УДК 004

Создание программы-скриншотер на языке программирования C#

Эрдман Александр Алексеевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В статье рассмотрен процесс создания программы-скриншотера, способной делать снимки экрана компьютера. Программа написана на языке программирования C# на базе библиотеки WinForms, которая позволяет создавать настольные приложения. Результатом исследования будет являться оконное приложения с возможностями создания снимка экрана компьютера.
Ключевые слова: C#, WinForms, скриншотер

Creating a screenshot program in the C programming language

Erdman Alexander Alekseevich
Sholom-Aleichem Priamursky State University
Student

Abstract
The article describes the process of creating a screenshot program capable of taking screenshots of a computer screen. The program is written in the C# programming language based on the WinForms library, which allows you to create desktop applications. The result of the study will be a windowed application with the ability to create a screenshot of a computer screen.
Keywords: C#, WinForms, screenshot

1 Введение

1.1 Актуальность

В сегодняшнюю цифровую эпоху возможность делать скриншоты и делиться ими стала неотъемлемой частью повседневной жизни. Будь то для работы, личного использования или просто для того, чтобы поделиться забавными моментами с друзьями, потребность в надежной и эффективной программе захвата экрана для ПК никогда не была так высока. Снимки экрана позволяют нам фиксировать важную информацию, документировать визуальный контент или просто сохранять памятные моменты нашего цифрового опыта. Таким образом, разработка программы для создания снимков экрана для ПК не только актуальна, но и необходима для удовлетворения растущего спроса на такой инструмент. Также в сфере IT популяризируется язык программирования C#, на котором удобно создавать оконные приложения, в том числе и приложение-скриншотер.

1.2 Обзор исследований

С. М. Баженов в своей работе представил преимущества языка программирования C# в разработке приложений [1]. А.С. Ерохин в своей статье рассмотрел процесс создания графической части приложения и написания кода к различным элементам приложения [2]. Т.В. Ромашкина и В.И. Шагалиев рассмотрели основополагающие возможностей языка C# для создания программ [3]. Д.И. Максимов рассмотрел принципы разработки модульных приложений на основе Managed Extensibility Framework, а также основные возможности фреймворка [4]. В.М. Куприенко, Н.В. Нечитайло в своей статье описали процесс разработки приложений для Windows на языке C# [5].

1.3 Исследования

Целью исследования является создание приложения-скриншотера, с помощью которого можно делать снимки экрана компьютера, при помощи языка программирования C# и библиотеки WinForms.

2 Материалы и методы

Для создания программы используется язык программирования C# и библиотека WinForms. В качестве IDE используется Visual Studio 2022.

3 Результаты и обсуждения

Создание приложение начинается с реализации графического интерфейса. Программа имеет два окна. Первое окно является начальным при запуске, то есть главным. На данном окне размещаются два элемента типа «Button» с названиями «Сделать скриншот» и «Выход» (рис. 1).

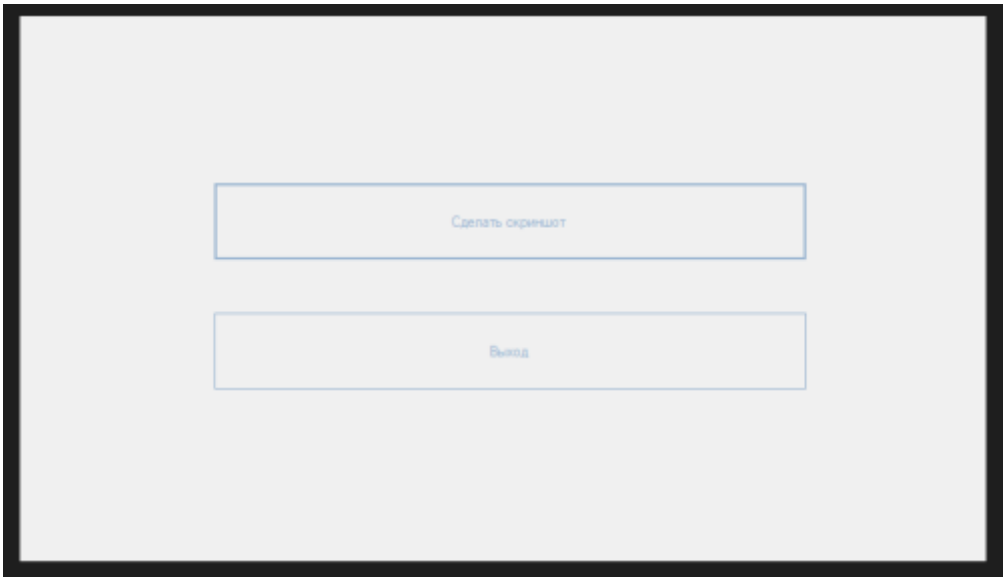

Рисунок 1. Внешний вид главного окна приложения

Второе окно имеет функции предпросмотра скриншота, возможность сохранить его в файл и закрыть скриншот (рис. 2). В данном окне

располагаются следующие элементы формы: две кнопки «Button», «pictureBox», два элемента «panel» и элемент «saveFileDialog». Элементы «Button» будут выполнять функции закрытия предпросмотра скриншота и вызова элемента «saveFileDialog», с помощью которого можно будет выбрать место сохранение скриншота. В свойствах элемента «saveFileDialog» указывается формат сохраняемого скриншота в параметре «Behavior-Filter» (рис. 3). Формат сохраняемого изображения был выбран «PNG».

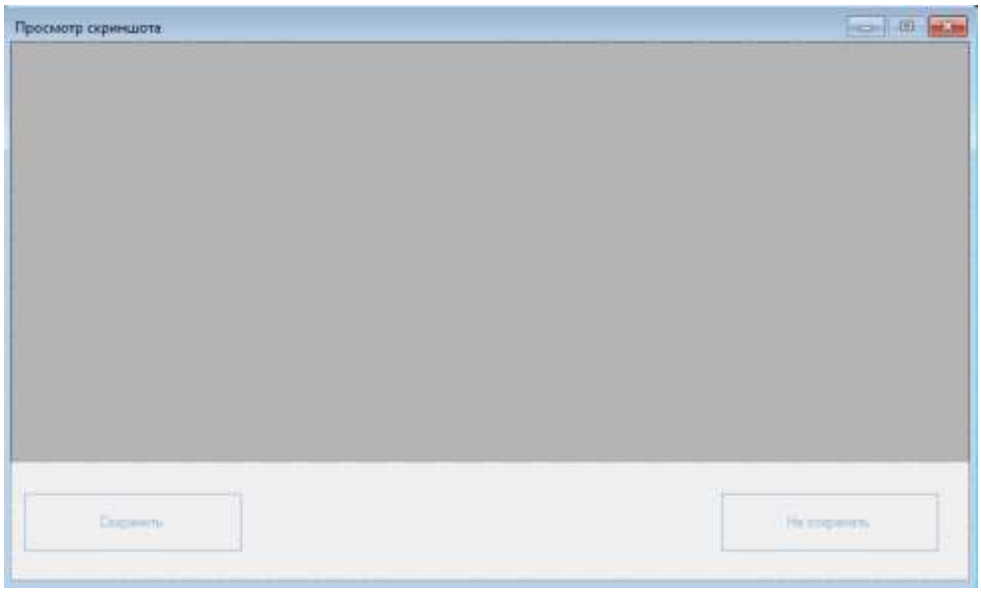

Рисунок 2. Внешний вид окна просмотра скриншота

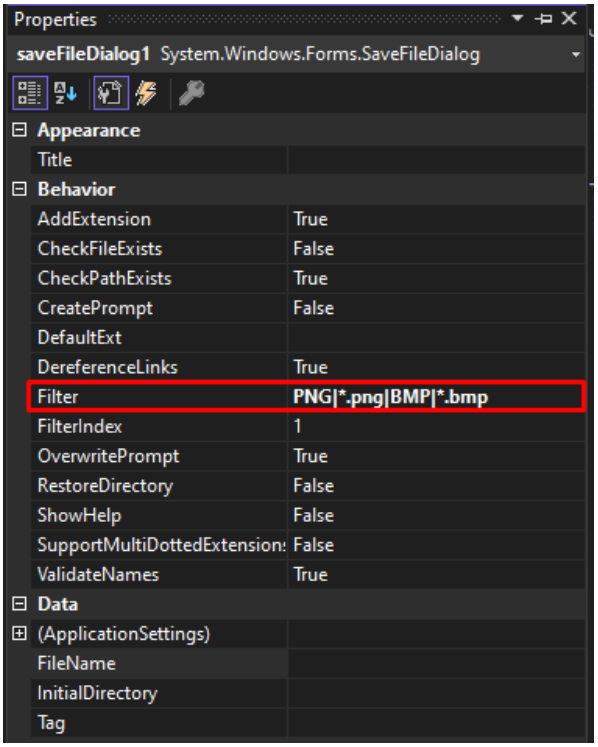

Рисунок 3. Установка формата изображения сохраняемого снимка

Элемент «pictureBox» позволяет выводить и размешать изображение. В данном элементе будет выводиться непосредственно снимок экрана

(скриншот). Элементы «panel» используются для более удобного и привального размещения объектов по ширине и высоте.

После создания графического интерфейса осуществляется программирования функций в главной форме (рис. 4).

```
public partial class Form1 : Form
{
    public static Bitmap img = new Bitmap(Screen.PrimaryScreen.Bounds.Width,
        Screen.PrimaryScreen.Bounds.Height);
    1 reference
    public Form1()
    {
        InitializeComponent();
    }

    1 reference
    private void button1_Click(object sender, EventArgs e)
    {
        this.Opacity = .0;
        Graphics gr = Graphics.FromImage(img as Image);
        gr.CopyFromScreen(0, 0, 0, 0, img.Size);
        Screenshootcs scr = new Screenshootcs();
        scr.ShowDialog();
        this.Opacity = 1;
    }

    1 reference
    private void button2_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }
}
```

Рисунок 4. Функционал главной формы приложения

В первом обработчике события нажатия на кнопку создаётся новый экземпляр класса «Bitmap» с именем «img». Класс «Bitmap» используется для представления изображений, и в данном случае он используется для создания пустого изображения с такими же размерами, как ширина и высота основного экрана. Свойства «Screen.PrimaryScreen.Bounds.Width» и «Screen.PrimaryScreen.Bounds.Height» возвращают ширину и высоту основного экрана в пикселях. В этой же форме создаются два обработчика событий нажатия на кнопки «Сделать скриншот» и «Выход». Обработчик нажатия на первую кнопку приводит главное окно в непрозрачное состояние с помощью метода «this.Opacity = .0» для того, чтобы скрыть визуально окно так, как дальше будет осуществлён переход на окно просмотра скриншота. Далее создаётся новый объект «Graphics gr» из предоставленного изображения «img» с использованием метода «Graphics.FromImage». Ключевое слово «as Image» используется для приведения переменной «img» к типу «Image». С помощью метода «CopyFromScreen» графического объекта «gr» осуществляется захват снимка экрана с исходными координатами. С помощью команды «scr.ShowDialog()» форма «scr» отображается как модальное диалоговое, и блокируются взаимодействия пользователя с другими окнами, пока оно не будет закрыто. С помощью команды «this.Opacity = 1» главное окно становится вновь непрозрачным для дальнейшего пользования.

Во втором обработчике события нажатия на кнопку прописывается элементарная команда закрытия приложения «Application.Exit()».

Второе окно программы содержит функции получения снимка экрана и два обработчика нажатия на кнопку, которые выполняют закрытие окна и вызова окна с возможностью выбора места сохранения скриншота (рис. 5).

```
public partial class Screenshootcs : Form
{
    1 reference
    public Screenshootcs()
    {
        InitializeComponent();
        pictureBox1.Image = Form1.img;
    }

    1 reference
    private void button1_Click(object sender, EventArgs e)
    {
        if (saveFileDialog1.ShowDialog() == DialogResult.OK)
        {
            Form1.img.Save(saveFileDialog1.FileName);
        }
    }

    1 reference
    private void button2_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}
```

Рисунок 5. Функционал окна предпросмотра скриншшота

В конструкторе класса происходит инициализация компонентов формы при помощи метода «InitializeComponent()». Далее, установленное изображение на «pictureBox1» принимает изображение «img», которое находится в классе «Form1». В методе «button1_Click» обрабатывается событие нажатия на кнопку «Сохранить». Если пользователь выбирает файл для сохранения при помощи «saveFileDialog1» и нажимает кнопку «ОК», то изображение «img» сохраняется в выбранный файл. В методе «button2_Click» обрабатывается событие нажатия на кнопку «Не сохранять». При нажатии на эту кнопку происходит закрытие текущей формы.

После реализации программной части совершается проверка приложения на практике (рис. 6, 7).

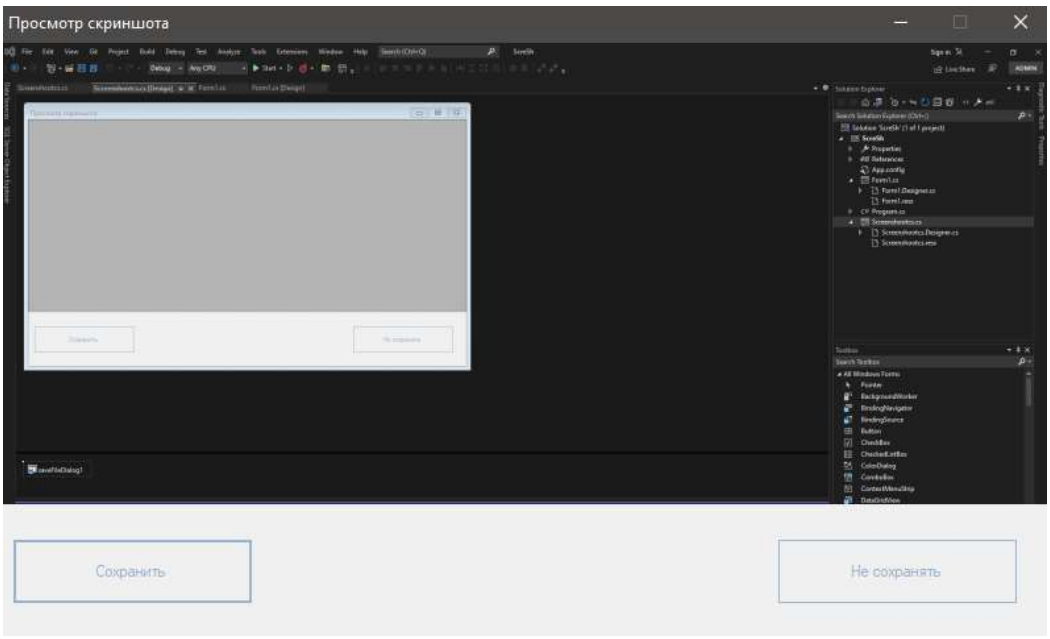

Рисунок 6. Работа приложения-скриншотер, демонстрация окна предпросмотра снимака

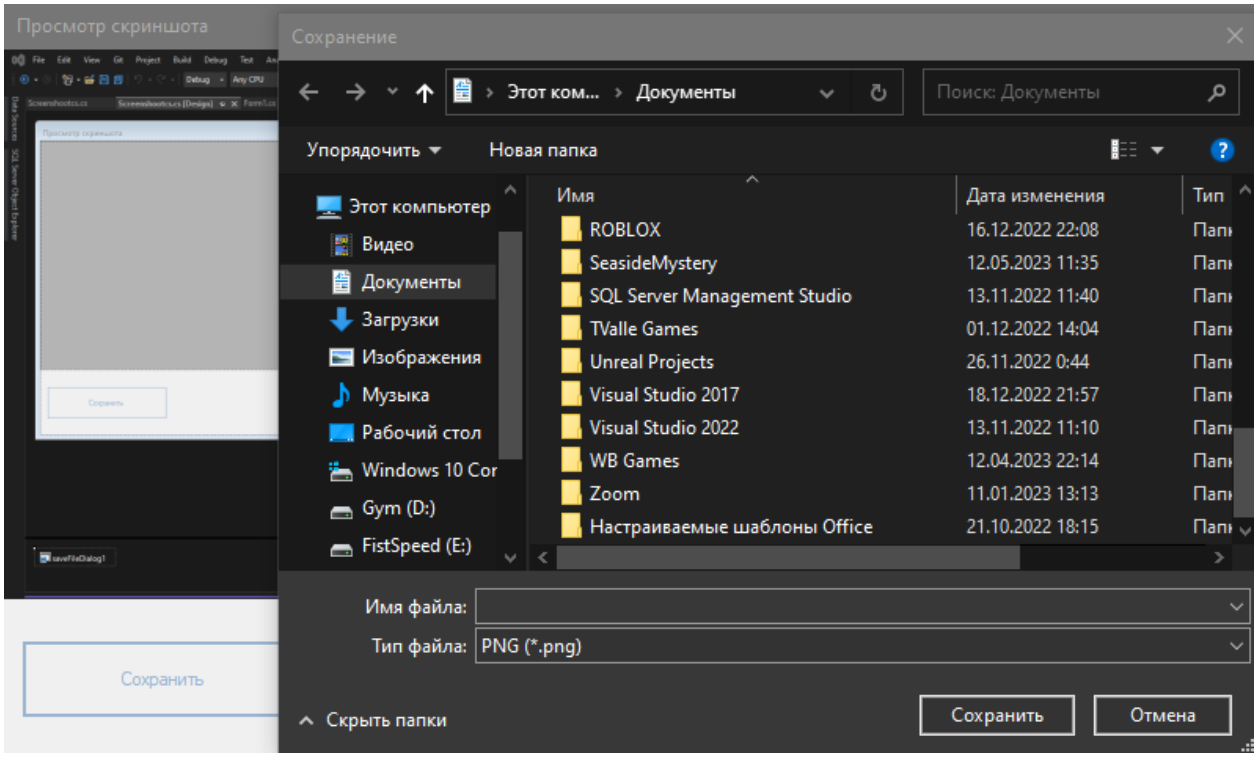

Рисунок 7. Работа приложения-скриншотер, демонстарция функции сохранения снимка

В результате исследования было создано оконное приложение-скриншотер, позволяющее делать снимок экрана компьютера, с помощью языка программирования C# и библиотеки WinForms. Также был описан процесс разработки программы и принцип её работы.

Библиографический список

1. Баженов С. М. Преимущества языковой платформы C# (си шарп) в разработке многозадачных приложений для контроля качества // Редакционная коллегия. – С. 118.
2. Ерохин А.С. Создание оконного приложения на языке C# // В сборнике: Интеллектуальный потенциал XXI века инновационной России. Материалы VIII Всероссийской научно-практической конференции обучающихся и студентов, посвященной 100-летию ОГУ. 2019. С. 29-32.
3. Ромашкина Т.В., Шагалиев В.И. Создание приложений для Windows средствами языка C# // Хроники объединенного фонда электронных ресурсов Наука и образование. 2015. № 12 (79). С. 115.
4. Максимов Д.И. Разработка модульных приложений на C# // В сборнике: Системы управления, технические системы: устойчивость, стабилизация, пути и методы исследования. Материалы молодежной секции в рамках IV Международной научно-практической конференции. 2018. С. 205-211.
5. Куприенко В.М., Нечитайло Н.В. Разработка Windows-приложений на языке C# // В сборнике: Современные инновации в науке и технике. Сборник научных трудов 4-ой Международной научно-практической конференции: В 4-х томах. Ответственный редактор Горохов А.А., 2014. С. 340-343.