

 **КОЛЛЕДЖ МНОГОУРОВНЕВОГО ПРОФЕССИОНАЛЬНОГО**

**ОБРАЗОВАНИЯ**

Федеральное государственное бюджетное образовательное учреждение  
среднего профессионального образования

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ СЛУЖБЫ ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

КОЛЛЕДЖ МНОГОУРОВНЕВОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

**ОТЧЁТ ПО УЧЕБНОЙ ПРАКТИКЕ**

по профессиональному модулю

**ПМ.01 Разработка модулей программного обеспечения**

**Выполнил:**

Студент группы 31ИС-20

Глушков Евгений Сергеевич

**Проверила:**

Руководитель практики   
Кукшева Б.А.

**Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Москва 2022

# **Содержание**

[**Содержание** 2](#_Toc122511370)

[**Введение.** 3](#_Toc122511371)

[**Задание:** 4](#_Toc122511372)

[**Разработка программного продукта.** 4](#_Toc122511373)

[**Создание графического интерфейса для компьютерных систем** 4](#_Toc122511374)

[**Реализация функционала.** 9](#_Toc122511375)

[**Демонстрация результата.** 10](#_Toc122511376)

[**Программный код** 12](#_Toc122511377)

[Заключение. 12](#_Toc122511378)

# **Введение.**

Для того чтобы решить поставленную задачу нужно осознавать, о чем идет речь и понимать суть вопроса. Для этого необходимо знать основные понятия о программировании.

Программированием называется процесс создания компьютерных программ. Компьютерная программа является связкой многочисленных строк специального текста. Он является специальным, потому что создан таким образом, чтобы машине было понятно, какие действия должны быть выполнены.

Компьютерный код — это специальный текст, состоящий из набора пошаговых инструкций. Он не всегда содержит в себе нули и единицы, также в нём есть определённые слова и дополнительные символы. Компьютер считывает код, который сообщает ему, какие операции следует выполнить с данными. Для создания программ программист, должен уметь правильно давать команды компьютеру, для этого были придуманы специальные языки- языки программирования.

Язык программирования — формальная знаковая система, предназначенная для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, задающих внешний вид программ и действия, которые выполнит исполнитель (компьютер) под её управлением. Получается, что для разных задач нужен определенный язык программирования.

Передо мной стоит задача создания программ для компьютерных систем и интерфейса для удобной работы пользователя с программой. Для выполнения я решил использовать язык С#, так как он прост в освоении и позволяет гибко изменять интерфейс.

## **Задание:**

Создайте программу «MarathonSkills»: для участия спонсирование в марафоне. Пользователь сможет зарегистрироваться на марафон и спонсировать его.

Эта система будет одновременно работать как общественная система (например, позволяя людям, зарегистрироваться на марафон, узнать больше о марафоне, посмотреть результаты марафона) и закрытой части (например, позволяя персоналу управлять регистраций спортсменов, учитывать волонтеров).

# **Разработка программного продукта.**

**Создание графического интерфейса для компьютерных систем**

Для удобства работы с программой нужно создать формы, на одной будет главное меню из которого будет переход на остальные. Можно увидеть на рисунке №1.

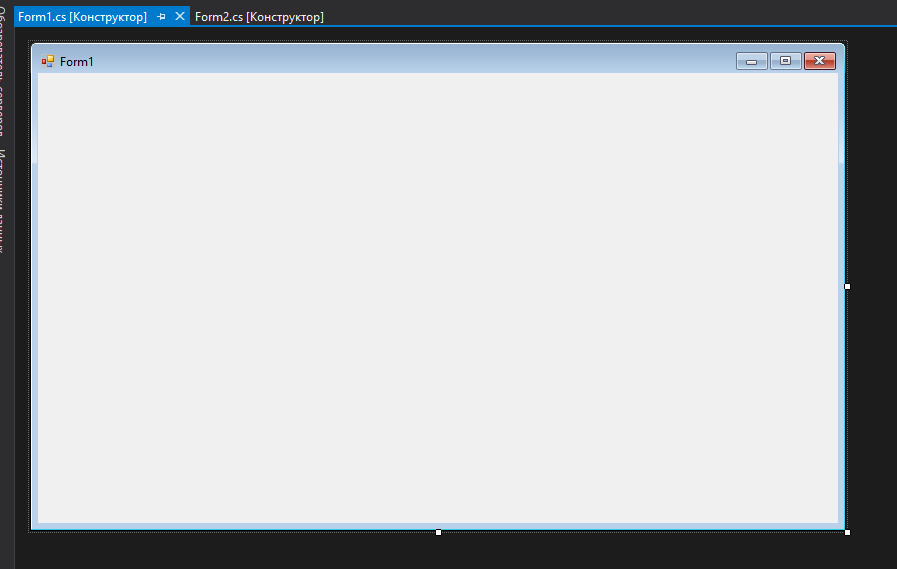


Рисунок 1. MainWindow

Размеры форм будем изменять под внутренние наполнение элементами. Добавим 4 кнопки и текстовые формы. После работы со свойствами MainWindow у нас получилась, уже готовая первая форма. Рисунок 2.

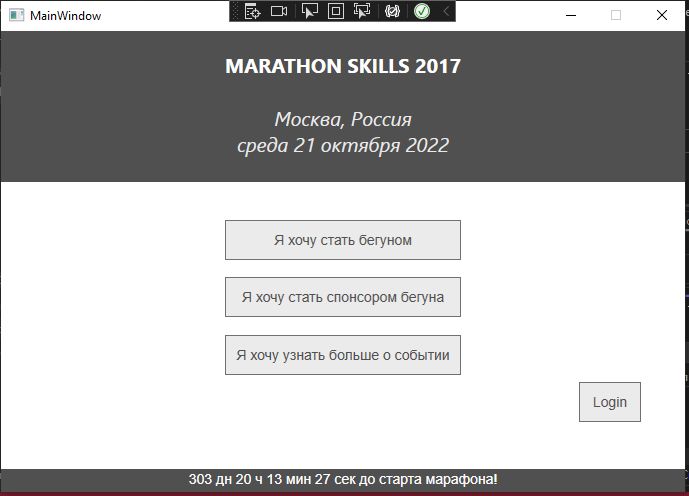


Рисунок 2. Готовая форма.

Теперь приступим к работе с SponsorPage формой. На ней нужно вывести textbox и combobox, чтобы добавить информацию о спонсоре и спонсировать бегуна. Создам поле и форму для текста, они нам понадобятся что бы спонсор смог выбрать сумму для пожертвования. Рисунок 3.

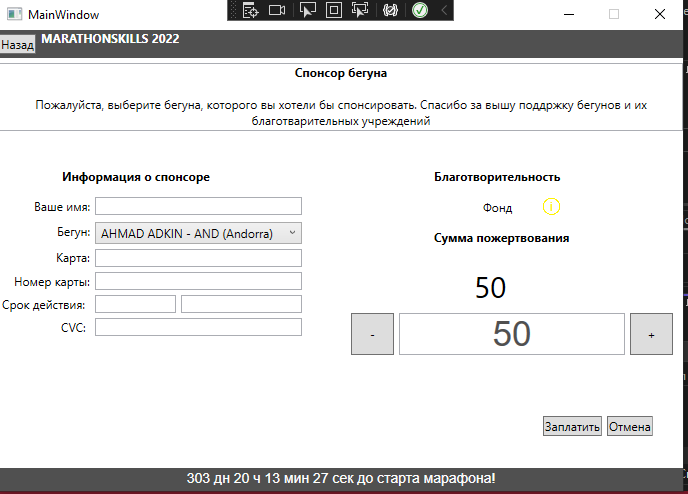
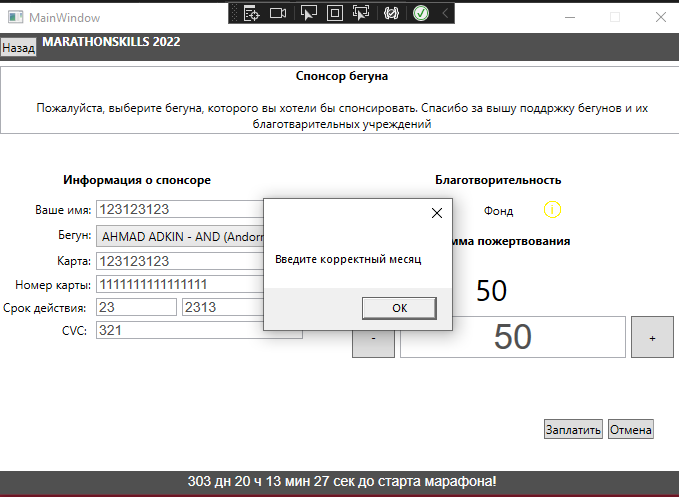


Рисунок 3. форма SpnsorPage

После нажатия кнопки “Заплатить” и данные заполнены не корректно, то будет выведена окно с ошибкой. В результате форма выглядит как на рисунке 4. Интерфейс программы готов.



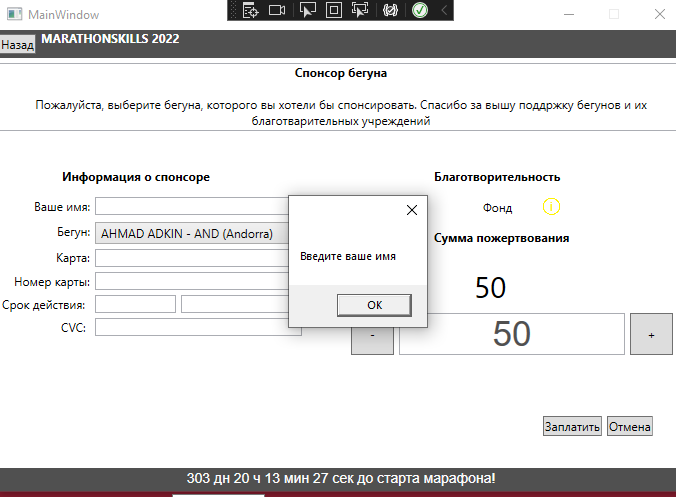


Рисунок 4. Вывод окна ошибки.

Дальше мы добавляем ещё одну форму с благодарностью за поддержку, где будет указано информация о бегуне и размер пожертвования. Рисунок 5.

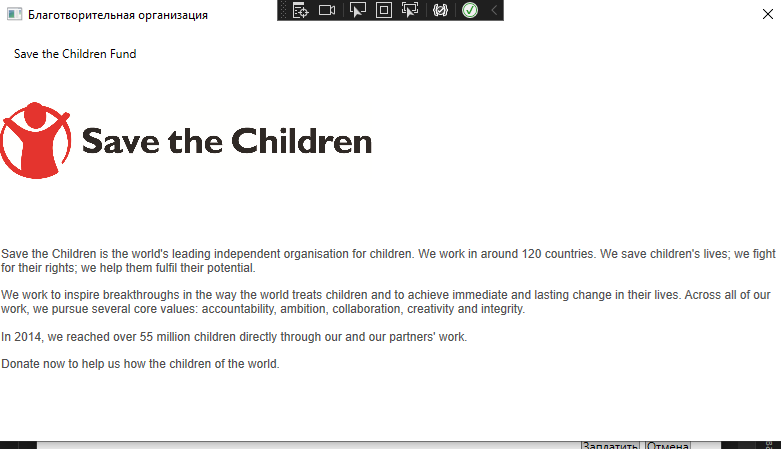


Рисунок 5. Готовая форма.

На главной форме реализуем кнопку “Хочу больше узнать о событии”

При нажатии которой будет выводится кнопка со списком благотворительных организаций. В ней будет находится информация о организациях, данные который загружаются из базы данных. Рисунок 6.

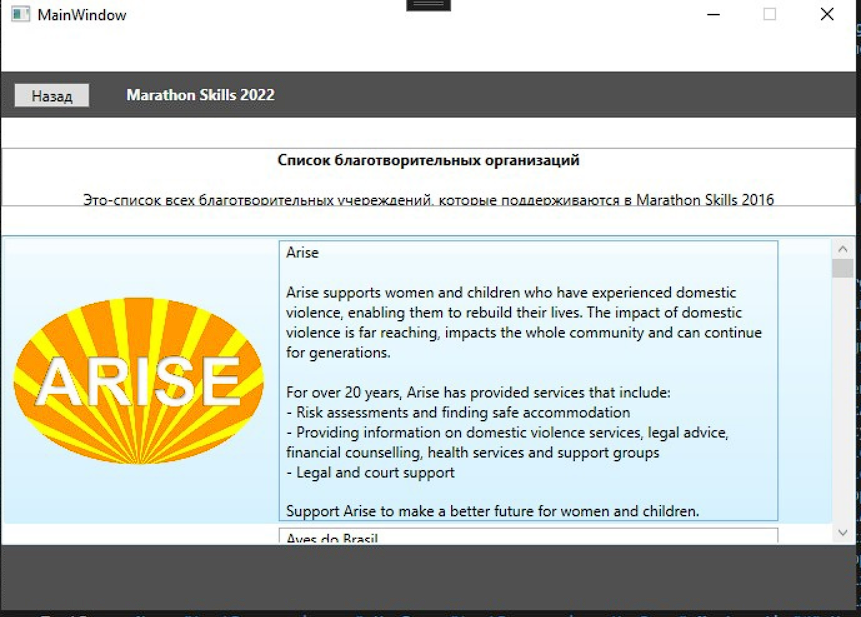


Рисунок 6. Форма с информацией.

# **Реализация функционала.**

Для работы программ нужно написать программный код, который будет выполнять компьютер. Первая часть программ будет находиться на первой форме MainPage. На добавлен таймер до марафона. Мы с помощью библиотеки “SystemTimer” написали код для таймера. Как показано на рисунке 7. Дальше 8



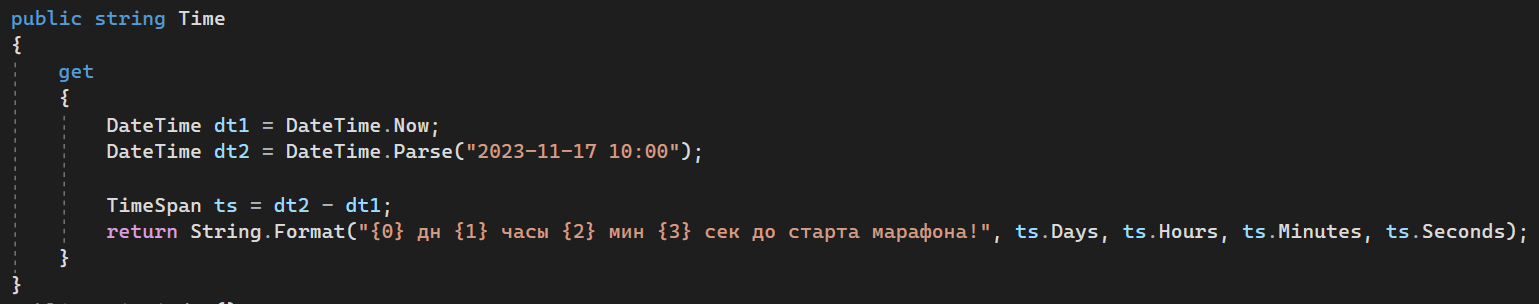
Рисунок 7. Библиотека.

Рисунок 8. Добавление таймера.

В форме SponsorPage подключаем базу данных с бегунами и фондами. Для начала нужно создать запросы в SQL и подключить базу данных в Visual Studio. С помощью кода получаем данные о бегуне. Рисунок 9.

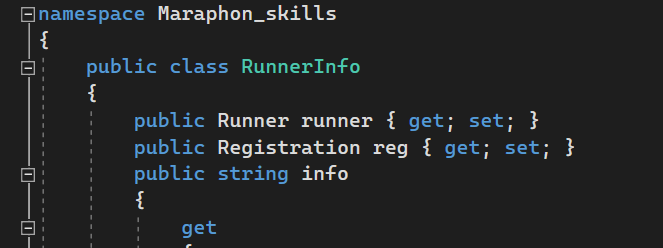


Рисунок 9. Код для вывода данных из базы данных.

Для реализации перехода между формами по кнопке используется функция NavigationService. Рисунок 10.

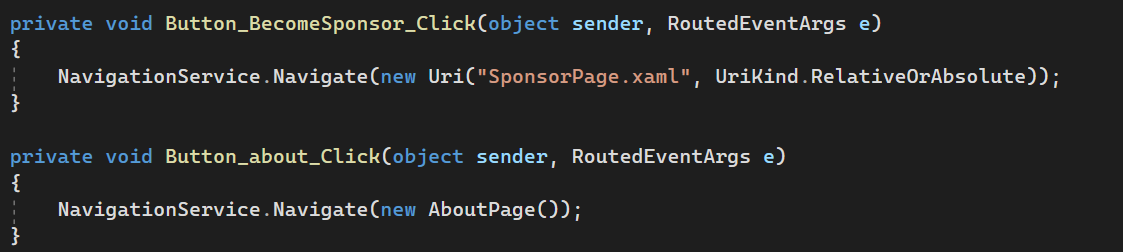


Рисунок 10. Реализация переходов между формами.

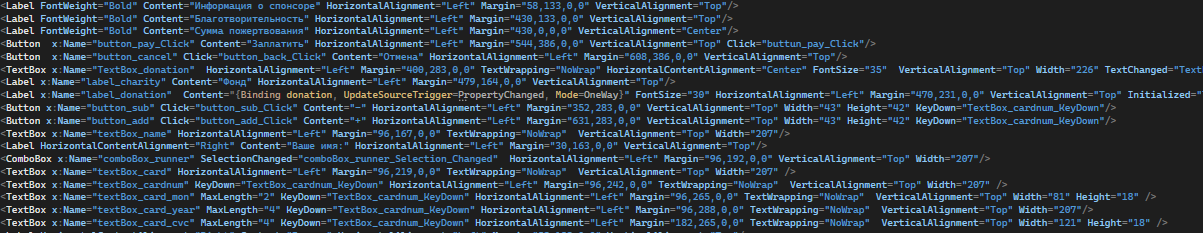


Рисунок 11. Оформление дизайна страницы пожертвований

В случае если программе на странице SponsorPage будет не хватать вводных данных будут выводиться диалоговые окна с пояснениями где требуеться проверить вводимые дынные



Рисунок 12. Вывод диалоговых окон

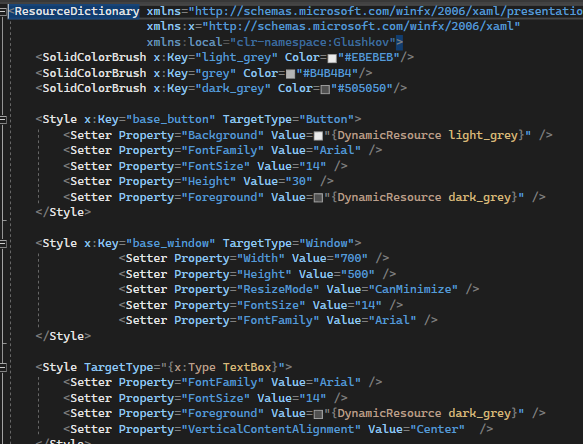


Рисунок 13. Присваивание определенных параметров к классам

# **Демонстрация результата.**

После написания программ, можно проверить ее работоспособность, для этого скомпилируем ее, нажав кнопку пуск в Visual Studio. У нас откроется наша первая форма. Рисунок10.

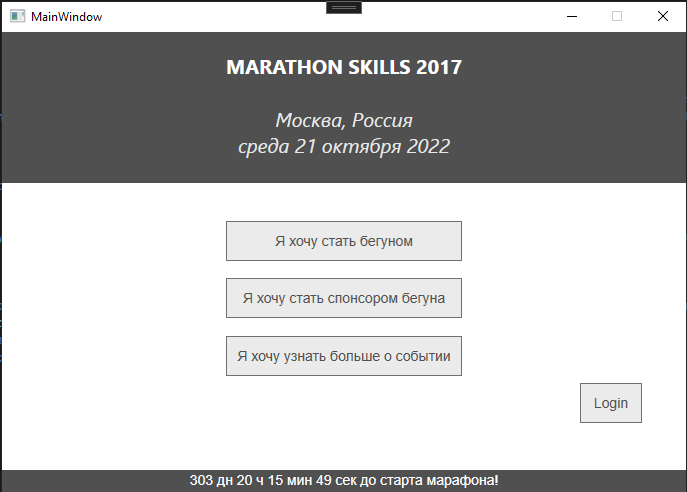


Рисунок 13.

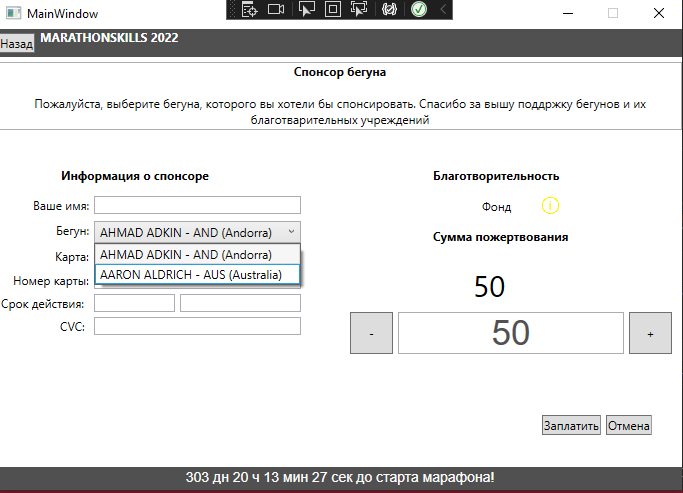


Рисунок14.

Как видим на Рисунке 14 бегуны из базы данных появились. Все кнопки и боксы кликабельны, а сумму пожертвования можно менять по кнопке.

# **Программный код**

MainWindow:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Timers;

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 Timer = System.Timers.Timer;

namespace Maraphon\_skills

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window, INotifyPropertyChanged

{

public string Time

{

get

{

DateTime dt1 = DateTime.Now;

DateTime dt2 = DateTime.Parse("2023-11-17 10:00");

TimeSpan ts = dt2 - dt1;

return String.Format("{0} дн {1} часы {2} мин {3} сек до старта марафона!", ts.Days, ts.Hours, ts.Minutes, ts.Seconds);

}

}

public MainWindow()

{

InitializeComponent();

this.DataContext = this;

}

~MainWindow()

{

Util.db.Database.Connection.Close();

}

public event PropertyChangedEventHandler PropertyChanged;

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

{

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

Timer tmr = new Timer();

tmr.Interval = 1000;

tmr.Elapsed += Tmr\_Elapsed;

tmr.Start();

}

private void Tmr\_Elapsed(object sender, ElapsedEventArgs e)

{

PropertyChange("Time");

}

private void PropertyChange(string name)

{

if (PropertyChanged != null)

PropertyChanged(this, new PropertyChangedEventArgs(name));

}

private void frame\_Navigated(object sender, NavigationEventArgs e)

{

}

}

}

SponsorPage:

using System;

using System.Collections.Generic;

using System.Collections.Specialized;

using System.ComponentModel;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

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 Maraphon\_skills

{

/// <summary>

/// Логика взаимодействия для SponsorPage.xaml

/// </summary>

public partial class SponsorPage : Page, INotifyPropertyChanged

{

private int donate = 50;

public int donation

{

get

{

return donate;

}

set

{

int temp = 0;

int.TryParse(value.ToString(), out temp);

if (temp >= 10)

{

donate = value;

PropertyChange("donation");

}

}

}

public SponsorPage()

{

InitializeComponent();

}

public event PropertyChangedEventHandler PropertyChanged;

private void PropertyChange(string name)

{

if (PropertyChanged != null)

PropertyChanged(this, new PropertyChangedEventArgs(name));

}

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

{

NavigationService.GoBack();

}

private void Page\_Loaded(object sender, RoutedEventArgs e)

{

List<RunnerInfo> runnerInfo = new List<RunnerInfo>();

Util.db.Registration.ToList().ForEach(r => runnerInfo.Add(new RunnerInfo() { runner = r.Runner, reg = r }));

comboBox\_runner.ItemsSource = runnerInfo;

comboBox\_runner.DisplayMemberPath = "info";

this.DataContext = this;

comboBox\_runner.SelectedIndex = 0;

}

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

{

if (donation >= 20)

{

donation -= 10;

}

}

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

{

donation += 10;

}

private void comboBox\_runner\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

RunnerInfo runner = comboBox\_runner.SelectedItem as RunnerInfo;

Registration reg = runner.runner.Registration.SingleOrDefault();

//if (reg != null)

{

Charity charity = reg.Charity;

label\_charity.Content = charity.CharityName;

}

}

private void Image\_MouseDown(object sender, MouseButtonEventArgs e)

{

RunnerInfo runner = comboBox\_runner.SelectedItem as RunnerInfo;

Registration reg = runner.runner.Registration.SingleOrDefault();

if (reg != null)

{

new wndCharityInfo(reg.Charity).ShowDialog();

}

}

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

{

if (string.IsNullOrWhiteSpace(textBox\_name.Text) || textBox\_name.Text.Length <= 0)

{

MessageBox.Show("Введите ваше имя");

return;

}

if (string.IsNullOrWhiteSpace(textBox\_card.Text) || textBox\_card.Text.Length <= 0)

{

MessageBox.Show("Введите владельца карты");

return ;

}

if (string.IsNullOrWhiteSpace(textBox\_cardnum.Text) || textBox\_cardnum.Text.Length != 16)

{

MessageBox.Show("Введите номер карты");

return;

}

if (string.IsNullOrWhiteSpace(textBox\_card\_mon.Text) || textBox\_card\_mon.Text.Length < 1)

{

MessageBox.Show("Введите месяц");

return;

}

if(int.Parse(textBox\_card\_mon.Text) < 1 || int.Parse(textBox\_card\_mon.Text) > 12)

{

MessageBox.Show("Введите корректный месяц");

return;

}

if (string.IsNullOrWhiteSpace(textBox\_card\_year.Text) || textBox\_card\_year.Text.Length != 4)

{

MessageBox.Show("Введиет год");

return;

}

if (string.IsNullOrWhiteSpace(textBox\_card\_cvc.Text) || textBox\_card\_cvc.Text.Length != 3)

{

MessageBox.Show("Введите CVC");

return;

}

Sponsorship sponsor = new Sponsorship();

sponsor.Amount = donation;

sponsor.Registration = (comboBox\_runner.SelectedItem as RunnerInfo).reg;

sponsor.SponsorName = textBox\_name.Text;

try

{

Util.db.Sponsorship.Add(sponsor);

Util.db.SaveChanges();

}

catch

{

MessageBox.Show("Произошла ошибка при добавлении");

}

finally

{

NavigationService.Navigate(new SponsorshipSuccessPage((comboBox\_runner.SelectedItem as RunnerInfo), sponsor.Registration.Charity, donation));

}

}

private void textBox\_cardnum\_KeyDown(object sender, KeyEventArgs e)

{

if (!new Regex("[0-9]").IsMatch(e.Key.ToString()))

e.Handled = true;

}

}

}

# Заключение.

В результате выполнения учебной практики был написано приложение «MarathonSkills» для администрирования и автоматизации марафоном для бегунов. В последствии для улучшения работоспособности программ можно сделать ее более простой и понятной для любого пользователя. Создать руководство пользователя , которое будет направлять пользователя в использовании программы.