

Шакиров Р.Л. гр.4204 дата 05.12.2025

Лабораторная работа №. 9

Создание меню

С#

Задание 3: Скомпилируйте и выполните приложение по примеру 3. Проанализируйте работу приложения. Определите для пунктов меню клавиши быстрого доступа. Комбинации клавиш быстрого доступа должна отображаться рядом с названием пункта меню.

Данная лабораторная работа также была выполнена с применением технологии контейнеризации в силу различной файловой системы Unix/Windows далее будет приложен dockerfile и readme файл

dockerfile

FROM ubuntu:20.04

Установка переменных окружения для избежания

интерактивных диалогов

ENV DEBIAN_FRONTEND=noninteractive

Установка Mono и зависимостей

RUN apt-get update && apt-get install -y

gnupg

ca-certificates

wget

&& wget -qO- https://download.mono-project.com/repo/xamarin.gpg | apt-key add -

&& echo "deb https://download.mono-project.com/repo/ubuntu stable-focal main" | tee

/etc/apt/sources.list.d/mono-official-stable.list

&& apt-get update && apt-get install -y

mono-devel

mono-runtime

libgdiplus

libx11-dev

&& rm -rf /var/lib/apt/lists/*

WORKDIR /app

COPY Form1.cs .

Компиляция приложения

RUN mcs -r:System.Windows.Forms.dll -r:System.Drawing.dll -r:System.dll -out:Form1.exe Form1.cs

CMD ["mono", "Form1.exe"]

Form1.cs

using System;

using System.Drawing;

using System.Windows.Forms;

namespace Lab9_Task3

{

public class Form1 : Form

```

{
    private int width = 50;
    private int height = 50;
    private Color myColor = Color.Red;

    public Form1()
    {
        InitializeComponent();
    }

    private void InitializeComponent()
    {
        // Настройка формы
        this.Text = "Лабораторная работа №3 - Меню с горячими клавишами";
        this.BackColor = Color.White;
        this.Size = new Size(500, 400);
        this.StartPosition = FormStartPosition.CenterScreen;

        CreateMainMenu();
        CreateContextMenu();
    }

    private void CreateMainMenu()
    {
        // Создание главного меню
        MainMenu mainMenu = new MainMenu();

        // Пункт меню "Размер" с подменю
        MenuItem sizeMenu = new MenuItem("&Размер");

        // Создание пунктов подменю с горячими клавишами
        MenuItem bigItem = new MenuItem("&Большой", (s, e) => SetSize(150, 150,
        "Большой"));
        MenuItem mediumItem = new MenuItem("&Средний", (s, e) => SetSize(100, 100,
        "Средний"));
        MenuItem smallItem = new MenuItem("&Маленький", (s, e) => SetSize(50, 50,
        "Маленький"));

        // Установка горячих клавиш
        bigItem.Shortcut = Shortcut.CtrlB;
        mediumItem.Shortcut = Shortcut.CtrlM;
        smallItem.Shortcut = Shortcut.CtrlS;

        // Включение отображения горячих клавиш
        bigItem.ShowShortcut = true;
        mediumItem.ShowShortcut = true;
        smallItem.ShowShortcut = true;

        // Добавление пунктов в меню
        sizeMenu.MenuItems.Add(bigItem);
        sizeMenu.MenuItems.Add(mediumItem);
        sizeMenu.MenuItems.Add(smallItem);

        // Добавление разделителя
    }
}

```

```

sizeMenu.MenuItems.Add(new MenuItem("-"));

// П у н к т в ы х о д а
MenuItem exitItem = new MenuItem("&Выйти", (s, e) => Application.Exit());
exitItem.Shortcut = Shortcut.CtrlQ;
exitItem.ShowShortcut = true;
sizeMenu.MenuItems.Add(exitItem);

mainMenu.MenuItems.Add(sizeMenu);
this.Menu = mainMenu;
}

private void CreateContextMenu()
{
    // С о з д а н и е к о н т е к с т н о г о м е н ю
    ContextMenu contextMenu = new ContextMenu();

    MenuItem redItem = new MenuItem("&Красный", (s, e) => SetColor(Color.Red));
    MenuItem blueItem = new MenuItem("&Синий", (s, e) => SetColor(Color.Blue));
    MenuItem greenItem = new MenuItem("&Зеленый", (s, e) =>
SetColor(Color.Green));

    // У с т а н о в к а г о р я ч и х к л а в и ш
    redItem.Shortcut = Shortcut.CtrlR;
    blueItem.Shortcut = Shortcut.CtrlL;
    greenItem.Shortcut = Shortcut.CtrlG;

    // В к л ю ч е н и е о т о б р а ж е н и я г о р я ч и х к л а в и ш
    redItem.ShowShortcut = true;
    blueItem.ShowShortcut = true;
    greenItem.ShowShortcut = true;

    contextMenu.MenuItems.Add(redItem);
    contextMenu.MenuItems.Add(blueItem);
    contextMenu.MenuItems.Add(greenItem);

    this.ContextMenu = contextMenu;

    // О б р а б о т ч и к п р а в о й к н о п к и м ѿш и
    this.MouseDown += (s, e) =>
    {
        if (e.Button == MouseButtons.Right)
        {
            contextMenu.Show(this, new Point(e.X, e.Y));
        }
    };
}

private void SetSize(int w, int h, string sizeName)
{
    width = w;
    height = h;
    this.Text = $"Лабораторная работа №3 - {sizeName}"
п р я м о у г о л ь н и к";
    Invalidate();
}

```

```

private void SetColor(Color color)
{
    myColor = color;
    Invalidate();
}

protected override void OnPaint(PaintEventArgs e)
{
    base.OnPaint(e);
    Graphics dc = e.Graphics;

    // Очистка фона
    dc.Clear(this.BackColor);

    // Рисование прямоугольника
    using (Pen pen = new Pen(myColor, 3))
    {
        dc.DrawRectangle(pen, 50, 50, width, height);
    }

    // Отображение информации
    using (Font font = new Font("Arial", 10))
    using (Brush brush = new SolidBrush(Color.Black))
    {
        string info = $"Размер: {width}x{height} | Цвет: {myColor.Name}";
        dc.DrawString(info, font, brush, 50, 20);

        string help = "Горячие клавиши:\n" +
                      "Размер: Ctrl+B, Ctrl+M, Ctrl+S\n" +
                      "Цвет: Ctrl+R, Ctrl+L, Ctrl+G\n" +
                      "Выход: Ctrl+Q\n" +
                      "Правый клик - контекстное меню";
        dc.DrawString(help, font, brush, 50, height + 70);
    }
}

[STAThread]
static void Main()
{
    Application.Run(new Form1());
}
}

```

README.md

Запуск приложения

1. Разрешите доступ к X11 серверу
bash
xhost +local:docker
2. Соберите Docker образ
bash
docker build -t mono-winforms-app .

3. Запустите контейнер

```
bash  
docker run -it --rm $  
-e DISPLAY=$DISPLAY  
-v /tmp/.X11-unix:/tmp/.X11-unix  
mono-winforms-app
```

Вывод: рассмотрено создание меню в C#