окна указан а рисунке 1.



Рисунок 1 – Пример главного окна

Далее был созданы три дополнительных окна в которых выводится информации о цене билетов каждого класса для каждого вида транспорта при помощи индексаторов. Пример окна для самолетов указна на рисунке 2, для автобусов указна на рисунке 3 и для поездов указана на рисунке 4.

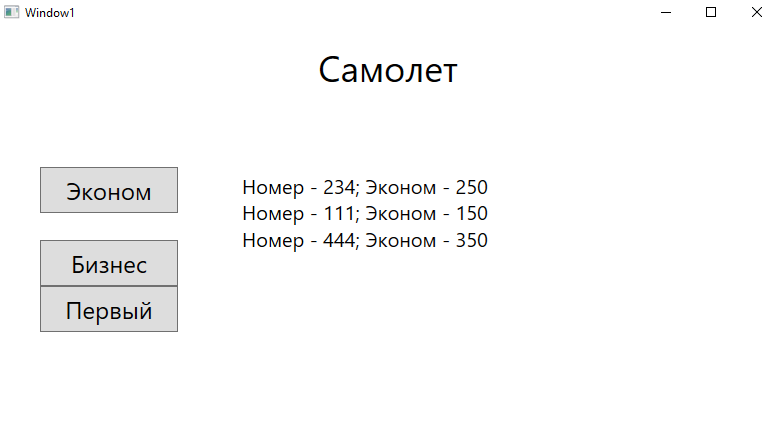


Рисунок 2 – Пример окна для самолета

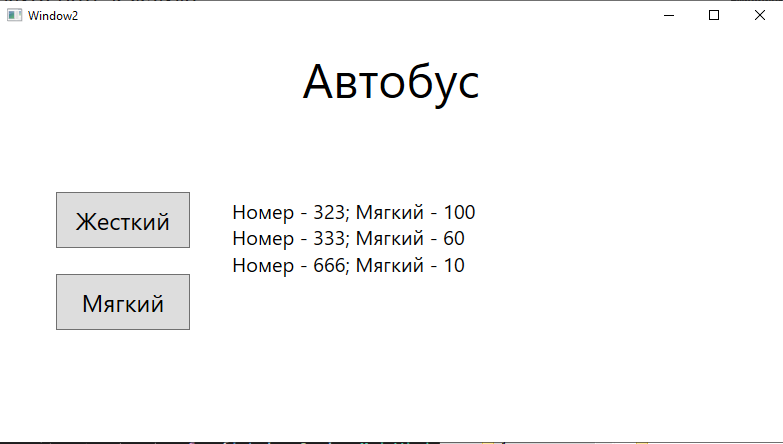


Рисунок 3 – Пример окна для автобуса

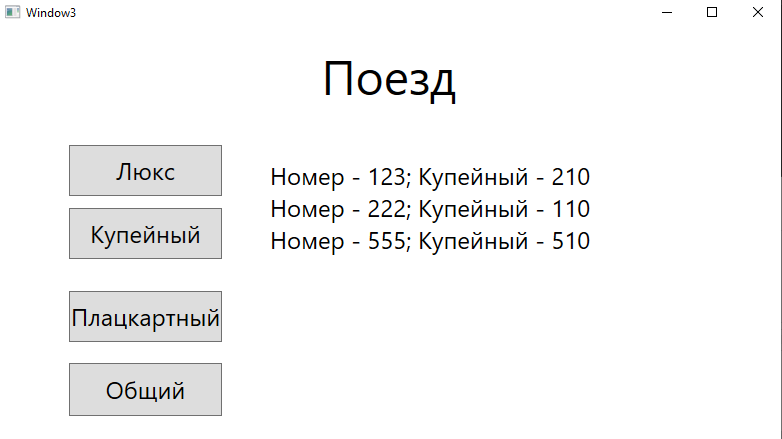


Рисунок 3 – Пример окна для поезда

**Вывод:** в ходе выполнения лабораторной работы была изучена работа с *WPF.* Изучена такая концепция как наследование в языке программирования *C*#. И такие типы данных как абстрактные классы и интерфейсы.

**Листинг Класса Transport**

using System;

using System.Collections.Generic;

using System.Diagnostics.CodeAnalysis;

using System.Linq;

using System.Runtime.ExceptionServices;

using System.Runtime.InteropServices.WindowsRuntime;

using System.Security.Policy;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Markup;

namespace Lab8

{

abstract class Transport

{

public string StartPoint;

public string EndPoint;

public string TransportType;

public Transport(string transportType, string startPoint, string endPoint)

{

StartPoint = startPoint;

EndPoint = endPoint;

TransportType = transportType;

}

public abstract int GetFreePlace();

public virtual string ShowInfo()

{

return $"{TransportType}; {StartPoint} - {EndPoint};";

}

}

}

**Листинг Класса GroundTransport**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.RightsManagement;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Markup;

namespace Lab8

{

abstract class GroundTransport : Transport

{

public int TripNum;

public int[] TicketPrice;

public GroundTransport(string transportType, string startPoint, string endPoint, int tripNum, int[] ticketPrice) : base(transportType, startPoint, endPoint)

{

TripNum = tripNum;

TicketPrice = ticketPrice;

}

}

}

**Листинг Класса Train**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab8

{

class Train : GroundTransport

{

int[] KolFreePlaceInTram;

public Train(string transportType, string startPoint, string endPoint, int tripNum, int[] ticketPrice) : base(transportType, startPoint, endPoint, tripNum, ticketPrice)

{

}

public int this[string placeType]

{

set

{

switch (placeType)

{

case "Люкс":

TicketPrice[0] = value;

break;

case "Купейный":

TicketPrice[1] = value;

break;

case "Плацкартный":

TicketPrice[2] = value;

break;

case "Общий":

TicketPrice[3] = value;

break;

}

}

get

{

switch (placeType)

{

case "Люкс":

return TicketPrice[0];

case "Купейный":

return TicketPrice[1];

case "Плацкартный":

return TicketPrice[2];

case "Общий":

return TicketPrice[3];

default:

return -1;

}

}

}

public override int GetFreePlace()

{

int sum = 0;

for (int i = 0; i < KolFreePlaceInTram.Length; i++)

{

sum += KolFreePlaceInTram[i];

}

return sum;

}

public override string ShowInfo()

{

return $"{base.ShowInfo()}; Номер - {TripNum}; Люкс - {TicketPrice[0]} Купейный - {TicketPrice[1]} Плацкартный - {TicketPrice[2]} Общий - {TicketPrice[3]}";

}

}

}

**Листинг Класса Bus**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab8

{

class Bus : GroundTransport

{

public int FreePlace;

public Bus(string transportType, string startPoint, string endPoint, int tripNum, int[] ticketPrice) : base(transportType, startPoint, endPoint, tripNum, ticketPrice)

{

}

public int this[string placeType]

{

set

{

if (placeType == "Мягкий")

{

TicketPrice[0] = value;

}

else if (placeType == "Жесткий")

{

TicketPrice[1] = value;

}

}

get

{

if (placeType == "Мягкий")

{

return TicketPrice[0];

}

else if (placeType == "Жесткий")

{

return TicketPrice[1];

}

else

{

return -1;

}

}

}

public override int GetFreePlace()

{

return FreePlace;

}

public override string ShowInfo()

{

return $"{base.ShowInfo()} Номер - {TripNum}; Мягкие - {TicketPrice[0]} Жесткие - {TicketPrice[1]}";

}

}

}

**Листинг Класса Plain**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab8

{

class Plain : Transport

{

public int FreePlace;

public int TripNum;

public int[] TicketPrice;

public Plain(string transportType, string startPoint, string endPoint, int tripNum, int[] ticketPrice) : base(transportType, startPoint, endPoint)

{

TripNum = tripNum;

TicketPrice = ticketPrice;

}

public int this[string placeType]

{

set

{

switch (placeType)

{

case "Эконом":

TicketPrice[0] = value;

break;

case "Бизнес":

TicketPrice[1] = value;

break;

case "Первый":

TicketPrice[2] = value;

break;

}

}

get

{

switch (placeType)

{

case "Эконом":

return TicketPrice[0];

case "Бизнес":

return TicketPrice[1];

case "Первый":

return TicketPrice[2];

default:

return -1;

}

}

}

public override int GetFreePlace()

{

return FreePlace;

}

public override string ShowInfo()

{

return $"{base.ShowInfo()} Номер - {TripNum}; Эконом - {TicketPrice[0]} Бизнес - {TicketPrice[1]} Первый - {TicketPrice[2]}";

}

}

}

**Листинг Класса MainWindow**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Lab8

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Path\_TextChanged(object sender, TextChangedEventArgs e)

{

}

private void Confirm\_Click(object sender, RoutedEventArgs e)

{

if (File.Exists(Path.Text))

{

if (Path.Text.Split('.')[Path.Text.Split('.').Length - 1] != "txt")

{

label1.Content = "Файл должен быть с расширением .txt";

}

else

{

label1.Content = "Введите путь к файлу";

string textFromFile;

using (FileStream fstream = File.OpenRead(Path.Text))

{

byte[] buffer = new byte[fstream.Length];

fstream.Read(buffer, 0, buffer.Length);

textFromFile = Encoding.Default.GetString(buffer);

}

string[] textArr = textFromFile.Split('\n');

string[] resault = new string[3] {"", "", ""};

MainWindowHelpers.transports = new Transport[textArr.Length];

for(int i = 0; i < textArr.Length; i++)

{

string[] trans = textArr[i].Split(' ');

if (trans[0] == "самолет")

{

MainWindowHelpers.transports[i] = new Plain("Cамолет", trans[2], trans[3], Convert.ToInt32(trans[1]),

new int[3] { Convert.ToInt32(trans[4]), Convert.ToInt32(trans[5]), Convert.ToInt32(trans[6]) });

resault[0] += ((Plain)MainWindowHelpers.transports[i]).ShowInfo() + "\n";

}

else if(trans[0] == "поезд")

{

MainWindowHelpers.transports[i] = new Train("Поезд", trans[2], trans[3], Convert.ToInt32(trans[1]),

new int[4] { Convert.ToInt32(trans[4]), Convert.ToInt32(trans[5]), Convert.ToInt32(trans[6]),

Convert.ToInt32(trans[7]) });

resault[1] += ((Train)MainWindowHelpers.transports[i]).ShowInfo() + "\n";

}

else if (trans[0] == "автобус")

{

MainWindowHelpers.transports[i] = new Bus("Автобус", trans[2], trans[3], Convert.ToInt32(trans[1]),

new int[2] { Convert.ToInt32(trans[4]), Convert.ToInt32(trans[5])});

resault[2] += ((Bus)MainWindowHelpers.transports[i]).ShowInfo() + "\n";

}

}

Plain.Foreground = Brushes.Red;

Train.Foreground = Brushes.Blue;

Plain.Content = resault[0];

Train.Content = resault[1];

Bus.Content = resault[2];

}

}

else

{

label1.Content = "Такого файла нет";

}

}

private void PlainBtn\_Click(object sender, RoutedEventArgs e)

{

Window1 window1 = new Window1();

window1.Show();

}

private void BusBtn\_Click(object sender, RoutedEventArgs e)

{

Window2 window2 = new Window2();

window2.Show();

}

private void TrainBtn\_Click(object sender, RoutedEventArgs e)

{

Window3 window3 = new Window3();

window3.Show();

}

}

}

**Листинг Класса Window1**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace Lab8

{

/// <summary>

/// Логика взаимодействия для Window1.xaml

/// </summary>

public partial class Window1 : Window

{

public Window1()

{

InitializeComponent();

}

private void Econom\_Click(object sender, RoutedEventArgs e)

{

ShowResault("Эконом");

}

private void Busness\_Click(object sender, RoutedEventArgs e)

{

ShowResault("Бизнес");

}

private void Pervi\_Click(object sender, RoutedEventArgs e)

{

ShowResault("Первый");

}

public void ShowResault(string type)

{

if (MainWindowHelpers.transports.Length == 0)

{

Resault.Content = "Ошибка";

}

else

{

string resault = "";

for (int i = 0; i < MainWindowHelpers.transports.Length; i++)

{

if (MainWindowHelpers.transports[i].TransportType == "Cамолет")

{

resault += $"Номер - {((Plain)MainWindowHelpers.transports[i]).TripNum}; {type} - {((Plain)MainWindowHelpers.transports[i])[type]}\n";

}

}

Resault.Content = resault;

}

}

}

}

**Листинг Класса Window2**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace Lab8

{

/// <summary>

/// Логика взаимодействия для Window2.xaml

/// </summary>

public partial class Window2 : Window

{

public Window2()

{

InitializeComponent();

}

private void Zhostki\_Click(object sender, RoutedEventArgs e)

{

ShowResault("Жесткий");

}

private void Soft\_Click(object sender, RoutedEventArgs e)

{

ShowResault("Мягкий");

}

public void ShowResault(string type)

{

if (MainWindowHelpers.transports.Length == 0)

{

Resault.Content = "Ошибка";

}

else

{

string resault = "";

for (int i = 0; i < MainWindowHelpers.transports.Length; i++)

{

if (MainWindowHelpers.transports[i].TransportType == "Автобус")

{

resault += $"Номер - {((Bus)MainWindowHelpers.transports[i]).TripNum}; {type} - {((Bus)MainWindowHelpers.transports[i])[type]}\n";

}

}

Resault.Content = resault;

}

}

}

}

**Листинг Класса Window3**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace Lab8

{

/// <summary>

/// Логика взаимодействия для Window3.xaml

/// </summary>

public partial class Window3 : Window

{

public Window3()

{

InitializeComponent();

}

private void Luxe\_Click(object sender, RoutedEventArgs e)

{

ShowResault("Люкс");

}

private void Kyp\_Click(object sender, RoutedEventArgs e)

{

ShowResault("Купейный");

}

private void Plac\_Click(object sender, RoutedEventArgs e)

{

ShowResault("Плацкартный");

}

private void Obz\_Click(object sender, RoutedEventArgs e)

{

ShowResault("Общий");

}

public void ShowResault(string type)

{

if (MainWindowHelpers.transports.Length == 0)

{

Resault.Content = "Ошибка";

}

else

{

string resault = "";

for (int i = 0; i < MainWindowHelpers.transports.Length; i++)

{

if (MainWindowHelpers.transports[i].TransportType == "Поезд")

{

resault += $"Номер - {((Train)MainWindowHelpers.transports[i]).TripNum}; {type} - {((Train)MainWindowHelpers.transports[i])[type]}\n";

}

}

Resault.Content = resault;

}

}

}

}