**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

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

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

**ЛАБОРАТОРНАЯ РАБОТА №7**

по дисциплине: «Объектно-ориентированное программирование»

на тему: «Наследование классов»

Выполнил: студент гр. ИТП-21

Леди Д.Д.

Принял: преподаватель-стажер

Гуменников Е.Д.

Гомель 2022

**Цель работы:** изучить способы реализации *WPF.* Написать программу в которой реализовать принципы наследования в *C*#.

**Задание**

1. Необходимо решить задачу, согласно варианта (таблица 7)

2. При создании классов руководствоваться Code Convention

3. Весь код должен содержать элементы документирования

4. При реализации иерархии классов использовать механизм наследования

5. Разработать модульные тесты для верификации созданных классов

6. Классы должен быть размещены в библиотеке классов

7. Модульные тесты – в отдельном проекте

8. В отдельном проекте реализовать интерфейс WPF.

9. WPF-приложение должно обеспечивать ввод, редактирование и просмотр данных

**Вариант 3**

* 1. Разработать иерархию классов для хранения информации о продукции ателье, которое занимается пошивом брюк, юбок, пиджаков и костюмов, которые могут состоять как из брюк и пиджака, так и из юбки и пиджака. Вся одежда может быть как мужской, так и женской. Об одежде известно: размер (рост и полнота), тип ткани, цвет ткани.
  2. Создать не менее 15 объектов различной одежды.
  3. Вывести отдельно все женские костюмы, все мужские костюмы и все женские костюмы с брюками.
  4. Вывести все брюки красного цвета (в том числе и в костюмах)

**Ход Работы**

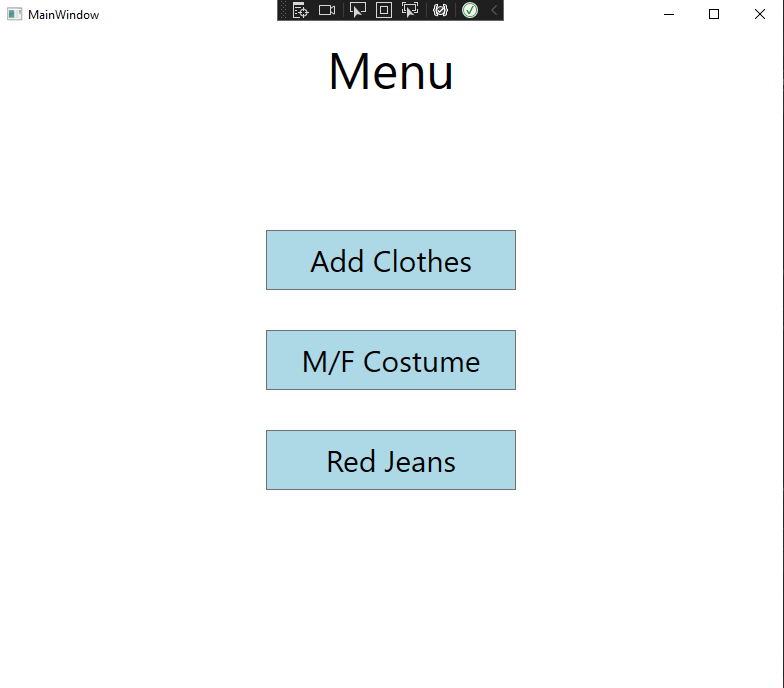


Рисунок 1 – Пример работы *MainWindow*

В *WindowResult* были реализованы поля для считывания данных и проверки их корректности. Пример работы *WindowResult* указан на рисунке 2.

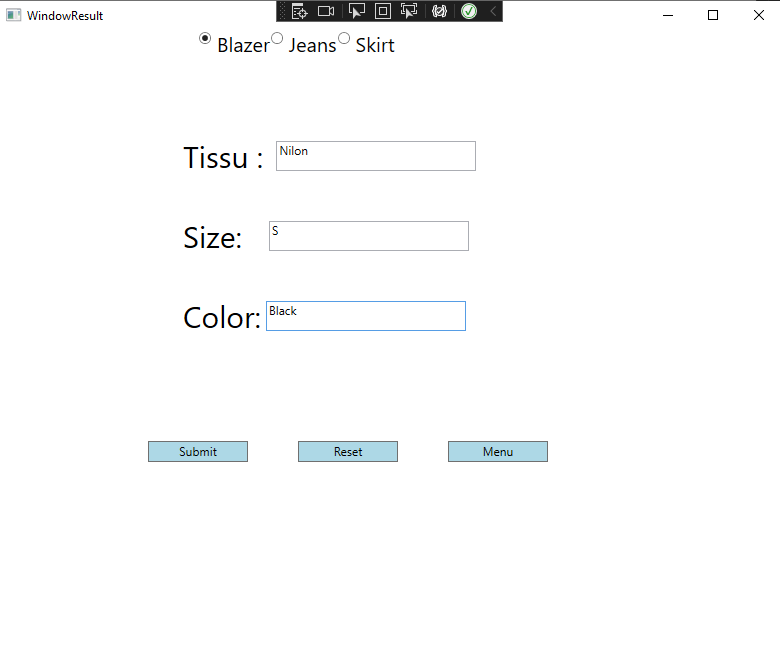
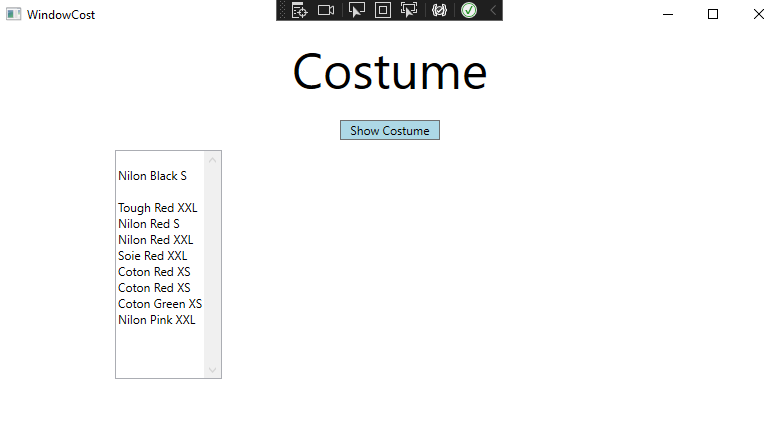


Рисунок 2 – Пример работы *WindowResult*

Для вывода всех работ была разработана *WindowCost* в которой выводятся все работы и информации о них. Пример работы *Form3* указан на рисунке 3



Для поиска Брюки красного цвета была разработана форма 4, в которой выбирается тип . Пример операции WindowRed показан на рисунке 4

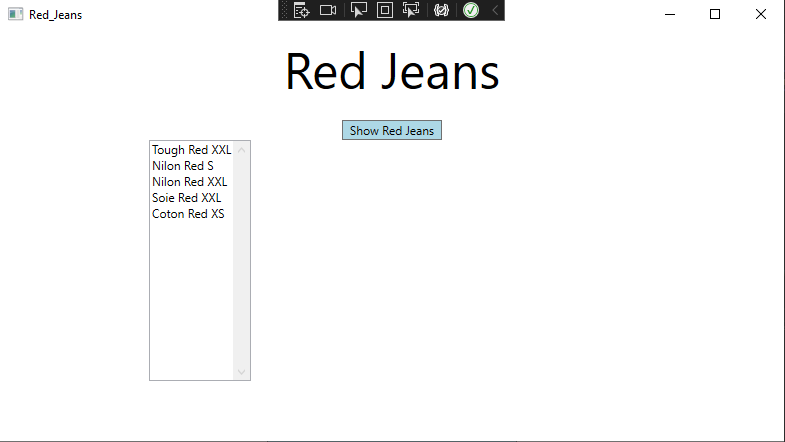


Рисунок 4 – Пример работы *WindowRed*

**Вывод:** в ходе выполнения лабораторной работы была изучена работа с *WPF.* Изучена такая концепция как наследование в языке программирования *C*#.

**Листинг класса *Clothes***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Controls;

namespace WpfApp1

{

public abstract class Clothes

{

public string Name { get; set; }

public string Color { get; set; }

public string Size { get; set; }

public string Tissu { get; set; }

}

}

**Листинг класса *MainWindow***

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;

namespace WpfApp1

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Button\_Add\_Click(object sender, RoutedEventArgs e)

{

var window = new WindowResult();

window.Show();

}

private void Button\_Costume\_Click(object sender, RoutedEventArgs e)

{

var window = new WindowCost();

window.Show();

}

private void Button\_Jeans\_Click(object sender, RoutedEventArgs e)

{

var window = new Red\_Jeans();

window.Show();

}

}

}