Учреждение образования

«Гродненский государственный политехнический колледж»

**ОТЧЁТ  
ПО УЧЕБНОЙ ПРАКТИКЕ ПО ПРОГРАММИРОВАНИЮ**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Учащегося |  | 3 | | | | |  | курса, группы |  | ПЗТ-40 |
| специальности | | |  | 2 - 40 01 01 «Программное обеспечение информационных | | | | | | |
|  | | |  | технологий» | | | | | | |
| Место прохождения практики | | | | |  | УО «Гродненский государственный | | | | |
|  | | | | |  | политехнический колледж» | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тема проекта: |  | «Разработка приложения «Компьютерные вирусы» » | | |
|  |  |  | | |
| Ссылка на проект: | | |  |  |
|  | | |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Выполнил |  |  |  | С.В. Ситкевич |
|  |  |  |  | (инициалы, фамилия) |
| Руководитель практики от колледжа |  |  |  | В.В. Хомич |
|  |  |  | (инициалы, фамилия) |

Содержание

[Введение. Описание структуры предприятия 3](#_Toc194324354)

[1 Анализ предметной области и формулировка требований к программе 4](#_Toc194324355)

[1.1 Исследование предметной области 4](#_Toc194324356)

[1.2 Инструменты разработки 4](#_Toc194324357)

[2 Проектирование](#_Toc194324358) 6

[2.1 Диаграмма вариантов использования](#_Toc194324359) 6

[2.2 Диаграмма деятельности](#_Toc194324360) 7

[2.3 Описание тестов](#_Toc194324361) 8

[3 Построение программы 10](#_Toc194324362)

[4 Тестирование 11](#_Toc194324363)

[5 Применение 12](#_Toc194324364)

[5.1 Назначение и условия применения программы 12](#_Toc194324365)

[5.2 Инсталляция 12](#_Toc194324366)

[5.3 Выполнение программы](#_Toc194324367) 14

[Заключение](#_Toc194324368) 24

[Список использованных источников](#_Toc194324369) 25

[Приложение A Листинг программы](#_Toc194324370) 26

Приложение A  
Листинг программы

MaimWindow.xaml(главная страница):

<Window x:Class="virus1.MainWindow" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:virus1"

mc:Ignorable="d"

Title=" " Height="600" Width="802" ResizeMode="NoResize" WindowStyle="None" WindowStartupLocation="CenterScreen">

<Window.Background>

<ImageBrush ImageSource="/image1/fonglav.jpg" Stretch="UniformToFill" Opacity="0.6" />

</Window.Background>

<Grid>

<Grid.RowDefinitions>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<Border Grid.Row="1" Grid.Column="1" Background="DarkRed" CornerRadius="10,0,0,0">

<Button Width="200" Height="200" BorderThickness="0" Background="Transparent" Click="Obych\_Click" VerticalAlignment="Center" HorizontalAlignment="Center" >

<Button.ToolTip>

<Border Background="LightCoral" CornerRadius="10" BorderThickness="2" BorderBrush="Brown" Padding="10">

<StackPanel>

<TextBlock Text="Обучение и информация" Foreground="Brown" FontWeight="Bold" FontSize="16" />

</StackPanel>

</Border>

</Button.ToolTip>

<Image Source="/obych.png" Stretch="Fill" Height="109" Width="122" />

</Button>

</Border>

<Border Grid.Row="1" Grid.Column="2" Background="DarkRed" CornerRadius="0,10,0,0">

<Button Width="200" Height="100" BorderThickness="0" Background="Transparent" Click="Button\_Click" VerticalAlignment="Center" HorizontalAlignment="Center" >

<Button.ToolTip>

<Border Background="LightCoral" CornerRadius="10" BorderThickness="2" BorderBrush="Brown" Padding="10">

<StackPanel>

<TextBlock Text="Тесты" Foreground="Brown" FontWeight="Bold" FontSize="16" />

</StackPanel>

</Border>

</Button.ToolTip>

<Image Source="/test-removebg-preview.png" Stretch="Fill" Height="111" Width="122" />

</Button>

</Border>

<Border Grid.Row="2" Grid.Column="1" Background="DarkRed" CornerRadius="0,0,0,0">

<Button Width="200" Height="300" BorderThickness="0" Background="Transparent" Click="Sovet\_Click" VerticalAlignment="Center" HorizontalAlignment="Center" >

<Button.ToolTip>

<Border Background="LightCoral" CornerRadius="10" BorderThickness="2" BorderBrush="Brown" Padding="10">

<StackPanel>

<TextBlock Text="Советы и рекомендации" Foreground="Brown" FontWeight="Bold" FontSize="16" />

</StackPanel>

</Border>

</Button.ToolTip>

<Image Source="/sovet-removebg-preview.png" Stretch="Fill" Height="103" Width="90" RenderTransformOrigin="0.514,0.558" Margin="0,-10,0,0" />

</Button>

</Border>

MaimWindow.xaml.сы(главная страница):

<Border Grid.Row="2" Grid.Column="2" Background="DarkRed" CornerRadius="0,0,0,0">

<Button Width="200" Height="200" BorderThickness="0" Background="Transparent" Click="Analiz\_Click" VerticalAlignment="Center" HorizontalAlignment="Center" >

<Button.ToolTip>

<Border Background="LightCoral" CornerRadius="10" BorderThickness="2" BorderBrush="Brown" Padding="10">

<StackPanel>

<TextBlock Text="Анализ файлов" Foreground="Brown" FontWeight="Bold" FontSize="16" />

</StackPanel>

</Border>

</Button.ToolTip>

<Image Source="/analiz-removebg-preview.png" Stretch="Fill" Height="103" Width="120" RenderTransformOrigin="0.512,0.768" VerticalAlignment="Bottom" Margin="0,0,0,-20" />

</Button>

</Border>

<Border Grid.Row="3" Grid.Column="1" Background="DarkRed" CornerRadius="0,0,0,10">

<Button Width="200" Height="200" BorderThickness="0" Background="Transparent" Click="Simul\_Click" VerticalAlignment="Center" HorizontalAlignment="Center" >

<Button.ToolTip>

<Border Background="LightCoral" CornerRadius="10" BorderThickness="2" BorderBrush="Brown" Padding="10">

<StackPanel>

<TextBlock Text="Симуляция атак" Foreground="Brown" FontWeight="Bold" FontSize="16" />

</StackPanel>

</Border>

</Button.ToolTip>

<Image Source="/simul-removebg-preview.png" Stretch="Fill" Height="95" Width="138" RenderTransformOrigin="0.512,0.768" VerticalAlignment="Bottom" />

</Button>

</Border>

<Border Grid.Row="3" Grid.Column="2" Background="DarkRed" CornerRadius="0,0,10,0">

<Button Width="200" Height="200" BorderThickness="0" Background="Transparent" Click="Vexod\_Click" VerticalAlignment="Center" HorizontalAlignment="Center" >

<Button.ToolTip>

<Border Background="LightCoral" CornerRadius="10" BorderThickness="2" BorderBrush="Brown" Padding="10">

<StackPanel>

<TextBlock Text="Выход" Foreground="Brown" FontWeight="Bold" FontSize="16" />

</StackPanel>

</Border>

</Button.ToolTip>

<Image Height="76" Source="/vexod.png" Stretch="Fill" Width="78" RenderTransformOrigin="0.526,0.283"/>

</Button>

</Border>

<Label Content="КОМПЬЮТЕРНЫЕ ВИРУСЫ "

HorizontalAlignment="Left"

VerticalAlignment="Top"

Grid.ColumnSpan="4"

Margin="172,18,0,0"

FontFamily="JetBrains Mono ExtraBold"

FontWeight="Bold"

FontSize="40"

Width="542"

Foreground="#FFDCA3A3" />

</Grid>

</Window>

MainWindow.xaml.cs(главная страница):

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 virus1

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

TaskWindow taskWindow = new TaskWindow();

taskWindow.Show();

this.Close();

}

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

{

TestWindow testWindow = new TestWindow();

testWindow.Show();

this.Close();

}

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

{

SovetWindow sovetWindow = new SovetWindow();

sovetWindow.Show();

this.Close();

}

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

{

Analiz analizWindow = new Analiz();

analizWindow.Show();

this.Close();

}

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

{

PlayWindow playWindow = new PlayWindow();

playWindow.Show();

this.Close();

}

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

{

this.Close(); // Закрытие текущего окна

}

}

}

Analiz.xaml(форма «Анализ файлов»):

<Window x:Class="virus1.Analiz"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:virus1"

mc:Ignorable="d"

WindowStyle="None"

Height="600" Width="802" ResizeMode="NoResize" WindowStartupLocation="CenterScreen" >

<Grid>

<Grid.Background>

<ImageBrush ImageSource="/image1/12.jpg" Stretch="UniformToFill"/>

</Grid.Background>

<ScrollViewer>

<StackPanel Margin="10" Height="632">

<TextBlock Text="Анализ файлов на наличие вирусов" FontFamily="JetBrains Mono NL ExtraBold" FontSize="24" Foreground="#FFDCA3A3" TextAlignment="Center" Margin="0,10,0,20" />

<TextBlock TextWrapping="Wrap" Text="На этой странице вы можете анализировать файлы на наличие вредоносного ПО. Обратите внимание на подозрительные признаки:

1) Большой размер файла;

2) Расширения .exe, .bat или другие исполняемые файлы;

3) Непривычные названия с подозрительными словами, такими как virus, hack, crack." FontFamily="JetBrains Mono NL ExtraLight" Foreground="#FFDCA3A3" FontSize="14" Margin="10,0,10,20" /> <GroupBox Header="Как распознать заражённый файл" Foreground="#FFDCA3A3" FontFamily="JetBrains Mono NL ExtraBold" Background="Firebrick" Height="114"> <StackPanel Margin="10,0,10,0" Height="58"> <TextBlock TextWrapping="Wrap" Margin="0,0,0,5" Text="1. Файлы со странными или длинными именами." /> <TextBlock TextWrapping="Wrap" Margin="0,0,0,5" Text="2. Файлы с неожиданно большим размером или нестандартными расширениями." /> <TextBlock TextWrapping="Wrap" Text="3. Файлы, которые вы получили из неизвестных источников." Height="17" />

</StackPanel>

</GroupBox>

<GroupBox Header="Популярные антивирусные решения" Foreground="#FFDCA3A3" FontFamily="JetBrains Mono NL ExtraBold" Background="Firebrick">

<StackPanel >

<TextBlock TextWrapping="Wrap" Margin="0,0,0,5" Text="1. Microsoft Defender Antivirus." /> <TextBlock TextWrapping="Wrap" Margin="0,0,0,5" Text="2. Kaspersky Anti-Virus." /> <TextBlock TextWrapping="Wrap" Margin="0,0,0,5" Text="3. ESET NOD32 Antivirus." /> <TextBlock TextWrapping="Wrap" Margin="0,0,0,5"

Text="1. Avast Free Antivirus." />

</StackPanel>

</GroupBox>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Margin="0,0,0,10"> <Button Content="Выбрать файл" Width="150" Height="30" Click="SelectFileButton\_Click" Foreground="#FFDCA3A3" FontFamily="JetBrains Mono NL ExtraLight"> <Button.Background> <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0"> <GradientStop Color="#FF675656"/> <GradientStop Color="#FF8D1717" Offset="1"/> <GradientStop Color="#FF793737" Offset="0.372"/> <GradientStop Color="#FF7C3131" Offset="0.651"/> </LinearGradientBrush> </Button.Background>

</Button>

<Button Content="Анализировать файл" Width="150" Height="30" Click="AnalyzeFileButton\_Click" Margin="5" Foreground="#FFDCA3A3" FontFamily="JetBrains Mono NL ExtraLight"> <Button.Background> <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0"> <GradientStop Color="#FF675656"/> <GradientStop Color="#FF8D1717" Offset="1"/> <GradientStop Color="#FF793737" Offset="0.372"/> <GradientStop Color="#FF7C3131" Offset="0.651"/> </LinearGradientBrush> </Button.Background>

</Button>

</StackPanel>

<TextBox x:Name="FilePathTextBox" Width="400" Height="30" Margin="5" IsReadOnly="True" HorizontalAlignment="Center" />

<TextBlock x:Name="ResultTextBlock" Text="Результат анализа появится здесь" TextWrapping="Wrap" FontFamily="JetBrains Mono NL ExtraLight" Foreground="#FFDCA3A3" FontSize="14" TextAlignment="Center" Height="60" /> <Button Width="75" Height="75" Margin="0,0,600,0" Background="DarkRed" Click="Nazad\_Click" > <Image Height="65" Source="/image1/18.png" Stretch="Fill" Width="65" RenderTransformOrigin="0.511,0.928" />

</Button>

</StackPanel>

</ScrollViewer>

<Image Height="100" Width="100" Margin="649,142,53,342"/>

</Grid>

</Window>

Analiz.xaml.cs(форма «Анализ файлов»):

using Microsoft.Win32;

using System;

using System.Collections.Generic;

using System.IO;

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 virus1

{

// Определение класса Analiz, который представляет окно анализа файлов

public partial class Analiz : Window

{

// Конструктор окна Analiz, вызывающий метод инициализации компонентов

public Analiz()

{

InitializeComponent();

}

// Метод обработки события нажатия кнопки выбора файла

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

{

// Создание объекта диалога для выбора файлов

OpenFileDialog openFileDialog = new OpenFileDialog();

// Открытие диалога и проверка, был ли выбран файл

if (openFileDialog.ShowDialog() == true)

{

// Отображение пути выбранного файла в текстовом поле

FilePathTextBox.Text = openFileDialog.FileName;

}

}

// Метод обработки события нажатия кнопки анализа файла

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

{

// Получение пути файла из текстового поля

string filePath = FilePathTextBox.Text;

// Проверка, выбран ли файл

if (string.IsNullOrEmpty(filePath))

{

// Вывод сообщения об ошибке, если файл не выбран

MessageBox.Show("Выберите файл для анализа!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

// Создание объекта FileInfo для получения информации о файле

FileInfo fileInfo = new FileInfo(filePath);

// Формирование строки с основными характеристиками файла

string result = $"Анализ файла: {fileInfo.Name}\nРазмер: {fileInfo.Length} байт\n";

// Проверка расширения файла на потенциально опасные форматы (исполняемые файлы)

if (fileInfo.Extension == ".exe" || fileInfo.Extension == ".bat")

{

result += "Предупреждение: файл исполняемый, возможен риск заражения!\n";

}

else

{

result += "Файл безопасен.";

}

// Отображение результата анализа в текстовом блоке

ResultTextBlock.Text = result;

}

// Метод обработки события нажатия кнопки "Назад"

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

{

// Создание нового объекта главного окна и его отображение

MainWindow mainWindow = new MainWindow();

mainWindow.Show();

// Закрытие текущего окна

this.Close();

}

}

}

SovetWindow.xaml(форма «Советы и рекомендации»):

<Window x:Class="virus1.SovetWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:virus1"

mc:Ignorable="d"

WindowStyle="None"

Title="Советы и рекомендации" Height="600" Width="830" ResizeMode="NoResize" WindowStartupLocation="CenterScreen">

<Grid Margin="0,0,0,-6">

<Grid.Background>

<ImageBrush ImageSource="/image1/12.jpg" Stretch="UniformToFill"/>

</Grid.Background>

<!-- Прокручиваемое содержимое -->

<ScrollViewer HorizontalScrollBarVisibility="Disabled" VerticalScrollBarVisibility="Auto" Margin="10,0,0,10">

<StackPanel>

<!-- Заголовок страницы -->

<TextBlock Text="СОВЕТЫ И РЕКОМЕНДАЦИИ"

FontFamily="JetBrains Mono NL ExtraBold" FontSize="24"

TextAlignment="Center"

Foreground="#FFDCA3A3" Margin="0,10,0,20" />

<TextBlock Text="Советы по защите"

FontFamily="JetBrains Mono NL ExtraBold" FontSize="16"

Foreground="#FFDCA3A3"

Width="174" Height="27" RenderTransformOrigin="0.5,0.5">

<TextBlock.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform Angle="0.262"/>

<TranslateTransform/>

</TransformGroup>

</TextBlock.RenderTransform>

</TextBlock>

<!-- Подзаголовок -->

<ListBox Margin="10" RenderTransformOrigin="0.5,0.5" Height="208">

<ListBox.Background>

<LinearGradientBrush StartPoint="0,0" EndPoint="1,1">

<GradientStop Color="#FF671102" Offset="0"/>

<GradientStop Color="#FF3C0606" Offset="1"/>

</LinearGradientBrush>

</ListBox.Background>

<ListBox.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform/>

<TranslateTransform X="0" Y="-3"/>

</TransformGroup>

</ListBox.RenderTransform>

<!-- Главный пункт -->

<ListBoxItem FontFamily="JetBrains Mono NL ExtraLight" Background="DarkRed" Height="202" Foreground="#FFBE93A3">

<StackPanel Margin="10,5,0,0" Width="678" Height="193">

<!-- Вложенные элементы -->

<TextBlock FontWeight="Bold" Text="1.Установите антивирусное ПО:" FontFamily="JetBrains Mono NL ExtraBold" Foreground="#FFBE93A3" />

<TextBlock Text="• Выберите проверенную антивирусную программу." TextWrapping="Wrap" Margin="20,0,0,0" />

<TextBlock Text="• Регулярно обновляйте базу данных вирусов, чтобы защититься от новых угроз." TextWrapping="Wrap" Margin="20,0,0,0" />

<TextBlock FontWeight="Bold" Text="2. Обновляйте системы и программы:" FontFamily="JetBrains Mono NL ExtraBold" Foreground="#FFBE93A3" />

<TextBlock Text="• Убедитесь, что операционная система и установленное ПО всегда актуальны." TextWrapping="Wrap" Margin="20,0,0,0" />

<TextBlock Text="• Устаревшие версии могут содержать уязвимости, которые используют вирусы." TextWrapping="Wrap" Margin="20,0,0,0" />

<TextBlock FontWeight="Bold" Text="3. Используйте надёжные пароли:" FontFamily="JetBrains Mono NL ExtraBold" Foreground="#FFBE93A3" />

<TextBlock Text="• Создавайте длинные и сложные пароли." TextWrapping="Wrap" Margin="20,0,0,0" />

<TextBlock Text="• Не используйте один и тот же пароль для нескольких аккаунтов." TextWrapping="Wrap" Margin="20,0,0,0" />

<TextBlock FontWeight="Bold" Text="3. Будьте осторожны с подозрительными файлами:" FontFamily="JetBrains Mono NL ExtraBold" Foreground="#FFBE93A3" Height="16" />

<TextBlock Text="• Не открывайте вложения в письмах от неизвестных отправителей." TextWrapping="Wrap" Margin="20,0,0,0" />

<TextBlock Text="• Всегда проверяйте скачанные файлы антивирусом." TextWrapping="Wrap" Margin="20,0,0,0" />

</StackPanel>

</ListBoxItem>

</ListBox>

<TextBlock Text="Пошаговый план действий при заражении"

FontFamily="JetBrains Mono NL ExtraBold" FontSize="16"

FontWeight="Bold" Foreground="#FFBE93A3"

Width="370"/>

<ListBox Margin="10" RenderTransformOrigin="0.5,0.5">

<ListBox.Background>

<LinearGradientBrush StartPoint="0,0" EndPoint="1,1">

<GradientStop Color="#FF671102" Offset="0"/>

<GradientStop Color="#FF3C0606" Offset="1"/>

</LinearGradientBrush>

</ListBox.Background>

<ListBox.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform/>

<TranslateTransform X="0" Y="-3"/>

</TransformGroup>

</ListBox.RenderTransform>

<!-- Главный пункт -->

<ListBoxItem FontFamily="JetBrains Mono NL ExtraLight" Background="DarkRed" Height="140" Foreground="#FFBE93A3">

<StackPanel Margin="10,5,0,0" Width="678" Height="193">

<!-- Вложенные элементы -->

<TextBlock FontWeight="Bold" Text="1. Отключите Wi-Fi или кабельное соединение:" FontFamily="JetBrains Mono NL ExtraBold" />

<TextBlock Text="• Найдите настройки сети на компьютере и отключите активное соединение." TextWrapping="Wrap" Margin="20,0,0,0"/>

<TextBlock Text="• Если используется кабель Ethernet, физически отключите его." TextWrapping="Wrap" Margin="20,0,0,0"/>

<TextBlock FontWeight="Bold" Text="2. Изолируйте заражённое устройство:" FontFamily="JetBrains Mono NL ExtraBold" />

<TextBlock Text="• Убедитесь, что заражённый компьютер не подключен к другим устройствам (например, через Bluetooth или локальную сеть)." TextWrapping="Wrap" Margin="20,0,0,0"/>

<TextBlock FontWeight="Bold" Text="3. Ограничьте взаимодействие с внешними устройствами:" FontFamily="JetBrains Mono NL ExtraBold" />

<TextBlock Text="• Избегайте подключения внешних носителей, таких как USB-накопители, пока система не будет защищена." TextWrapping="Wrap" Margin="20,0,0,0"/>

</StackPanel>

</ListBoxItem>

</ListBox>

<TextBlock Text="Часто задаваемые вопросы"

FontFamily="JetBrains Mono NL ExtraBold" FontSize="16"

FontWeight="Bold" Foreground="#FFBE93A3"

Width="230" Height="29"/>

<ListBox Margin="10" RenderTransformOrigin="0.5,0.5" Height="295" Width="751">

<ListBox.Background>

<LinearGradientBrush StartPoint="0,0" EndPoint="1,1">

<GradientStop Color="#FF671102" Offset="0"/>

<GradientStop Color="#FF3C0606" Offset="1"/>

</LinearGradientBrush>

</ListBox.Background>

<ListBox.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform/>

<TranslateTransform X="0" Y="-3"/>

</TransformGroup>

</ListBox.RenderTransform>

<!-- Главный пункт -->

<ListBoxItem FontFamily="JetBrains Mono NL ExtraLight" Background="DarkRed" Height="296" Width="745" Foreground="#FFBE93A3">

<StackPanel Margin="10,0,0,0" Width="678" Height="297">

<!-- Вложенные элементы -->

<!-- Вложенные элементы -->

<!-- Вложенные элементы -->

<TextBlock FontWeight="Bold" Text="1.Как вирусы попадают на компьютер?" FontFamily="Cascadia Code PL" Height="20" Width="684" />

<TextBlock Text="Вирусы могут попасть через:" TextWrapping="Wrap" Margin="20,0,0,0"/>

<TextBlock Text="• Заражённые вложения в письмах." TextWrapping="Wrap" Margin="20,0,0,0"/>

<TextBlock Text="• Подозрительные ссылки." TextWrapping="Wrap" Margin="20,0,0,0"/>

<TextBlock Text="• Скачивание файлов с ненадёжных сайтов." TextWrapping="Wrap" Margin="20,0,0,0"/>

<TextBlock Text="• Уязвимости в программном обеспечении." TextWrapping="Wrap" Margin="20,0,0,0"/>

<TextBlock FontWeight="Bold" Text="2. Может ли антивирус полностью защитить меня от вирусов?" FontFamily="Cascadia Mono PL SemiBold" />

<TextBlock Text="• Антивирус значительно снижает риск заражения, но важно также соблюдать базовые правила безопасности, такие как осторожное поведение в интернете." TextWrapping="Wrap" Margin="20,0,0,0"/>

<TextBlock FontWeight="Bold" Text="3. Какие последствия может вызвать вирус?" FontFamily="Cascadia Mono PL SemiBold" Height="20" />

<TextBlock Text="• Уничтожать или повреждать файлы." TextWrapping="Wrap" Margin="20,0,0,0"/>

<TextBlock Text="• Замедлять работу компьютера." TextWrapping="Wrap" Margin="20,0,0,0"/>

<TextBlock Text="• Красть конфиденциальную информацию." TextWrapping="Wrap" Margin="20,0,0,0"/>

<TextBlock Text="• Включать ваш компьютер в сеть ботнетов для атак." TextWrapping="Wrap" Margin="20,0,0,0"/>

<TextBlock FontWeight="Bold" Text="4. Как я могу узнать, что мой компьютер заражён вирусом??" FontFamily="Cascadia Mono PL SemiBold" Height="20" />

<TextBlock Text="• Медленная работа системы." TextWrapping="Wrap" Margin="20,0,0,0"/>

<TextBlock Text="• Неожиданные всплывающие окна." TextWrapping="Wrap" Margin="20,0,0,0"/>

<TextBlock Text="• Автоматическое изменение настроек." TextWrapping="Wrap" Margin="20,0,0,0"/>

<TextBlock Text="• Удаление или повреждение файлов." TextWrapping="Wrap" Margin="20,0,0,0"/>

</StackPanel>

</ListBoxItem>

</ListBox>

<TextBlock Text="Советы по безопасному серфингу в интернете"

FontFamily="JetBrains Mono NL ExtraBold" FontSize="16"

FontWeight="Bold" Foreground="#FFBE93A3"

Width="374" Height="29"/>

<ListBox Margin="10" RenderTransformOrigin="0.5,0.5" Height="119" Width="738">

<ListBox.Background>

<LinearGradientBrush StartPoint="0,0" EndPoint="1,1">

<GradientStop Color="#FF671102" Offset="0"/>

<GradientStop Color="#FF3C0606" Offset="1"/>

</LinearGradientBrush>

</ListBox.Background>

<ListBox.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform/>

<TranslateTransform X="0" Y="-3"/>

</TransformGroup>

</ListBox.RenderTransform>

<!-- Главный пункт -->

<ListBoxItem FontFamily="JetBrains Mono NL ExtraLight" Background="DarkRed" Height="122" Width="733" Foreground="#FFBE93A3">

<StackPanel Margin="10,0,0,0" Width="678" Height="296" RenderTransformOrigin="0.5,0.5">

<StackPanel.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform AngleY="0.101"/>

<RotateTransform/>

<TranslateTransform Y="0.598"/>

</TransformGroup>

</StackPanel.RenderTransform>

<!-- Вложенные элементы -->

<!-- Вложенные элементы -->

<!-- Вложенные элементы -->

<TextBlock FontFamily="Cascadia Code PL" Height="30" Width="714" ><Run Text="1. Использование надёжных браузеров с функциями защиты (например, блокировкой вредоносных "/><LineBreak/><Run Text="сайтов)."/></TextBlock>

<TextBlock Text="2. Осторожность при переходе по ссылкам — особенно в письмах, соцсетях и на форумах." FontFamily="Cascadia Mono PL" Width="683" />

<TextBlock FontFamily="Cascadia Mono PL" Height="28" Width="702" ><Run Text="3. Включение защиты против фишинга в браузере (например, в Chrome и Edge есть встроенные "/><LineBreak/><Run Text="механизмы)."/></TextBlock>

<TextBlock Text="4. Избегать подозрительных сайтов, предлагающих бесплатные загрузки или взломанные программы." FontFamily="Cascadia Mono PL" Height="20" />

</StackPanel>

</ListBoxItem>

</ListBox>

<TextBlock Text="Правила загрузки файлов"

FontFamily="JetBrains Mono NL ExtraBold" FontSize="16"

FontWeight="Bold" Foreground="#FFBE93A3"

Width="241" Height="29" Margin="0,11,0,0"/>

<ListBox Margin="10" RenderTransformOrigin="0.5,0.5" Height="78" Width="742">

<ListBox.Background>

<LinearGradientBrush StartPoint="0,0" EndPoint="1,1">

<GradientStop Color="#FF671102" Offset="0"/>

<GradientStop Color="#FF3C0606" Offset="1"/>

</LinearGradientBrush>

</ListBox.Background>

<ListBox.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform/>

<TranslateTransform X="0" Y="-3"/>

</TransformGroup>

</ListBox.RenderTransform>

<!-- Главный пункт -->

<ListBoxItem Background="DarkRed" Height="73" Width="737" Foreground="#FFBE93A3">

<StackPanel Margin="10,0,0,0" Width="678" Height="296" RenderTransformOrigin="0.5,0.5">

<StackPanel.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform AngleY="0.101"/>

<RotateTransform/>

<TranslateTransform Y="0.598"/>

</TransformGroup>

</StackPanel.RenderTransform>

<!-- Вложенные элементы -->

<!-- Вложенные элементы -->

<!-- Вложенные элементы -->

<TextBlock FontFamily="JetBrains Mono NL ExtraBold" Height="19" Width="714" ><Run Language="ru-ru" Text="1. "/><Run Text="Загружать файлы только с официальных сайтов разработчиков."/></TextBlock>

<TextBlock Text="2. Проверять расширение файлов перед открытием (например, .exe и .bat могут быть опасными)." FontFamily="JetBrains Mono NL ExtraBold" Width="683" />

<TextBlock FontFamily="JetBrains Mono NL ExtraBold" Height="18" Width="702" Text="3. Перед установкой сканировать файлы антивирусом." />

<TextBlock Text="4. Никогда не загружать ПО с торрент-трекеров и ненадежных ресурсов." FontFamily="JetBrains Mono NL ExtraBold" Height="20" />

</StackPanel>

</ListBoxItem>

</ListBox>

<TextBlock Text="Настройка брандмауэра и защиты ОС"

FontFamily="JetBrains Mono NL ExtraBold" FontSize="16"

FontWeight="Bold" Foreground="#FFBE93A3"

Width="333" Height="29" Margin="0,10,0,0"/>

<ListBox Margin="10" RenderTransformOrigin="0.5,0.5" Height="78" Width="739">

<ListBox.Background>

<LinearGradientBrush StartPoint="0,0" EndPoint="1,1">

<GradientStop Color="#FF671102" Offset="0"/>

<GradientStop Color="#FF3C0606" Offset="1"/>

</LinearGradientBrush>

</ListBox.Background>

<ListBox.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform/>

<TranslateTransform X="0" Y="-3"/>

</TransformGroup>

</ListBox.RenderTransform>

<!-- Главный пункт -->

<ListBoxItem Background="DarkRed" Height="73" Width="735" Foreground="#FFBE93A3">

<StackPanel Width="678" Height="79" RenderTransformOrigin="0.5,0.5">

<StackPanel.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform AngleY="0.101"/>

<RotateTransform/>

<TranslateTransform Y="0.598"/>

</TransformGroup>

</StackPanel.RenderTransform>

<!-- Вложенные элементы -->

<!-- Вложенные элементы -->

<!-- Вложенные элементы -->

<TextBlock FontFamily="JetBrains Mono NL ExtraBold" Height="19" Width="714" ><Run Language="ru-ru" Text="1. "/><Run Text="Включить брандмауэр Windows для блокировки подозрительных подключений."/></TextBlock>

<TextBlock Text="2. Регулярно обновлять систему и антивирус, чтобы закрывать уязвимости." FontFamily="JetBrains Mono NL ExtraBold" Width="683" />

<TextBlock FontFamily="JetBrains Mono NL ExtraBold" Height="18" Width="702" Text="3. Ограничить автоматический запуск подозрительных программ." />

<TextBlock Text="4. Использовать VPN для защиты при работе в публичных Wi-Fi сетях." FontFamily="JetBrains Mono NL ExtraBold" Height="20" />

</StackPanel>

</ListBoxItem>

</ListBox>

<TextBlock Text="Полезные ссылки "

FontFamily="JetBrains Mono NL ExtraBold" FontSize="16"

FontWeight="Bold" Foreground="#FFBE93A3"

Width="156" Height="29" Margin="0,10,0,0"/>

<TextBlock FontSize="16" Margin="30,0,0,0">

<Hyperlink NavigateUri="https://nsportal.ru/ap/library/drugoe/2024/03/13/individualnyy-proekt-raznovidnosti-kompyuternyh-virusov-i-metody" RequestNavigate="Hyperlink\_RequestNavigate">

Разновидности компьютерных вирусов и методы защиты

</Hyperlink>

</TextBlock>

<TextBlock FontSize="16" Margin="30,0,0,0" Foreground="#FFBE93A3">

<Hyperlink NavigateUri="https://cyberleninka.ru/article/n/vidy-kompyuternyh-virusov-sposoby-zaschity-kompyutera-ot-virusov" RequestNavigate="Hyperlink\_RequestNavigate">

Виды компьютерных вирусов и способы защиты

</Hyperlink> </TextBlock>

<TextBlock FontSize="16" Margin="30,0,0,0" Foreground="#FFBE93A3">

<Hyperlink NavigateUri="https://www.kaspersky.ru/resource-center/preemptive-safety/antivirus-software-how-to-choose" RequestNavigate="Hyperlink\_RequestNavigate">

Как выбрать надежный антивирус

</Hyperlink>

</TextBlock>

<TextBlock FontSize="16" Margin="30,0,0,0" Foreground="#FFBE93A3">

<Hyperlink NavigateUri="https://hi-tech.mail.ru/review/59912-top-10-free-antiviruses/#anchor169329133228388311"

RequestNavigate="Hyperlink\_RequestNavigate">

Топ-10 лучших бесплатных антивирусов 2025 года

</Hyperlink>

</TextBlock>

<Button Width="74" Height="75" Margin="0,20,650,0" Click="Nazad\_Click" Background="DarkRed">

<Image Height="65" Source="/image1/18.png" Stretch="Fill" Width="74" />

</Button>

<StackPanel Height="50">

</StackPanel>

</StackPanel>

</ScrollViewer>

</Grid>

</Window>

SovetWindow.xaml.cs(форма «Советы и рекомендации»):

using System;

using System.Text;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Media;

using System.Windows.Shapes;

using System.Windows.Threading;

namespace virus1

{

public partial class TaskWindow : Window

{

private DispatcherTimer timer; // Таймер для симуляции распространения инфекции

private Random rand = new Random(); // Генератор случайных значений

private int infectedCount = 0; // Количество зараженных узлов

private const int TotalNodes = 10; // Общее количество узлов в сети

// Конструктор TaskWindow, инициализирует UI и запускает симуляцию

public TaskWindow()

{

InitializeComponent();

StartSimulation(); // Запуск симуляции при создании окна

}

/// Метод StartSimulation очищает канвас и создает новые узлы.

private void StartSimulation()

{ NetworkCanvas.Children.Clear(); // Очищаем канвас перед новой симуляцией

infectedCount = 0; // Сбрасываем количество зараженных узлов

// Создаем узлы на канвасе

for (int i = 0; i < TotalNodes; i++)

{

Ellipse node = new Ellipse

{

Width = 30,

Height = 30,

Fill = Brushes.White, // Узел по умолчанию белый

Stroke = Brushes.Gray, // Серый контур

StrokeThickness = 2

};

// Случайное размещение узла на канвасе

double x = rand.Next(50, 700);

double y = rand.Next(50, 500);

Canvas.SetLeft(node, x);

Canvas.SetTop(node, y);

NetworkCanvas.Children.Add(node); // Добавляем узел на канвас

}

// Настраиваем таймер, который будет распространять вирус

timer = new DispatcherTimer { Interval = TimeSpan.FromSeconds(1) };

timer.Tick += SpreadInfection; // Привязываем обработчик события

timer.Start(); // Запускаем таймер

}

/// <summary>

/// Метод SpreadInfection заражает узлы по очереди, создавая соединения между ними.

/// </summary>

private void SpreadInfection(object sender, EventArgs e)

{

// Если все узлы уже заражены, перезапускаем симуляцию

if (infectedCount >= TotalNodes)

{

timer.Stop(); // Останавливаем таймер

StartSimulation(); // Запускаем новую симуляцию

return;

}

// Получаем текущий узел для заражения

Ellipse node = NetworkCanvas.Children[infectedCount] as Ellipse;

if (node != null)

{

node.Fill = Brushes.Red; // Меняем цвет узла на красный, показывая заражение

// Создаем соединение с предыдущим узлом, если это не первый узел

if (infectedCount > 0)

{

Ellipse previousNode = NetworkCanvas.Children[infectedCount - 1] as Ellipse;

if (previousNode != null)

{

Line connection = new Line

{

Stroke = Brushes.Red, // Красная линия для зараженного соединения StrokeThickness = 2, X1 = Canvas.GetLeft(node) + 15, Y1 = Canvas.GetTop(node) + 15, X2 = Canvas.GetLeft(previousNode) + 15, Y2 = Canvas.GetTop(previousNode) + 15 }; NetworkCanvas.Children.Add(connection); // Добавляем соединение на канвас

}

}

}

infectedCount++; // Увеличиваем счетчик зараженных узлов

}

/// <summary>

/// Метод Button\_Click возвращает пользователя в главное окно.

/// </summary>

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

{

MainWindow mainWindow = new MainWindow();

mainWindow.Show();

this.Close();

}

/// <summary>

/// Метод OpenPage открывает страницу с информацией о вирусе ILOVEYOU.

/// </summary>

private void OpenPage(object sender, RoutedEventArgs e)

{ OpenForm2("https://ru.wikipedia.org/wiki/ILOVEYOU");

}

/// <summary>

/// Метод OpenForm2 создает новое окно PageWindow для отображения веб-страницы.

/// </summary>

private void OpenForm2(string url)

{

PageWindow form2 = new PageWindow(url);

form2.Show();

}

// Методы открытия страниц с описаниями известных вирусов

private void WannaCry(object sender, RoutedEventArgs e)

{ OpenForm2("https://ru.wikipedia.org/wiki/WannaCry");

}

private void MyDoom(object sender, RoutedEventArgs e)

{ OpenForm2("https://ru.wikipedia.org/wiki/Mydoom");

}

private void Stuxnet(object sender, RoutedEventArgs e)

{ OpenForm2("https://ru.wikipedia.org/wiki/Stuxnet");

}

private void NotPetya(object sender, RoutedEventArgs e)

{ OpenForm2("https://ru.wikipedia.org/wiki/Petya\_ransomware");

}

private void SQL(object sender, RoutedEventArgs e)

{

OpenForm2("https://ru.wikipedia.org/wiki/SQL\_Slammer");

}

//Метод Nazad\_Click возвращает пользователя в главное окно.

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

{

MainWindow mainWindow = new MainWindow();

mainWindow.Show();

this.Close();

}

}

}

TestWindow.xaml (страница с тестом):

<Window x:Class="virus1.TestWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:virus1"

mc:Ignorable="d"

WindowStyle="None"

Title="Тест" Height="630" Width="802" ResizeMode="NoResize" WindowStartupLocation="CenterScreen">

<Canvas Name="MyCanvas">

<Image Source="image1/12.jpg" Stretch="Fill" Panel.ZIndex="-1" Height="630" Width="802" HorizontalAlignment="Center" VerticalAlignment="Top" OpacityMask="Black"/>

<Label x:Name="ScoreText" FontSize="16" Canvas.Left="14" Canvas.Top="16" HorizontalAlignment="Center" VerticalAlignment="Top" Content="Количество правильных ответов 0/10" Height="64" Width="367" Foreground="#FFDCA3A3" FontFamily="JetBrains Mono ExtraBold"/>

<Label Name="questionsOrder" FontSize="18" Canvas.Left="426" Canvas.Top="16" HorizontalAlignment="Left" VerticalAlignment="Center" Height="32" Width="325" Foreground="#FFDCA3A3" FontFamily="JetBrains Mono ExtraBold"></Label>

<Image Name="qImage" Source="image1/1.jpg" Stretch="Uniform" Height="257" Width="426" Canvas.Left="197" Canvas.Top="69" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<TextBlock x:Name="txtQuestions" TextAlignment="Center" Padding="3" FontSize="16" Width="611" TextWrapping="Wrap" Canvas.Left="110" Canvas.Top="349" HorizontalAlignment="Center" VerticalAlignment="Top" Text="Добро пожаловать в тест по компьютерным вирусам!" Foreground="#FFDCA3A3" FontFamily="JetBrains Mono ExtraBold"/>

<Button Background="DarkBlue" FontSize="14" Click="checkAnswer" Name="ans1" Width="544" Height="38" Content="Ответ: 1" Tag="0" Canvas.Left="129" Canvas.Top="395" HorizontalAlignment="Left" VerticalAlignment="Top" Foreground="#FFDCA3A3" FontFamily="JetBrains Mono ExtraBold"/>

<Button Background="DarkBlue" FontSize="14" Click="checkAnswer" Name="ans2" Width="544" Height="39" Content="Ответ: 2" Tag="0" Canvas.Left="129" Canvas.Top="442" HorizontalAlignment="Center" VerticalAlignment="Top" Foreground="#FFDCA3A3" FontFamily="JetBrains Mono ExtraBold"/>

<Button Background="DarkBlue" FontSize="14" Click="checkAnswer" Name="ans3" Width="544" Height="36" Content="Ответ: 3" Tag="0" Canvas.Left="129" Canvas.Top="490" HorizontalAlignment="Center" VerticalAlignment="Top" Foreground="#FFDCA3A3" FontFamily="JetBrains Mono ExtraBold"/>

<Image Height="167" Canvas.Left="70" Source="/13.png" Stretch="Fill" Canvas.Top="192" Width="206" RenderTransformOrigin="0.5,0.5" HorizontalAlignment="Left" VerticalAlignment="Top">

<Image.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform Angle="-0.489"/>

<TranslateTransform/>

</TransformGroup>

</Image.RenderTransform>

</Image>

<Image Height="112" Canvas.Left="189" Source="/image1/15 (1).png" Stretch="Fill" Canvas.Top="250" Width="124" RenderTransformOrigin="0.5,0.5">

<Image.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform Angle="0.012"/>

<TranslateTransform/>

</TransformGroup>

</Image.RenderTransform>

</Image>

<Image Height="121" Canvas.Left="532" Source="/image1/16.png" Stretch="Fill" Canvas.Top="25" Width="120" HorizontalAlignment="Left" VerticalAlignment="Top" RenderTransformOrigin="0.5,0.5">

<Image.RenderTransform>

<TransformGroup>

<ScaleTransform ScaleY="-1" ScaleX="-1"/>

<SkewTransform/>

<RotateTransform Angle="180.503"/>

<TranslateTransform/>

</TransformGroup>

</Image.RenderTransform>

</Image>

<Image Height="135" Canvas.Left="516" Source="/image1/17.png" Stretch="Fill" Canvas.Top="224" Width="147"/>

<Button Canvas.Left="30" Canvas.Top="516" Height="71" Width="71" HorizontalAlignment="Center" VerticalAlignment="Top" Foreground="#FFE02222" Click="Button\_Click">

<Image Source="image1/18.png" Stretch="Fill" Height="63" Width="66" />

</Button>

</Canvas>

</Window>

TestWindow.xaml.cs (страница с тестом):

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using static virus1.TestWindow;

namespace virus1

{

public partial class TestWindow : Window

{

// Список номеров вопросов

List<int> questionNumbers = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

int qNum = 0; // Номер текущего вопроса

int i; // Хранит номер вопроса после его извлечения

int score; // Количество правильных ответов

// Конструктор окна тестирования

public TestWindow()

{

InitializeComponent();

StartGame(); // Инициализация игры

NextQuestion(); // Загрузка первого вопроса

}

//Проверяет правильность ответа, обновляет счет и загружает следующий вопрос.

private void checkAnswer(object sender, RoutedEventArgs e)

{

Button senderButton = sender as Button;

// Если тег кнопки равен "1", значит, это правильный ответ

if (senderButton.Tag.ToString() == "1")

{

score++; // Увеличиваем количество правильных ответов

}

// Предотвращаем отрицательный номер вопроса

if (qNum < 0)

{

qNum = 0;

}

else

{

qNum++; // Переход к следующему вопросу

}

// Обновляем счетчик правильных ответов

ScoreText.Content = " Правильный ответ " + score + "/" + questionNumbers.Count;

NextQuestion(); // Загружаем новый вопрос

}

// Сбрасывает прогресс игры и начинает тест заново.

private void RestartGame()

{

score = 0;

qNum = -1;

i = 0;

StartGame();

}

// Загружает следующий вопрос и обновляет UI.

private void NextQuestion()

{

// Если еще есть вопросы, загружаем следующий

if (qNum < questionNumbers.Count)

{

i = questionNumbers[qNum];

}

else

{

RestartGame(); // Если вопросов больше нет, перезапускаем тест

}

// Сбрасываем цвет кнопок и теги ответов

foreach (var x in MyCanvas.Children.OfType<Button>())

{

x.Tag = "0";

x.Background = Brushes.Maroon;

}

// Выбираем вопрос в зависимости от номера

switch (i)

{

case 1:

txtQuestions.Text = "Что делает компьютерный вирус?";

ans1.Content = "Защищает компьютер";

ans2.Content = "Повышает скорость работы";

ans3.Content = "Нарушает работу системы";

ans3.Tag = "1"; // Правильный ответ

qImage.Source = new BitmapImage(new Uri("pack://application:,,,/image1/1.jpg"));

break;

case 2:

txtQuestions.Text = "Как называется вирус, который требует выкуп?";

ans1.Content = "Руткит";

ans2.Content = "Вымогатель";

ans3.Content = "Троян";

ans2.Tag = "1"; // Правильный ответ

qImage.Source = new BitmapImage(new Uri("pack://application:,,,/image1/2.jpg"));

break;

case 3:

txtQuestions.Text = "Что такое ILOVEYOU?";

ans1.Content = "Компьютерный вирус, который шифрует файлы и требует выкуп";

ans2.Content = "Вредоносный скрипт, распространяющийся через электронную почту";

ans3.Content = "Программа для защиты компьютера от угроз";

ans2.Tag = "1"; // Правильный ответ

qImage.Source = new BitmapImage(new Uri("pack://application:,,,/image1/3.jpg"));

break;

case 4:

txtQuestions.Text = "Какую цель преследовал вирус MyDoom?";

ans1.Content = "Создание ботнета для проведения DDoS-атак";

ans2.Content = "Шифрование данных и требование выкупа";

ans3.Content = "Автоматическое обновление операционных систем";

ans1.Tag = "1"; // Правильный ответ

qImage.Source = new BitmapImage(new Uri("pack://application:,,,/image1/4.jpg"));

break;

// (Добавлены аналогичные блоки для оставшихся вопросов)

case 10:

txtQuestions.Text = "Какой вирус использовал уязвимость EternalBlue?";

ans1.Content = "ILOVEYOU";

ans2.Content = "Melissa";

ans3.Content = "WannaCry";

ans3.Tag = "1"; // Правильный ответ

qImage.Source = new BitmapImage(new Uri("pack://application:,,,/image1/9.jpg"));

break;

}

}

// Перемешивает список вопросов для случайного порядка.

private void StartGame()

{

var randomList = questionNumbers.OrderBy(a => Guid.NewGuid()).ToList();

questionNumbers = randomList;

questionsOrder.Content = null;

// Выводим порядок вопросов для отладки

for (int i = 0; i < questionNumbers.Count; i++)

{

questionsOrder.Content += " " + questionNumbers[i].ToString();

}

}

// Метод Button\_Click возвращает пользователя в главное окно.

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

{

MainWindow mainWindow = new MainWindow();

mainWindow.Show();

this.Close();

}

}

}

PlayWindow.xaml (страница «Симуляция атак»):

<Window x:Class="virus1.PlayWindow" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:virus1"

mc:Ignorable="d"

WindowStyle="None"

Title="Симуляция атак" Height="610" Width="865" ResizeMode="NoResize" WindowStartupLocation="CenterScreen">

<Window.Resources>

<!-- Градиентный фон -->

<!-- Стили для кнопок -->

<Style TargetType="Button">

<Setter Property="Foreground" Value="White"/>

<Setter Property="Background" Value="#440000"/>

<Setter Property="FontSize" Value="16"/>

<Setter Property="Width" Value="200"/>

<Setter Property="Height" Value="40"/>

<Setter Property="BorderThickness" Value="2"/>

<Setter Property="BorderBrush" Value="Red"/>

<Setter Property="FontWeight" Value="Bold"/>

</Style>

<!-- Неоновый стиль текста -->

<Style TargetType="TextBlock">

<Setter Property="FontFamily" Value="JetBrains Mono ExtraBold"/>

<Setter Property="Foreground" Value="#FFDCA3A3"/>

</Style>

</Window.Resources>

<Grid Margin="0,0,0,0">

<!-- Лог атак -->

<TextBox Name="AttackLog" IsReadOnly="True" Background="Black"

Foreground="Lime" TextWrapping="Wrap"

VerticalScrollBarVisibility="Auto" Margin="42,101,343,151"/>

<!-- Кнопки атаки и защиты -->

<Button Name="StartAttackButton" Content="Запустить атаку"

Width="NaN" Margin="557,212,100,315"

Click="StartAttack\_Click"/>

<!-- Выбор типа защиты -->

<ComboBox Name="DefenseOptions" Margin="557,297,100,238" RenderTransformOrigin="0.585,-2.913">

<ComboBoxItem Content="Антивирусное сканирование"/>

<ComboBoxItem Content="Файрвол блокирует атаку"/>

<ComboBoxItem Content="Восстановление системы"/>

<ComboBoxItem Content="Изоляция заражённых файлов"/>

<ComboBoxItem Content="Глубокое антивирусное сканирование"/>

<ComboBoxItem Content="Отключение зараженной сети"/>

<ComboBoxItem Content="Обновление системы"/>

<ComboBoxItem Content="Блокрировка подозрительных доменов"/>

<ComboBoxItem Content="Очистка вредоносных расширений"/>

<ComboBoxItem Content="Ограничение исходящих подключений "/>

</ComboBox>

<Button Name="ApplyDefenseButton" Content="Применить защиту"

Width="NaN" Margin="557,395,100,127" Click="ApplyDefense\_Click"/>

<!-- Таймер -->

<TextBlock Name="TimerText" Text="⏳ Осталось: 20 сек"

Foreground="#FF1A1A08" FontSize="16" Margin="554,104,64,429" RenderTransformOrigin="0.5,0.5">

<TextBlock.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform AngleX="2.08"/>

<RotateTransform/>

<TranslateTransform X="0.564"/>

</TransformGroup>

</TextBlock.RenderTransform>

</TextBlock>

<ProgressBar Name="AttackProgressBar"

Maximum="20" Value="20" Margin="557,156,10,417"/>

<TextBlock Text="Симуляция вирусных атак" Foreground="#FFDCA3A3" FontFamily="JetBrains Mono ExtraBold" FontSize="24" HorizontalAlignment="Left" Margin="282,25,0,508" Grid.ColumnSpan="2" Width="348"/>

<Button Width="75" Height="75" Click="Nazad\_Click" Margin="0,450,0,0" >

<Image Height="64" Source="/image1/18.png" Stretch="Fill" Width="65" RenderTransformOrigin="0.585,0.586" />

</Button>

</Grid>

</Window>

PlayWindow.xaml.cs(страница «Симуляция атак»):

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows; // Основное пространство имен WPF

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media; // Работа с графическим интерфейсом

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using System.Windows.Threading; // Позволяет использовать таймеры внутри WPF

namespace virus1

{

public partial class PlayWindow : Window

{

// Таймеры для различных эффектов атаки

private DispatcherTimer attackTimer; // Управляет прогрессией вирусной атаки

private DispatcherTimer dangerTimer; // Отсчет времени до критической стадии заражения

private DispatcherTimer defeatTimer; // Таймер до полного заражения системы

private DispatcherTimer glitchTimer; // Таймер, вызывающий визуальные "глитчи"

// Переменные состояния атаки

private int attackStage; // Этап атаки

private int defeatCounter; // Количество секунд до поражения системы

private string activeAttackType; // Текущий тип вирусной атаки

private bool defenseApplied = false; // Флаг, указывающий, была ли применена защита

private Random random = new Random(); // Генератор случайных значений

// Конструктор окна PlayWindow, инициализирует таймеры и логику атаки

public PlayWindow()

{

InitializeComponent();

// Настройка таймеров

attackTimer = new DispatcherTimer { Interval = TimeSpan.FromSeconds(2) };

attackTimer.Tick += AttackEffect; // Обновляет эффект атаки

dangerTimer = new DispatcherTimer { Interval = TimeSpan.FromSeconds(10) };

dangerTimer.Tick += DefeatCountdown; // Обновляет отсчет до критического момента

defeatTimer = new DispatcherTimer { Interval = TimeSpan.FromSeconds(1) };

defeatTimer.Tick += DefeatCountdown; // Обновляет таймер до полного заражения

glitchTimer = new DispatcherTimer { Interval = TimeSpan.FromMilliseconds(300) };

glitchTimer.Tick += GlitchEffect; // Вызывает визуальные глитчи

}

// Метод запуска вирусной атаки

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

{

attackStage = 0;

defeatCounter = 20; // Устанавливаем время до поражения

defenseApplied = false; // Сбрасываем защиту

activeAttackType = SelectRandomAttack(); // Выбираем случайный тип вируса

// Запускаем таймеры атаки

attackTimer.Start();

dangerTimer.Start();

defeatTimer.Start();

// Выводим предупреждение в логах и меняем фон

AttackLog.Text = $"⚠ Вирус запущен! Тип атаки: {activeAttackType}\n";

Background = Brushes.DarkRed;

// Обновляем UI таймера

UpdateTimerUI();

}

// Метод случайного выбора типа атаки

private string SelectRandomAttack()

{

string[] attacks = { "DDoS-атака", "Шифратор", "Кейлоггер", "Троян", "Руткит", "Червь", "Эксплойт", "Фишинг", "Рекламный вирус", "Ботнет" };

return attacks[random.Next(attacks.Length)]; // Выбираем случайную атаку

}

// Эффект усиления вирусной атаки

private void AttackEffect(object sender, EventArgs e)

{

attackStage++; // Увеличиваем уровень атаки

AttackLog.Text += $"⚠ Этап {attackStage}: {activeAttackType} усиливается...\n";

// Если атака достигла 5-го этапа, вирус мутирует

if (attackStage >= 5)

{

AttackLog.Text += "🔥 Вирус мутировал, заражение усиливается!\n";

}

}

// Метод обработки поражения системы

private void DefeatCountdown(object sender, EventArgs e)

{

if (!defenseApplied)

{

defeatCounter--;

UpdateTimerUI();

// Если осталось 5 секунд — включаем визуальные глитчи

if (defeatCounter <= 5) glitchTimer.Start();

if (defeatCounter <= 0)

{

defeatTimer.Stop();

attackTimer.Stop();

dangerTimer.Stop();

glitchTimer.Stop();

AttackLog.Text += "💀 Система полностью заражена! Требуется перезагрузка...\n";

Background = Brushes.Black;

}

}

}

// Обновление UI таймера

private void UpdateTimerUI()

{

TimerText.Text = $"⏳ Осталось: {defeatCounter} сек";

AttackProgressBar.Value = defeatCounter;

// Меняем цвет текста в зависимости от оставшегося времени

if (defeatCounter <= 10)

TimerText.Foreground = Brushes.Orange;

if (defeatCounter <= 5)

TimerText.Foreground = Brushes.Red;

}

// Визуальные глитчи при критическом уровне заражения

private void GlitchEffect(object sender, EventArgs e)

{

Color[] glitchColors = { Colors.Red, Colors.Magenta, Colors.DarkBlue, Colors.Black };

Background = new SolidColorBrush(glitchColors[random.Next(glitchColors.Length)]);

string[] glitchText = { "###", "!!!", "&&&", "@@@" };

AttackLog.Text += glitchText[random.Next(glitchText.Length)] + " Глитч в системе! " + glitchText[random.Next(glitchText.Length)] + "\n";

}

// Применение защиты против атаки

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

{

if (DefenseOptions.SelectedItem is ComboBoxItem selectedDefense)

{

AttackLog.Text += $"🛡 Выбрана защита: {selectedDefense.Content}\n";

if (IsDefenseCorrect(selectedDefense.Content.ToString()))

{

// Если защита правильная — атака останавливается

defenseApplied = true;

defeatTimer.Stop();

dangerTimer.Stop();

glitchTimer.Stop();

AttackLog.Text += "✅ Атака успешно нейтрализована!\n";

Background = Brushes.Green;

attackTimer.Stop();

}

else

{

// Если защита неверная — атака мутирует

AttackLog.Text += "⚠ Выбранная защита неэффективна! Вирус продолжает атаку...\n";

Background = Brushes.DarkOrange;

attackStage++;

AttackLog.Text += "🔥 Вирус мутировал! Защита теперь сложнее!\n";

if (defeatCounter <= 5) glitchTimer.Start();

}

}

else

{

AttackLog.Text += "⚠ Выберите метод защиты!\n";

}

}

// Проверка корректности выбранной защиты

private bool IsDefenseCorrect(string defenseType)

{

return activeAttackType.Equals(defenseType.Trim(), StringComparison.OrdinalIgnoreCase);

}// Простая проверка соответствия

// Метод возврата к главному окну

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

{

MainWindow mainWindow = new MainWindow();

mainWindow.Show();

this.Close();

}

}

}

PageWindow.xaml(страница для отображения веб-страниц):

<Window x:Class="virus1.PageWindow" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:virus1" xmlns:wpf="clr-namespace:Microsoft.Web.WebView2.Wpf;assembly=Microsoft.Web.WebView2.Wpf"

mc:Ignorable="d" WindowStartupLocation="CenterScreen"

WindowStyle="None"

Title="PageWindow" Height="617" Width="876" ResizeMode="NoResize">

<Grid Background="DarkRed">

<wpf:WebView2 Name="webView" Width="800" Height="500" Margin="38,10,38,107"/>

<Button Background="DarkRed" Click="Nazad\_Click" Width="72" Height="66"

HorizontalAlignment="Center" VerticalAlignment="Top"

Foreground="#FFE02222" Margin="0,524,0,0"

Panel.ZIndex="100" RenderTransformOrigin="0.51,0.88">

<Image Source="image1/18.png" Stretch="Fill" Height="62" Width="66" />

</Button>

</Grid>

</Window>

PageWindow.xaml.cs(страница для отображения веб-страниц):

using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Policy;

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 virus1

{

/// <summary>

/// Логика взаимодействия для PageWindow.xaml

/// </summary>

public partial class PageWindow : Window

{

public PageWindow(string url)

{

InitializeComponent();

webView.Source = new Uri(url);

}

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

{

this.Close();

}

}

}

SplashScreen.xaml (загрузка):

<Window x:Class="virus1.SplashScreen" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:virus1"

mc:Ignorable="d"

ResizeMode="NoResize"

WindowStyle="None" WindowStartupLocation="CenterScreen"

Title="SplashScreen" Height="613" Width="800">

<Window.Background>

<ImageBrush ImageSource="/image1/fonscreen.jpg" Opacity="0.8"/>

</Window.Background>

<Grid>

<TextBlock Text="Загрузка..." FontSize="35" HorizontalAlignment="Left" FontFamily="JetBrains Mono ExtraBold" Foreground="#FFDCA3A3" Margin="302,10,0,0"/>

<ProgressBar Name="LoadingBar" Width="300" Height="20" Margin="275,440,225,153" RenderTransformOrigin="0.5,0.5">

<ProgressBar.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform Angle="-3.42"/>

<TranslateTransform/>

</TransformGroup>

</ProgressBar.RenderTransform>

</ProgressBar>

</Grid>

</Window>

SplashScreen.xaml.cs (загрузка):

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 virus1

{

public partial class SplashScreen : Window

{

public SplashScreen()

{

InitializeComponent();

StartLoading();

}

private async void StartLoading()

{

for (int i = 0; i <= 100; i += 5)

{

LoadingBar.Value = i;

await Task.Delay(100); // Задержка для эффекта загрузки

}

OpenMainWindow();

}

private void OpenMainWindow()

{

MainWindow mainWindow = new MainWindow();

mainWindow.Show();

this.Close(); // Закрываем заставку

}

}

}

App.xaml:

<Application x:Class="virus1.App" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="clr-namespace:virus1"

>

<Application.Resources>

</Application.Resources>

</Application>

App.xaml.cs:

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Threading.Tasks;

using System.Windows;

namespace virus1

{

/// <summary>

/// Логика взаимодействия для App.xaml

/// </summary>

public partial class App : Application

{

protected override void OnStartup(StartupEventArgs e)

{

base.OnStartup(e);

SplashScreen splashScreen = new SplashScreen();

splashScreen.Show();

}

}

}