МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7

# по дисциплине «Объектно-ориентированное проектирование и программирование»

# на тему: «Наследование классов»

Выполнил студент гр. ИТИ-21:  
Буякевич Д.О.  
 Принял преподаватель:  
 Чернявская А.Э.

Гомель 2022

**Цель:** научиться работать с принципом ООП наследование, закрепить знания с уже знакомыми принципами. Научиться создавать приложения с помощью WPF.

**Ход работы**

На рисунке 1 показаны требования к выполнению лабораторной работы.

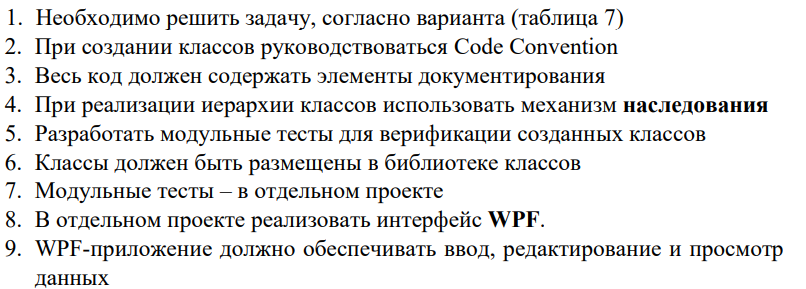


Рисунок 1 – Требования

На рисунке 2 показан вариант лабораторной работы

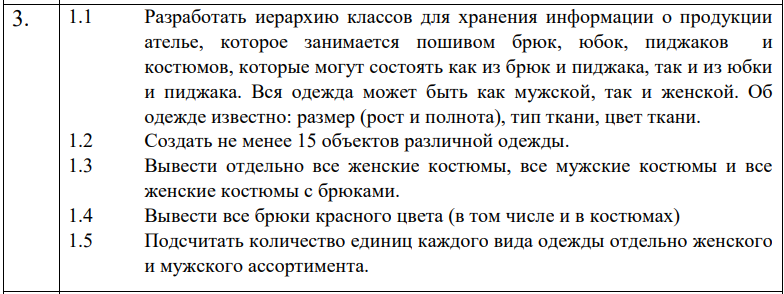


Рисунок 2 – Вариант Л/Р

На последующих рисунках показана работа программы для лабораторной работы с помощью WPF приложения.

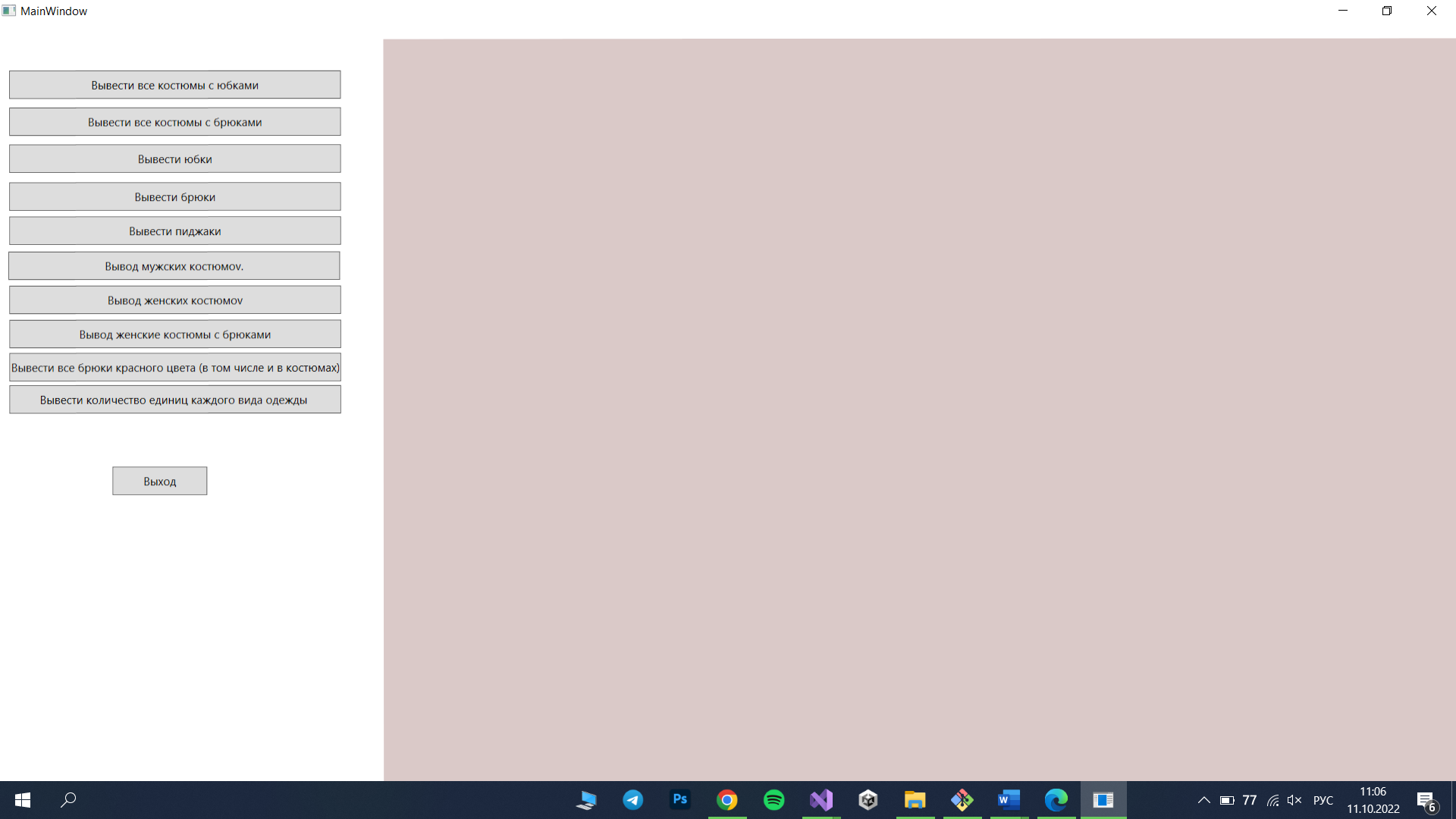


Рисунок 3 – Меню

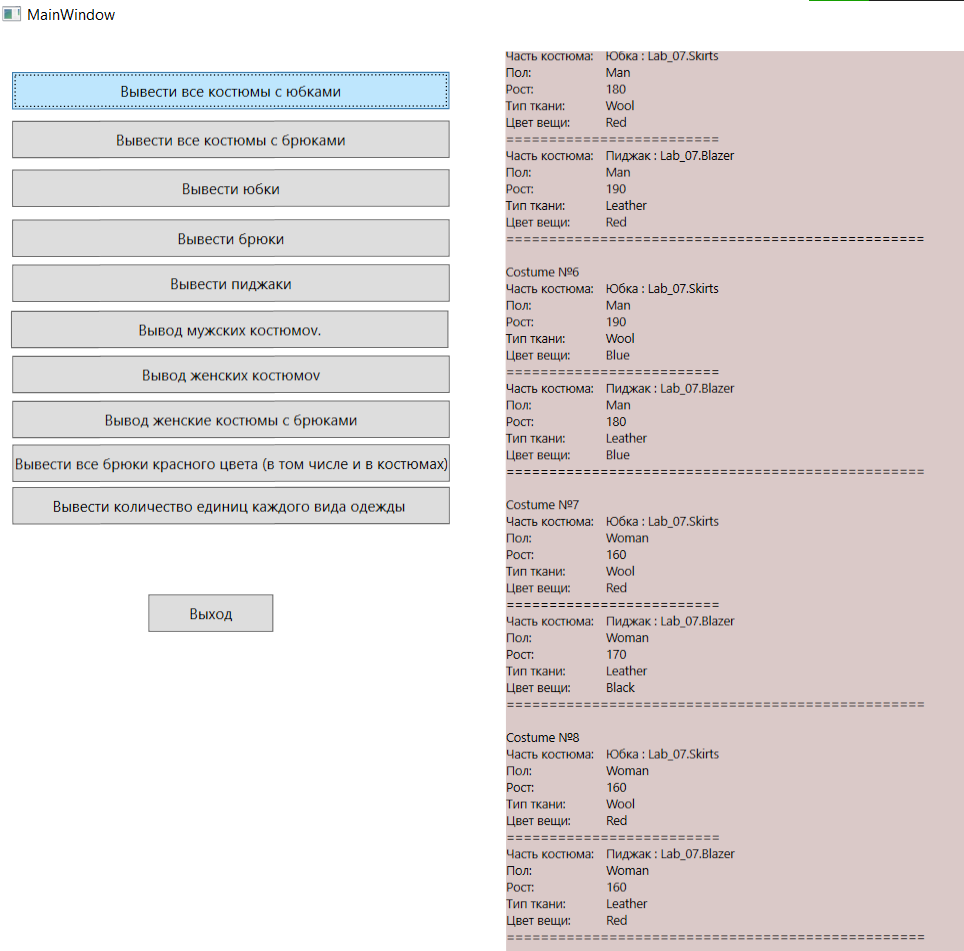


Рисунок 4 – Пункт 1

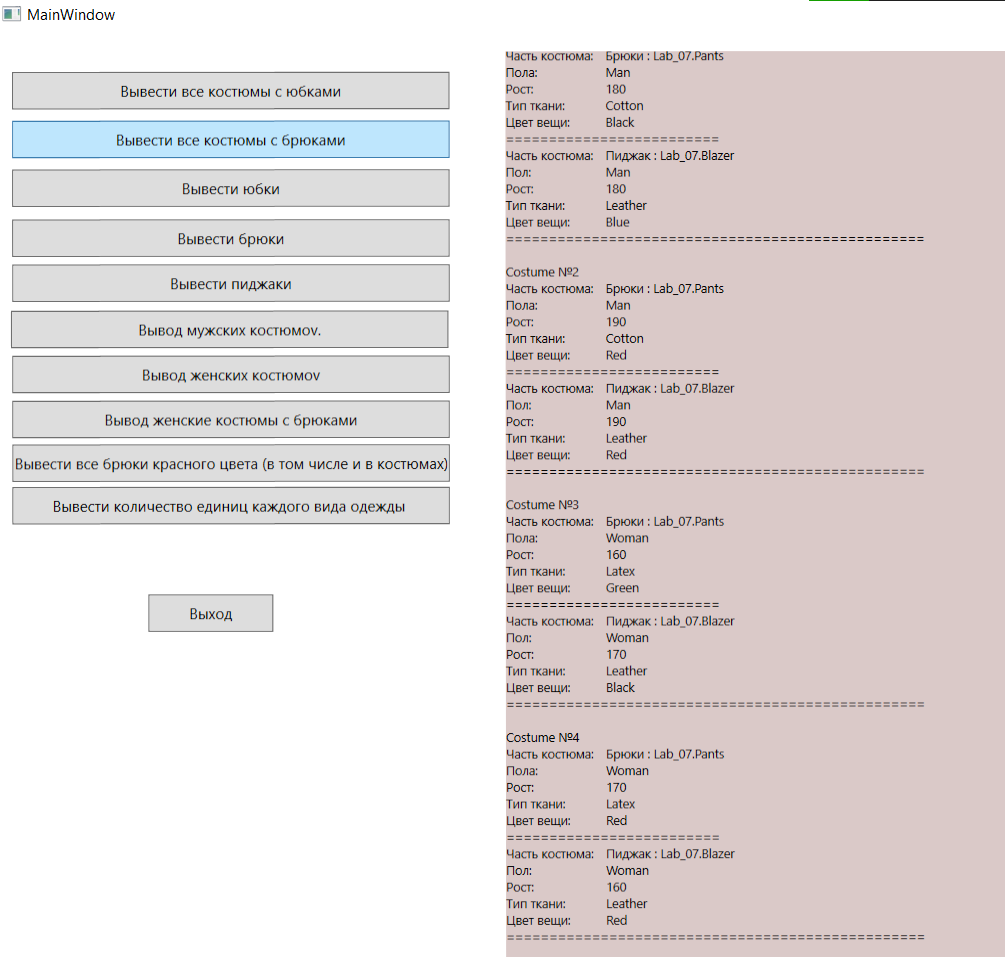


Рисунок 5 – Пункт 2

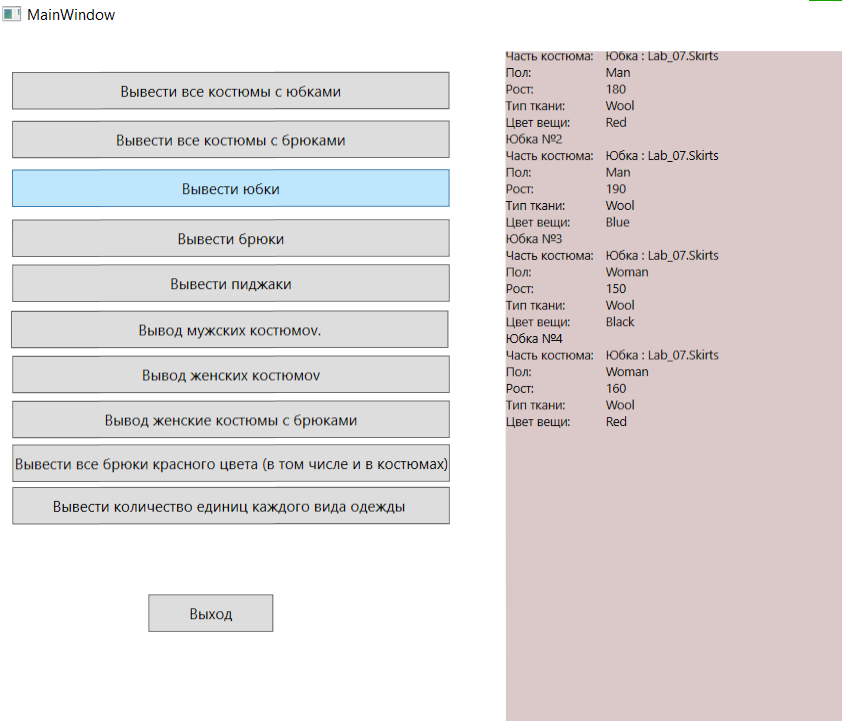


Рисунок 6 – Пункт 3

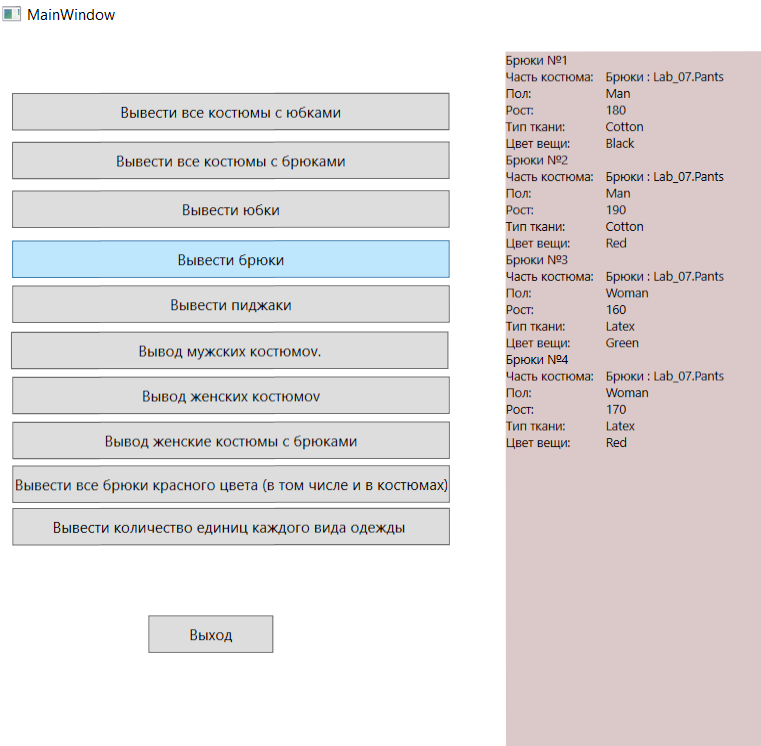


Рисунок 7 – Пункт 4

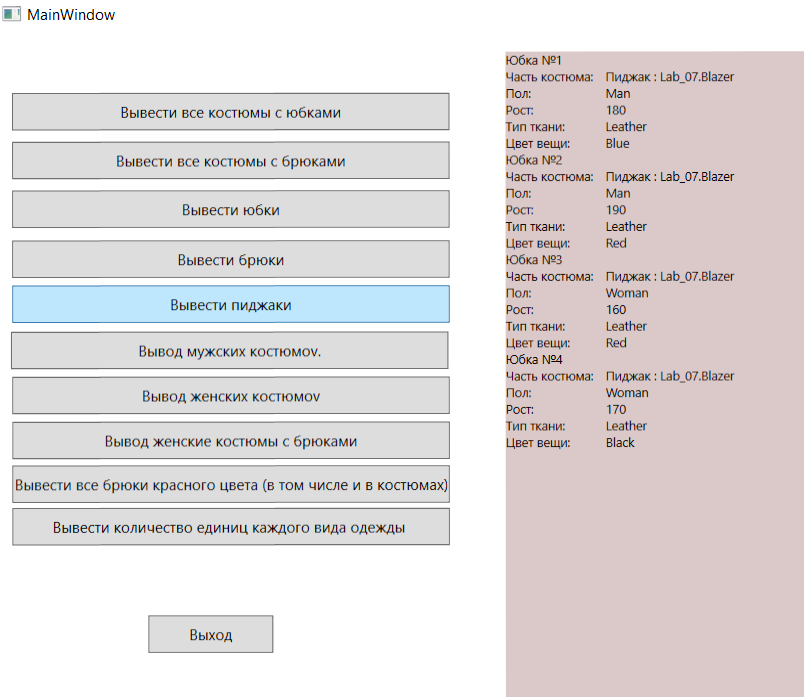


Рисунок 8 – Пункт 5

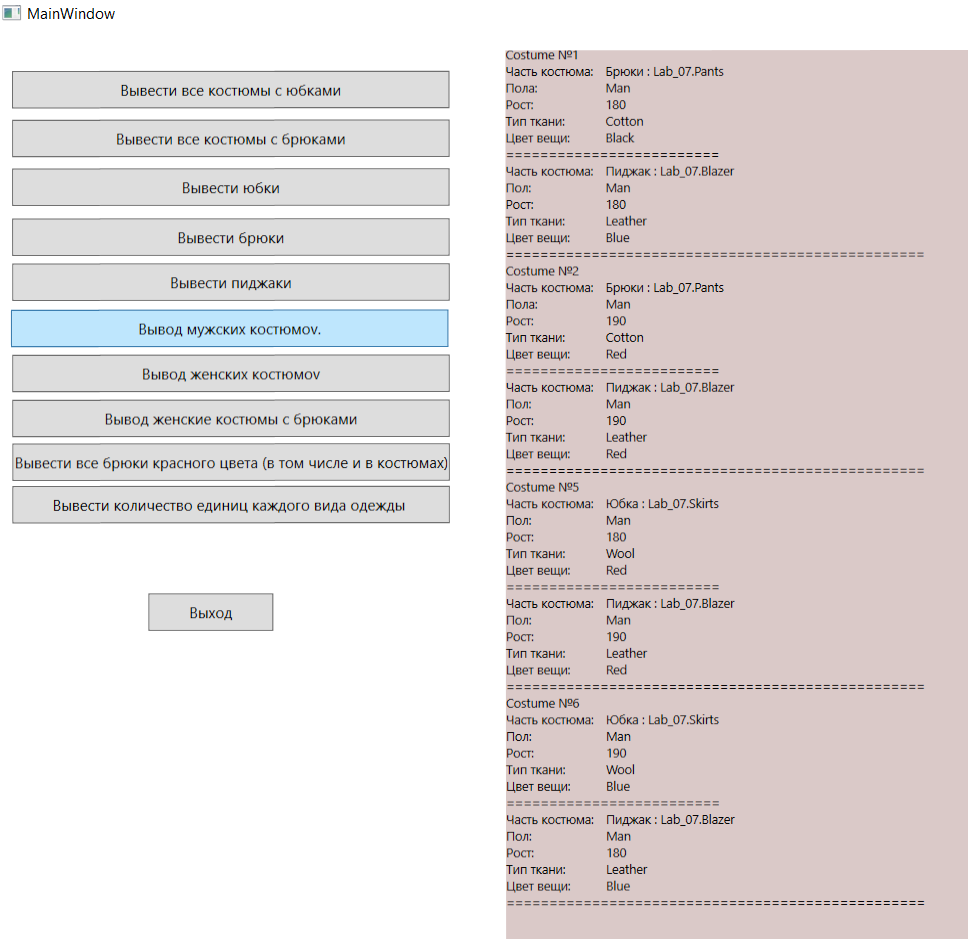


Рисунок 9 – Пункт 6

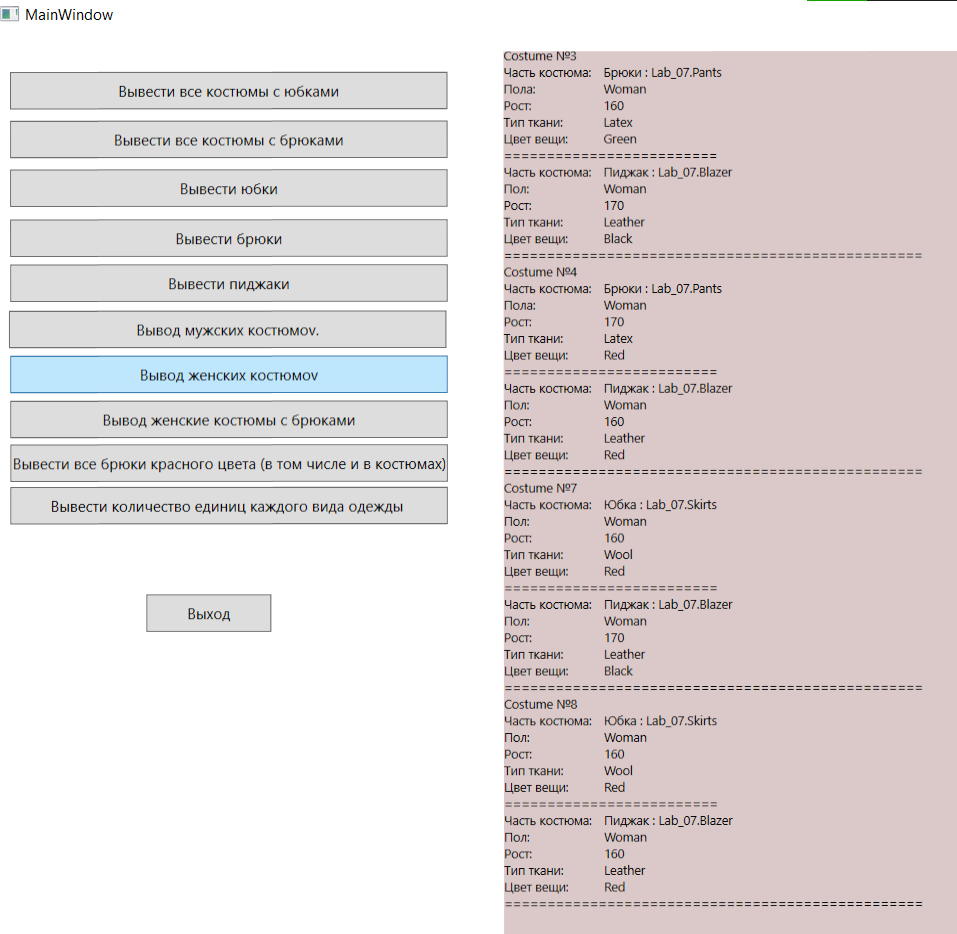


Рисунок 10 – Пункт 7

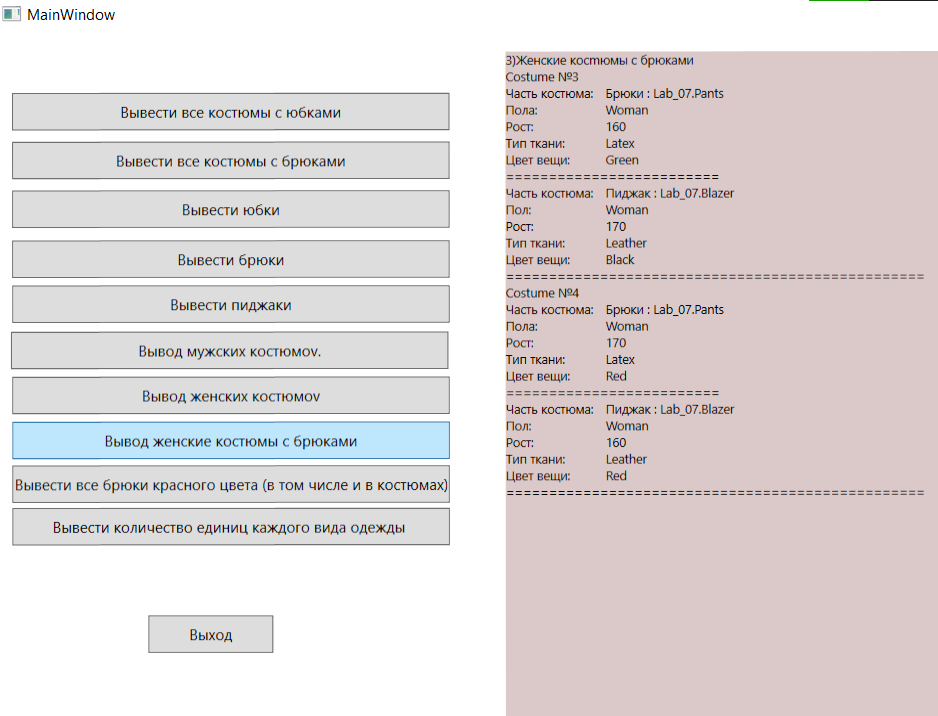


Рисунок 11 – Пункт 8

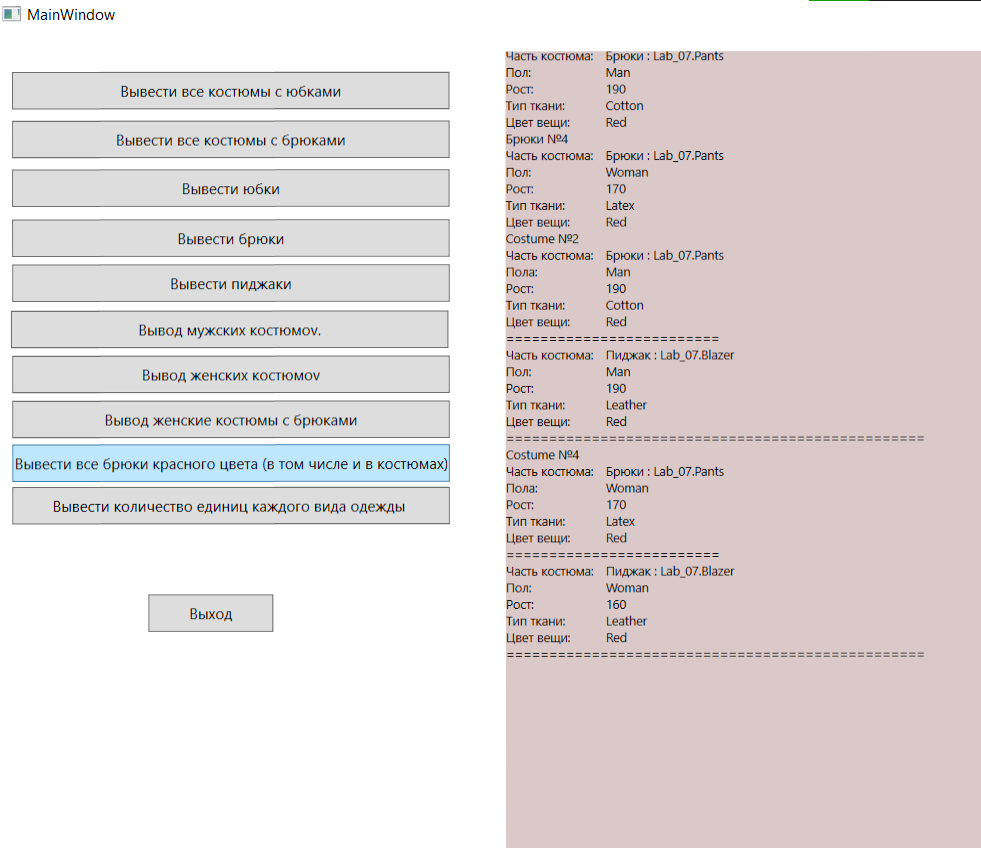


Рисунок 12 – Пункт 9

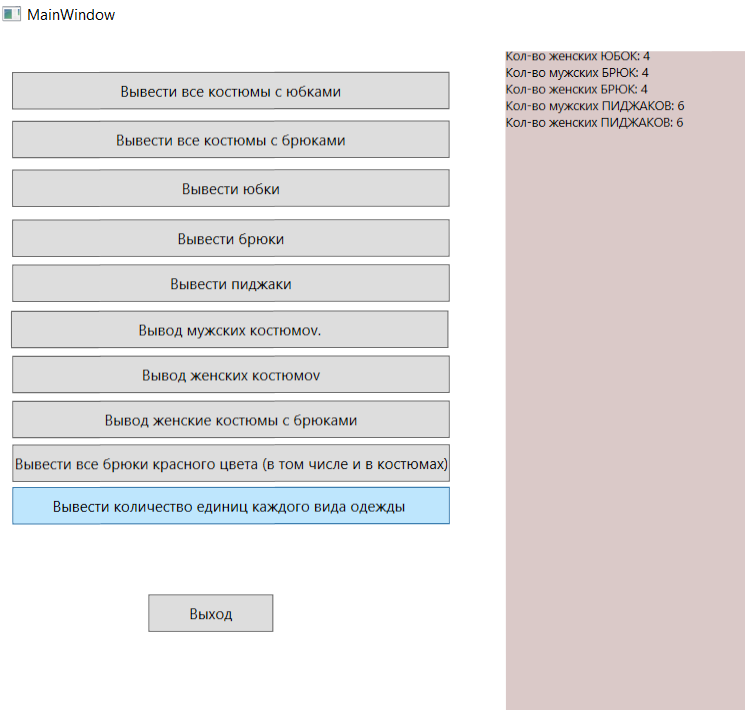


Рисунок 13 – Пункт 10

**Вывод:** научились работать с принципом ООП: наследование. Закрепили знания с уже знакомыми принципами. Научились создавать приложения с помощью WPF.

**ПРИЛОЖЕНИЕ А**

**Листинг программы**

**Program.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_07

{

internal class Program

{

static void Main(string[] args)

{

Pants[] pants =

{

new Pants(Gender.Man, 180, Type\_clothes.Cotton, Color.Black),

new Pants(Gender.Man, 190, Type\_clothes.Cotton, Color.Red),

new Pants(Gender.Woman, 160, Type\_clothes.Latex, Color.Green),

new Pants(Gender.Woman, 170, Type\_clothes.Latex, Color.Red),

};

Skirts[] skirts =

{

new Skirts(Gender.Man, 180, Type\_clothes.Wool, Color.Red),

new Skirts(Gender.Man, 190, Type\_clothes.Wool, Color.Blue),

new Skirts(Gender.Woman, 150, Type\_clothes.Wool, Color.Black),

new Skirts(Gender.Woman, 160, Type\_clothes.Wool, Color.Red),

};

Blazer[] blazers =

{

new Blazer(Gender.Man, 180, Type\_clothes.Leather, Color.Blue),

new Blazer(Gender.Man, 190, Type\_clothes.Leather, Color.Red),

new Blazer(Gender.Woman, 160, Type\_clothes.Leather, Color.Red),

new Blazer(Gender.Woman, 170, Type\_clothes.Leather, Color.Black),

};

Costume[] costumesWithPants =

{

new Costume(pants[0],blazers[0]), //MAN

new Costume(pants[1],blazers[1]), //MAN

new Costume(pants[2],blazers[3]), //WOMAN

new Costume(pants[3],blazers[2]), //WOMAN

};

Costume[] costumesWithSkirts =

{

new Costume(skirts[0],blazers[1]), //MAN

new Costume(skirts[1],blazers[0]), //MAN

new Costume(skirts[3],blazers[3]), //WOMAN

new Costume(skirts[3],blazers[2]), //WOMAN

};

bool isWork = true;

while (isWork)

{

Console.WriteLine("=============Меню=============");

Console.WriteLine("1)Вывести все костюмы");

Console.WriteLine("2)Вывести только ЮБКИ");

Console.WriteLine("3)Вывести только ПИДЖАКИ");

Console.WriteLine("4)Вывести только БРЮКИ");

Console.WriteLine("5)Вывод мужских/женских костюмоv.Вывод женские костюмы с брюками");

Console.WriteLine("6)Вывести все брюки красного цвета (в том числе и в костюмах)");

Console.WriteLine("7)Вывести количество единиц каждого вида одежды отдельно женского и мужского ассортимента");

Console.WriteLine("8)Выход");

int menuPunct = Convert.ToInt32(Console.ReadLine());

switch (menuPunct)

{

case 1:

Console.WriteLine("1)Костюм: ПИДЖАК + БРЮКИ");

Console.WriteLine("2)Костюм: ПИДЖАК + ЮБКА");

byte whichCostume = Convert.ToByte(Console.ReadLine());

if (whichCostume == 1)

{

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

{

Console.WriteLine( Costume.ShowCostumeWithPants(costumesWithPants,i));

}

}

else if (whichCostume == 2)

{

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

{

Console.WriteLine(Costume.ShowCostumeWithSkirts(costumesWithSkirts,i));

}

}

else

{

Console.WriteLine("ТАКОГО ПУНКТА НЕТ!");

}

Console.ReadKey();

Console.Clear();

break;

case 2:

Console.WriteLine("Весь ассортимент ЮБОК");

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

{

Console.WriteLine(Skirts.ShowSkirt(skirts, i));

}

Console.ReadKey();

Console.Clear();

break;

case 3:

Console.WriteLine("Весь ассортимент ПИДЖАКОВ");

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

{

Console.WriteLine(Blazer.ShowBlazers(blazers, i));

}

Console.ReadKey();

Console.Clear();

break;

case 4:

Console.WriteLine("Весь ассортимент БРЮК");

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

{

Console.WriteLine(Pants.ShowPants(pants,i));

}

Console.ReadKey();

Console.Clear();

break;

case 5:

Console.WriteLine("1)Мужские косmюмы");

Console.WriteLine("2)Жеснкие косmюмы");

Console.WriteLine("3)Женские косmюмы с брюками");

int whichGenderCostume = Convert.ToInt32(Console.ReadLine());

if (whichGenderCostume == 1)

{

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

{

if (costumesWithPants[i].Pants.Gender == Gender.Man)

Console.WriteLine(Costume.ShowCostumeWithPants(costumesWithPants, i));

}

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

{

if (costumesWithSkirts[i].Skirt.Gender == Gender.Man)

Console.WriteLine(Costume.ShowCostumeWithSkirts(costumesWithSkirts, i));

}

}

else if (whichGenderCostume == 2)

{

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

{

if (costumesWithPants[i].Pants.Gender == Gender.Woman)

Console.WriteLine(Costume.ShowCostumeWithPants(costumesWithPants, i));

}

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

{

if (costumesWithSkirts[i].Skirt.Gender == Gender.Woman)

Console.WriteLine(Costume.ShowCostumeWithSkirts(costumesWithSkirts, i));

}

}

else if (whichGenderCostume == 3)

{

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

{

if ((costumesWithPants[i].Pants.Gender == Gender.Woman))

Console.WriteLine(Costume.ShowCostumeWithPants(costumesWithPants, i));

}

}

else

{

Console.WriteLine("ТАКОГО ПУНКТА НЕТ!");

}

Console.ReadKey();

Console.Clear();

break;

case 6:

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

{

if (pants[i].Color == Color.Red)

Console.WriteLine(Pants.ShowPants(pants, i));

}

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

{

if (costumesWithPants[i].Pants.Color == Color.Red)

Console.WriteLine(Costume.ShowCostumeWithPants(costumesWithPants, i));

}

Console.ReadKey();

Console.Clear();

break;

case 7:

Console.WriteLine(Costume.ShowClothesAssortment(pants, skirts, blazers, costumesWithPants, costumesWithSkirts));

Console.ReadKey();

Console.Clear();

break;

case 8:

isWork = false;

break;

default:

Console.Clear();

continue;

}

}

}

}

public enum Gender

{

Man,

Woman

}

public enum Color

{

Red,

Green,

Blue,

Black

}

public enum Type\_clothes

{

Wool,

Cotton,

Leather,

Latex

}

}

**Pants.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_07

{

public class Pants : Costume

{

public Pants(Gender gender, double growth, Type\_clothes type, Color color) : base(gender, growth, type, color)

{

}

static public string ShowPants(Pants[] pants,int index)

{

return $"Брюки №{index + 1}" + "\n" +

"Часть костюма:\t" + "Брюки : " + pants[index].GetType() + "\n" +

"Пол:\t\t" + pants[index].Gender + "\n" +

"Рост:\t\t" + pants[index].Growth + "\n" +

"Тип ткани:\t" + pants[index].Type + "\n" +

"Цвет вещи:\t" + pants[index].Color + "\n";

}

}

}

**Skirts.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_07

{

public class Skirts : Costume

{

public Skirts(Gender gender, double growth, Type\_clothes type, Color color) : base(gender, growth, type, color)

{

}

static public string ShowSkirt(Skirts[] skirts,int index)

{

return $"Юбка №{index + 1}" + "\n" +

"Часть костюма:\t" + "Юбка : " + skirts[index].GetType() + "\n" +

"Пол:\t\t" + skirts[index].Gender + "\n" +

"Рост:\t\t" + skirts[index].Growth + "\n" +

"Тип ткани:\t" + skirts[index].Type + "\n" +

"Цвет вещи:\t" + skirts[index].Color + "\n";

}

}

}

**Costume.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_07

{

public class Costume : Clothes

{

public Pants Pants { get; set; }

public Blazer Blazer { get; set; }

public Skirts Skirt { get; set; }

protected Costume(Gender gender, double growth, Type\_clothes type, Color color) : base(gender, growth, type, color)

{

}

public Costume(Pants pants, Blazer blazer) : base(pants, blazer)

{

Pants = pants;

Blazer = blazer;

}

public Costume(Skirts skirt, Blazer blazer) : base(skirt, blazer)

{

Skirt = skirt;

Blazer = blazer;

}

static public string ShowCostumeWithPants(Costume[] costumesWithPants,int index)

{

//costumesWithPants.Length

return $"Costume №{index + 1}" + "\n" +

"Часть костюма:\t" + "Брюки : " + costumesWithPants[index].Pants.GetType() + "\n" +

"Полa:\t\t" + costumesWithPants[index].Pants.Gender + "\n" +

"Рост:\t\t" + costumesWithPants[index].Pants.Growth + "\n" +

"Тип ткани:\t" + costumesWithPants[index].Pants.Type + "\n" +

"Цвет вещи:\t" + costumesWithPants[index].Pants.Color + "\n" +

"=========================" + "\n" +

"Часть костюма:\t" + "Пиджак : " + costumesWithPants[index].Blazer.GetType() + "\n" +

"Пол:\t\t" + costumesWithPants[index].Blazer.Gender + "\n" +

"Рост:\t\t" + costumesWithPants[index].Blazer.Growth + "\n" +

"Тип ткани:\t" + costumesWithPants[index].Blazer.Type + "\n" +

"Цвет вещи:\t" + costumesWithPants[index].Blazer.Color + "\n" +

"=================================================" + "\n";

}

static public string ShowCostumeWithSkirts(Costume[] costumesWithSkirts,int index)

{

return $"Costume №{index + costumesWithSkirts.Length + 1}" + "\n" +

"Часть костюма:\t" + "Юбка : " + costumesWithSkirts[index].Skirt.GetType() + "\n" +

"Пол:\t\t" + costumesWithSkirts[index].Skirt.Gender + "\n" +

"Рост:\t\t" + costumesWithSkirts[index].Skirt.Growth + "\n" +

"Тип ткани:\t" + costumesWithSkirts[index].Skirt.Type + "\n" +

"Цвет вещи:\t" + costumesWithSkirts[index].Skirt.Color + "\n" +

"=========================" + "\n" +

"Часть костюма:\t" + "Пиджак : " + costumesWithSkirts[index].Blazer.GetType() + "\n" +

"Пол:\t\t" + costumesWithSkirts[index].Blazer.Gender + "\n" +

"Рост:\t\t" + costumesWithSkirts[index].Blazer.Growth + "\n" +

"Тип ткани:\t" + costumesWithSkirts[index].Blazer.Type + "\n" +

"Цвет вещи:\t" + costumesWithSkirts[index].Blazer.Color + "\n" +

"=================================================" + "\n";

}

static public string ShowClothesAssortment(Pants[] pants, Skirts[] skirts, Blazer[] blazers, Costume[] costumesWithPants, Costume[] costumesWithSkirts)

{

int pantsCountMan = 0;

int skirtsCountMan = 0;

int blazersCountMan = 0;

int pantsCountWoMan = 0;

int skirtsCountWoMan = 0;

int blazersCountWoMan = 0;

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

{

if (pants[i].Gender == Gender.Man)

{

pantsCountMan++;

}

if (pants[i].Gender == Gender.Woman)

{

pantsCountWoMan++;

}

}

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

{

if (skirts[i].Gender == Gender.Man)

{

skirtsCountMan++;

}

if (pants[i].Gender == Gender.Woman)

{

skirtsCountWoMan++;

}

}

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

{

if (blazers[i].Gender == Gender.Man)

{

blazersCountMan++;

}

if (blazers[i].Gender == Gender.Woman)

{

blazersCountWoMan++;

}

}

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

{

if (costumesWithPants[i].Pants.Gender == Gender.Man)

{

pantsCountMan++;

blazersCountMan++;

}

if (costumesWithPants[i].Pants.Gender == Gender.Woman)

{

pantsCountWoMan++;

blazersCountWoMan++;

}

}

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

{

if (costumesWithSkirts[i].Skirt.Gender == Gender.Man)

{

skirtsCountMan++;

blazersCountMan++;

}

if (costumesWithSkirts[i].Skirt.Gender == Gender.Woman)

{

skirtsCountWoMan++;

blazersCountWoMan++;

}

}

return $"Кол-во мужских ЮБОК: {skirtsCountMan}" + "\n" +

$"Кол-во женских ЮБОК: {skirtsCountWoMan}" + "\n" +

$"Кол-во мужских БРЮК: {skirtsCountMan}" + "\n" +

$"Кол-во женских БРЮК: {skirtsCountWoMan}" + "\n" +

$"Кол-во мужских ПИДЖАКОВ: {blazersCountMan}" + "\n" +

$"Кол-во женских ПИДЖАКОВ: {blazersCountWoMan}" + "\n";

}

/\* static public string ShowManCostumes(Costume[] costumesWithPants, Costume[] costumesWithSkirts)

{

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

{

if (costumesWithPants[i].Pants.Gender == Gender.Man) ;

Costume.ShowCostumeWithPants(costumesWithPants,i);

}

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

{

if (costumesWithSkirts[i].Skirt.Gender == Gender.Man) ;

Costume.ShowCostumeWithSkirts(costumesWithSkirts,i);

}

}\*/

}

}

**Clothes.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_07

{

public class Clothes

{

public Gender Gender { get; set; }

public double Growth { get; set; }

public Type\_clothes Type { get; set; }

public Color Color { get; set; }

public Pants Pants { get; set; }

public Blazer Blazer { get; set; }

public Skirts Skirt { get; set; }

protected Clothes(Gender gender, double growth, Type\_clothes type, Color color)

{

Gender = gender;

Growth = growth;

Type = type;

Color = color;

}

public Clothes(Pants pants, Blazer blazer)

{

Pants = pants;

Blazer = blazer;

}

public Clothes(Skirts skirt, Blazer blazer)

{

Skirt = skirt;

Blazer = blazer;

}

}

}

**Blazer.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Reflection;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_07

{

public class Blazer : Costume

{

public Blazer(Gender gender, double growth, Type\_clothes type, Color color) : base(gender, growth, type, color)

{

}

static public string ShowBlazers(Blazer[] blazers, int index)

{

return $"Юбка №{index + 1}" + "\n" +

"Часть костюма:\t" + "Пиджак : " + blazers[index].GetType() + "\n" +

"Пол:\t\t" + blazers[index].Gender + "\n" +

"Рост:\t\t" + blazers[index].Growth + "\n" +

"Тип ткани:\t" + blazers[index].Type + "\n" +

"Цвет вещи:\t" + blazers[index].Color + "\n";

}

}

}

**MainWindow.xaml.cs**

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.Navigation;

using System.Windows.Shapes;

using Lab\_07;

using Color = Lab\_07.Color;

namespace Lab\_07\_WPF

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

static Pants[] pants =

{

new Pants(Gender.Man, 180, Type\_clothes.Cotton, Color.Black),

new Pants(Gender.Man, 190, Type\_clothes.Cotton, Color.Red),

new Pants(Gender.Woman, 160, Type\_clothes.Latex, Color.Green),

new Pants(Gender.Woman, 170, Type\_clothes.Latex, Color.Red),

};

static Skirts[] skirts =

{

new Skirts(Gender.Man, 180, Type\_clothes.Wool, Color.Red),

new Skirts(Gender.Man, 190, Type\_clothes.Wool, Color.Blue),

new Skirts(Gender.Woman, 150, Type\_clothes.Wool, Color.Black),

new Skirts(Gender.Woman, 160, Type\_clothes.Wool, Color.Red),

};

static Blazer[] blazers =

{

new Blazer(Gender.Man, 180, Type\_clothes.Leather, Color.Blue),

new Blazer(Gender.Man, 190, Type\_clothes.Leather, Color.Red),

new Blazer(Gender.Woman, 160, Type\_clothes.Leather, Color.Red),

new Blazer(Gender.Woman, 170, Type\_clothes.Leather, Color.Black),

};

static Costume[] costumesWithPants =

{

new Costume(pants[0],blazers[0]), //MAN

new Costume(pants[1],blazers[1]), //MAN

new Costume(pants[2],blazers[3]), //WOMAN

new Costume(pants[3],blazers[2]), //WOMAN

};

static Costume[] costumesWithSkirts =

{

new Costume(skirts[0],blazers[1]), //MAN

new Costume(skirts[1],blazers[0]), //MAN

new Costume(skirts[3],blazers[3]), //WOMAN

new Costume(skirts[3],blazers[2]), //WOMAN

};

public MainWindow()

{

InitializeComponent();

}

//ПОЛОСА ПРОКРУТА

// РАЗМЕР КНОПКИ И ТЕКСТ В 2 СТРОКИ

// МЕТОДЫ ПЕРЕОБРЕДЕЛИТЬ ДЛЯ РАБОТЫ

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

{

textBox1.Text = "";

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

{

textBox1.Text += Costume.ShowCostumeWithSkirts(costumesWithSkirts, i) + Environment.NewLine;

}

}

private void button1\_2\_Click(object sender, RoutedEventArgs e)

{

textBox1.Text = "";

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

{

textBox1.Text += Costume.ShowCostumeWithPants(costumesWithPants, i) + Environment.NewLine;

}

}

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

{

textBox1.Text = "";

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

{

textBox1.Text += Skirts.ShowSkirt(skirts, i);

}

}

private void button3\_Click(object sender, RoutedEventArgs e)

{

textBox1.Text = "";

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

{

textBox1.Text += Pants.ShowPants(pants, i);

}

}

private void button4\_Click\_1(object sender, RoutedEventArgs e)

{

textBox1.Text = "";

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

{

textBox1.Text += Blazer.ShowBlazers(blazers, i);

}

}

private void button5\_Click(object sender, RoutedEventArgs e)

{

textBox1.Text = "";

textBox1.Text += "1)Мужские косmюмы\n";

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

{

if (costumesWithPants[i].Pants.Gender == Gender.Man)

textBox1.Text += Costume.ShowCostumeWithPants(costumesWithPants, i);

}

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

{

if (costumesWithSkirts[i].Skirt.Gender == Gender.Man)

textBox1.Text += Costume.ShowCostumeWithSkirts(costumesWithSkirts, i);

}

}

private void button5\_2\_Click(object sender, RoutedEventArgs e)

{

textBox1.Text = "";

textBox1.Text += "2)Женские косmюмы\n";

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

{

if (costumesWithPants[i].Pants.Gender == Gender.Woman)

textBox1.Text += Costume.ShowCostumeWithPants(costumesWithPants, i);

}

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

{

if (costumesWithSkirts[i].Skirt.Gender == Gender.Woman)

textBox1.Text += Costume.ShowCostumeWithSkirts(costumesWithSkirts, i);

}

}

private void button5\_3\_Click(object sender, RoutedEventArgs e)

{

textBox1.Text = "";

textBox1.Text += "3)Женские косmюмы с брюками\n";

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

{

if ((costumesWithPants[i].Pants.Gender == Gender.Woman))

textBox1.Text += Costume.ShowCostumeWithPants(costumesWithPants, i);

}

}

private void button6\_Click(object sender, RoutedEventArgs e)

{

textBox1.Text = "";

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

{

if (pants[i].Color == Color.Red)

textBox1.Text += Pants.ShowPants(pants,i);

}

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

{

if (costumesWithPants[i].Pants.Color == Color.Red)

textBox1.Text += Costume.ShowCostumeWithPants(costumesWithPants,i);

}

}

private void button7\_Click(object sender, RoutedEventArgs e)

{

textBox1.Text = "";

textBox1.Text += Costume.ShowClothesAssortment(pants, skirts, blazers, costumesWithPants, costumesWithSkirts);

}

private void button8\_Click(object sender, RoutedEventArgs e)

{

Close();

}

}

}