МОЛДАВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ

ДЕПАРТАМЕНТ ИНФОРМАТИКИ

# Проект по предмету:

# Технологии программирования

# *«newSearcher»*

Группа: MIA2302

Преподаватель:\_\_\_\_\_\_\_\_\_ Арнаут Всеволод

Студенты: \_\_\_\_\_\_\_\_Павленко Антон

\_\_\_\_\_\_\_\_\_\_Нартя Никита

\_\_\_\_\_\_\_\_\_\_Малай Михаил

\_\_\_\_\_\_\_\_\_\_Сыч Домашний

КИШИНЁВ, 2023

## Содержание:

## Введение и постановка задачи ------------------------------------------------------------ 3

## Язык программирования C# ------------------------------------------------------------ 4

## User Interface ------------------------------------------------------------ 5

## Зона для Никиты ------------------------------------------------------------ 3

## Зона для Миши ------------------------------------------------------------ 3

## Зона для Сыча ------------------------------------------------------------ 3

## Заключение ------------------------------------------------------------ 3

Введение и постановка задачи

Современный мир информации требует удобных и эффективных способов получения и обработки контента из различных источников. В этом контексте приложение, разработанное на языке программирования C# для парсинга статей с веб-сайтов, таких как Tproger и Habr, представляет собой ценный инструмент для улучшения опыта пользователя.

Преимущества Приложения

1. **Удобство Использования**

Пользовательский интерфейс приложения обеспечивает простоту и удобство в работе. Пользователи могут легко находить и читать статьи без лишних усилий.

1. **Сокращение Затрат Времени**

Приложение позволяет пользователям сэкономить время на поиске и отборе интересующего контента, предоставляя собранные статьи в удобном формате.

**Практическое Применение**

1. **Исследования и Образование**

Приложение может быть полезным инструментом для исследователей и студентов, позволяя эффективно собирать информацию для образовательных целей.

1. **Профессиональное Развитие**

Для профессионалов в области информационных технологий приложение предоставляет возможность отслеживать последние тенденции и новости в отрасли.

Далее мы рассмотрим какие технологии были применены для написания данного приложения и иллюстрации его работы.

Язык Программирования C#

C# — современный и многофункциональный язык программирования, разработанный компанией Microsoft. Он является частью технологии Microsoft .NET и предназначен для создания разнообразных приложений, начиная от десктопных и веб-приложений до мобильных приложений и облачных сервисов.

**Основные Характеристики**

1. **Объектно-Ориентированный Язык**

C# поддерживает принципы объектно-ориентированного программирования (ООП), что упрощает создание модульного и структурированного кода.

1. **Интеграция с Платформой .NET**

Язык тесно связан с платформой .NET, предоставляя разработчикам мощные библиотеки классов, которые упрощают разработку и обеспечивают переносимость кода.

1. **Современный Синтаксис**

C# предлагает чистый и современный синтаксис, включая такие возможности, как LINQ (Language Integrated Query) для удобной работы с данными и асинхронное программирование.

1. **Безопасность Типов**

Система строгой типизации и управление памятью делают код на C# более безопасным и предотвращают многие ошибки, связанные с работой с памятью.

1. **Поддержка Параллельного Программирования**

C# предоставляет средства для эффективного использования многозадачности и параллелизма через асинхронные операции и библиотеки для параллельного программирования.

**Области Применения**

1. **Разработка Приложений Windows**

C# широко используется для создания десктопных приложений под управлением операционных систем Windows.

1. **Веб-Разработка**

С использованием ASP.NET, C# применяется для создания веб-приложений и веб-сайтов.

1. **Мобильная Разработка**

Xamarin, платформа для разработки мобильных приложений, позволяет использовать C# для создания кросс-платформенных приложений для iOS и Android.

1. **Серверные Приложения и Облачные Сервисы**

C# эффективно используется для создания серверных приложений и облачных сервисов, благодаря своей интеграции с технологией .NET.

User Interface

Для разработки приложения был выбран framework языка C# Forms. System.Windows.Forms (WinForms) представляет собой часть .NET Framework и является библиотекой для создания графических пользовательских интерфейсов (GUI) в Windows-приложениях с использованием языка программирования C#. Этот фреймворк обеспечивает разработчиков мощными инструментами для создания интерактивных и интуитивно понятных приложений с использованием стандартных элементов управления.

**Основные Компоненты:**

**1. Формы (Forms)**

Формы представляют собой оконные интерфейсы приложений. Они служат контейнерами для размещения других элементов управления, таких как кнопки, текстовые поля и списки.

**2. Элементы Управления (Controls)**

WinForms предоставляет богатый набор стандартных элементов управления, таких как TextBox, Button, ComboBox и многие другие. Эти элементы обеспечивают базовую функциональность для взаимодействия с пользователем.

**3. Меню и Панели Инструментов**

Фреймворк позволяет легко создавать меню и панели инструментов, что облегчает навигацию и управление приложением.

**4. Графическое Рисование**

System.Windows.Forms поддерживает возможность графического рисования, что полезно для создания пользовательских интерфейсов и визуальных эффектов.

**5. Обработка Событий**

Механизм обработки событий позволяет реагировать на действия пользователя или изменения в приложении, обеспечивая отзывчивость интерфейса.

**Преимущества Использования:**

1. **Простота Использования**

WinForms предоставляет простой и интуитивно понятный способ создания пользовательских интерфейсов, что делает его идеальным выбором для разработчиков с разным уровнем опыта.

1. **Быстрое Развертывание**

Windows-приложения, построенные с использованием WinForms, легко разворачиваются на машинах конечных пользователей, так как .NET Framework включен в большинство версий Windows.

1. **Интеграция с Windows**

Фреймворк интегрирован с функциональностью Windows, что позволяет создавать приложения с естественным пользовательским опытом.

1. **Богатые Возможности Элементов Управления**

WinForms предоставляет обширный выбор стандартных элементов управления и позволяет легко создавать пользовательские элементы.

**Пример Кода:**

using System;

using System.Windows.Forms;

public class MainForm : Form

{

public MainForm()

{

// Инициализация компонентов формы

InitializeComponent();

}

private void InitializeComponent()

{

// Код для инициализации элементов управления и других компонентов

}

[STAThread]

static void Main()

{

// Запуск главной формы

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new MainForm());

}

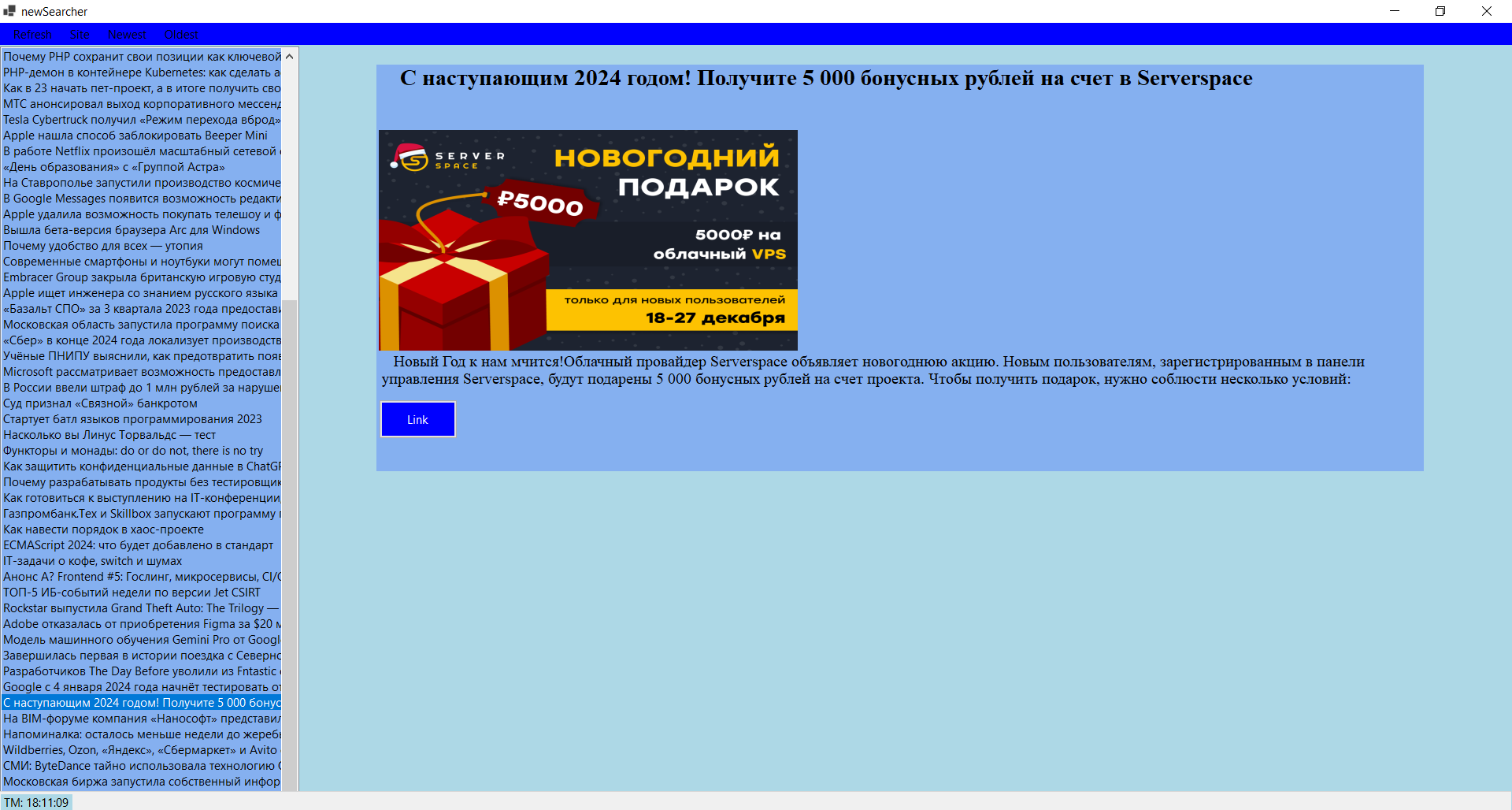
}

**Использованные классы:**

* Form
* ListBox
* Panel
* StatusStrip
* ToolStripStatusLabel
* Timer
* FlowLayoutPanel
* ToolStripMenuItem
* UserControl

**Иллюстрации работы приложения:**





Впрочем, как и у любого инструмента. Работа с Forms происходит не без трудностей. Одна из основных сложностей есть управление расположением и размерами элементов управления при изменении размеров окна. Это требует более тщательного подхода к размещению элементов для обеспечения корректного отображения на разных разрешениях экрана. Также возникает некоторые проблемы при кастомизации стандартных элементов управления для соответствия дизайну приложения. Однако, благодаря обширной документации и активному сообществу, преодолеть эти трудности и создать функциональный и эстетичный пользовательский интерфейс возможно.

Заключение:

Разработка приложения, включающего в себя веб-скрапинг, управление базой данных и создание пользовательского интерфейса с использованием фреймворка System.Windows.Forms, представляет собой многогранный и увлекательный процесс. Взаимодействие с различными компонентами, такими как парсер сайта, база данных и API, требует системного подхода и глубокого понимания каждого элемента системы.

Интеграция парсера сайта предполагает анализ веб-страниц и эффективное извлечение информации. Создание и управление базой данных требует оптимального проектирования схемы данных, выбора подходящей СУБД и реализации механизмов взаимодействия. Разработка API для взаимодействия с базой данных предоставляет удобные методы доступа и обновления данных.

Создание пользовательского интерфейса с System.Windows.Forms вносит свои вызовы, включая дизайн и взаимодействие с пользователем. Однако, учитывая аспекты парсинга, работы с базой данных и API, разработчик формирует приложение, способное не только предоставлять удобный пользовательский интерфейс, но и взаимодействовать с веб-ресурсами и эффективно управлять данными через API.

Эти сложности в разработке приложения сопровождаются уникальными возможностями. Преодоление этих вызовов приносит ценный опыт, позволяет создать комплексное приложение, способное взаимодействовать с веб-ресурсами, хранить данные в базе данных, предоставлять API для доступа к информации, а также обеспечивать приятный и функциональный пользовательский опыт. Этот процесс объединяет различные навыки и технологии в единое решение, готовое удовлетворить потребности пользователей в эффективном и удобном управлении данными.