

## Создание приложения для рисования с использованием языка программирования C#

*Ульянов Егор Андреевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### Аннотация

Целью данной статьи является применение языка программирования C# для создания в среде разработки Visual Studio 2019 программы для рисования. Практическим результатом является разработанное приложение.

**Ключевые слова:** C#, Visual Studio, программа для рисования

### Creating a drawing app using C # programming language

*Ulianov Egor Andreevich*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

The purpose of this article is to use the C # programming language to create a drawing program in the Visual Studio 2019 development environment. The practical result is a developed application.

**Keywords:** C #, Visual Studio, drawing program

## 1 Введение

### 1.1 Актуальность исследования

Одним из самых популярных видов досуга многих людей на сегодняшний момент является рисование. Рисование - оживление фантазий на листе, которое приносит художнику огромное удовлетворение: успокаивает, развивает фантазию, способствуют развитию интеллекта, как умственного, так и эмоционального, развивает внимание, реакцию, пространственную ориентацию и логическое мышление. Сейчас многие обыденные занятия проходят процесс цифровизации, как и рисование. Популярные программы для рисования: Krita, SmoothDraw, Gimp, Artweaver, Paint.NET, PixBuilder Studio, Inskape, Adobe Photoshop. Разработкой программ может заниматься как один человек, так и команда. Создание программы - это продолжительный и трудоёмкий процесс, состоящий из самых разнообразных этапов, включающий в себя как технические, так и творческие моменты.

## **1.2 Обзор исследований**

В своей работе Н. Н. Додобоев, О. И. Кукарцева, Я. А. Тынченко рассмотрели вопросы появления различных языков программирования (в частности C#), определения особенностей этих языков, а также составления основных видов и классификаций языков программирования [1]. З. С. Магомадова рассмотрела языки программирования высокого уровня, особенности, недостатки и сложности в изучении, а также описала несколько легких алгоритмов [2]. В своей работе И. Ю. Просвирнина создала приложение «Морской бой», обладающее игровым искусственным интеллектом, в котором предусмотрен режим игры «Игрок против компьютера» [3]. В статье Ф.В. Патюченко, И.С. Слащев, А.В. Клименко, Л.А. Трегубенко были рассмотрены два подхода для создания программ на базе Windows, обоснование выбора одного из них [4]. Д.З. Хасаева, А.Ю. Демин в своей работе проанализировали графическую библиотека «3Dbodies» и привели пример ее использования [5]. В статье J. Li, Q.Liu, H.Su описано использование WPF для создания модуля виртуальной реальности портальной системы мониторинга поворотного крана [6]. S.Torsten, S.Toshiyuki, M. Derek в своем отчете описали интегрированную программную систему контроля качества, разработанную для значительного повышения производительности и качества анализа проб в лабораториях по анализу бериллия на территории комплекса Министерства энергетики США [7].

## **1.3 Цель исследования**

Цель исследования – применяя возможности среды разработки Visual Studio и языка программирования C#, создать программу для рисования, с базовыми функциями.

## **2. Материалы и методы**

### **2.1 Методы исследования**

Для создания программы будем использовать программное обеспечение Visual Studio, а также язык программирования C#.

Для проекта использовалась Windows Presentation Foundation (WPF) [8], платформа пользовательского интерфейса для создания клиентских приложений для настольных систем. Платформа разработки WPF поддерживает широкий набор компонентов для разработки приложений, включая модель приложения, ресурсы, элементы управления, графику, макет, привязки данных, документы и безопасность. Эта платформа является частью платформы .NET.

## **3 Результаты и дискуссия**

Начнём разработку приложения, с создания WPF приложения. Сразу назовем проект «Paint2.0» (рис. 1).

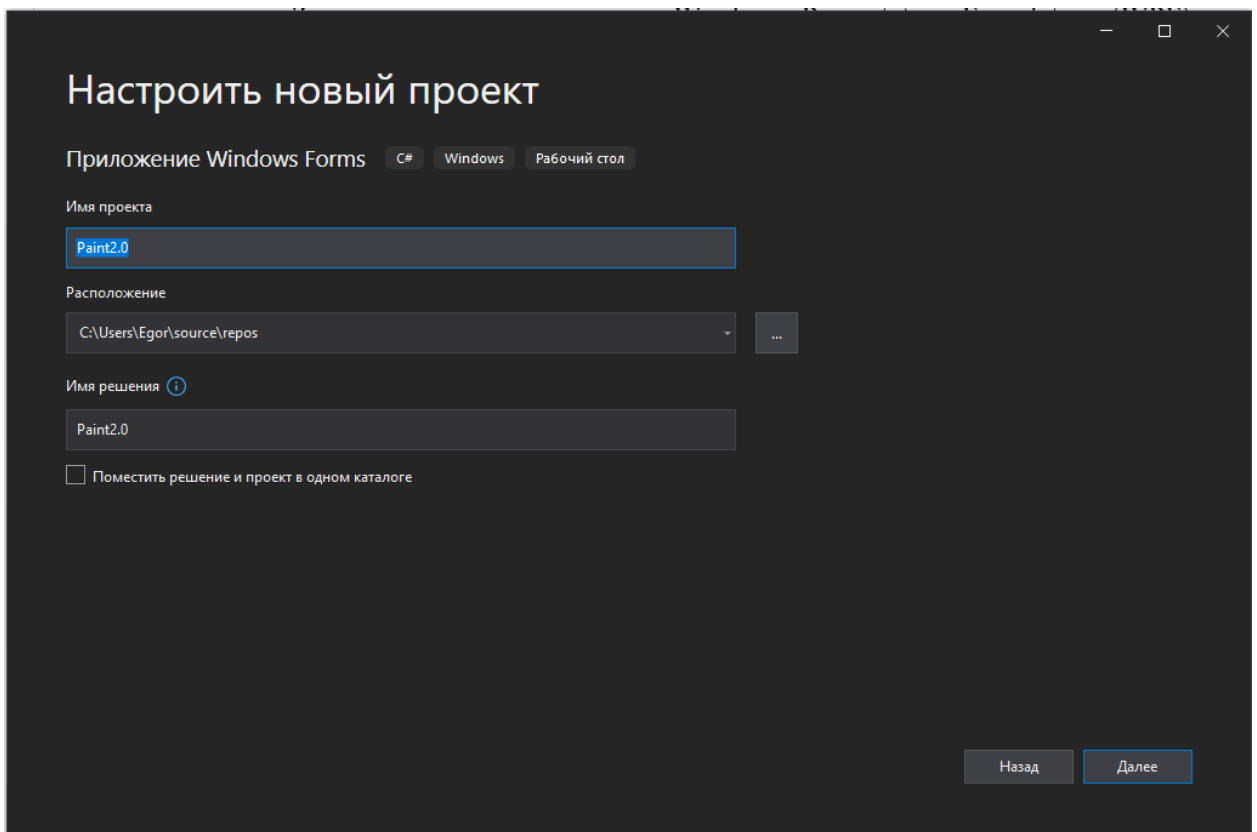


Рисунок 1. Создание проекта

И сразу же добавим элемент меню «MenuStrip», переименуем пункт меню в «File», а подпункты «Save» и «Open» (рис.2-3).

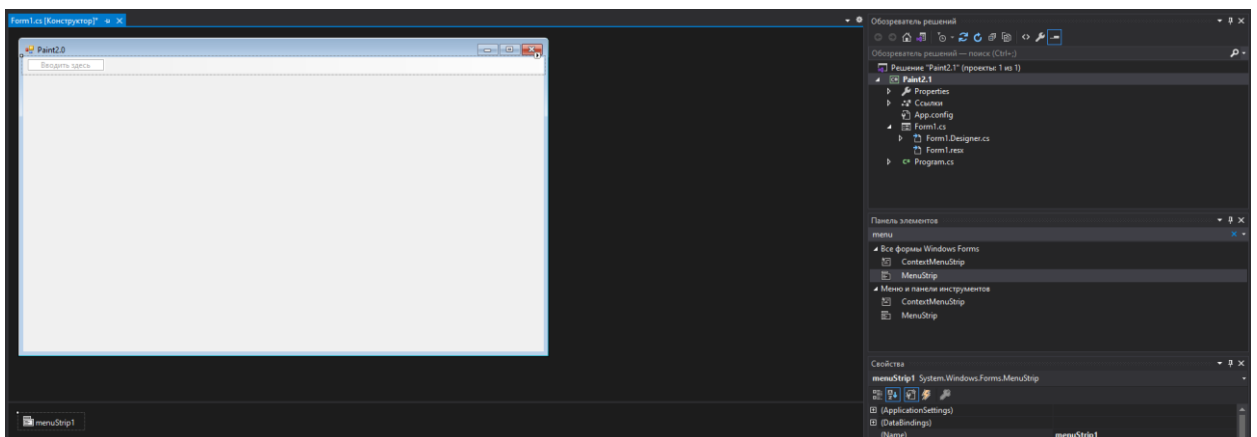


Рисунок 2. Создание элементов меню

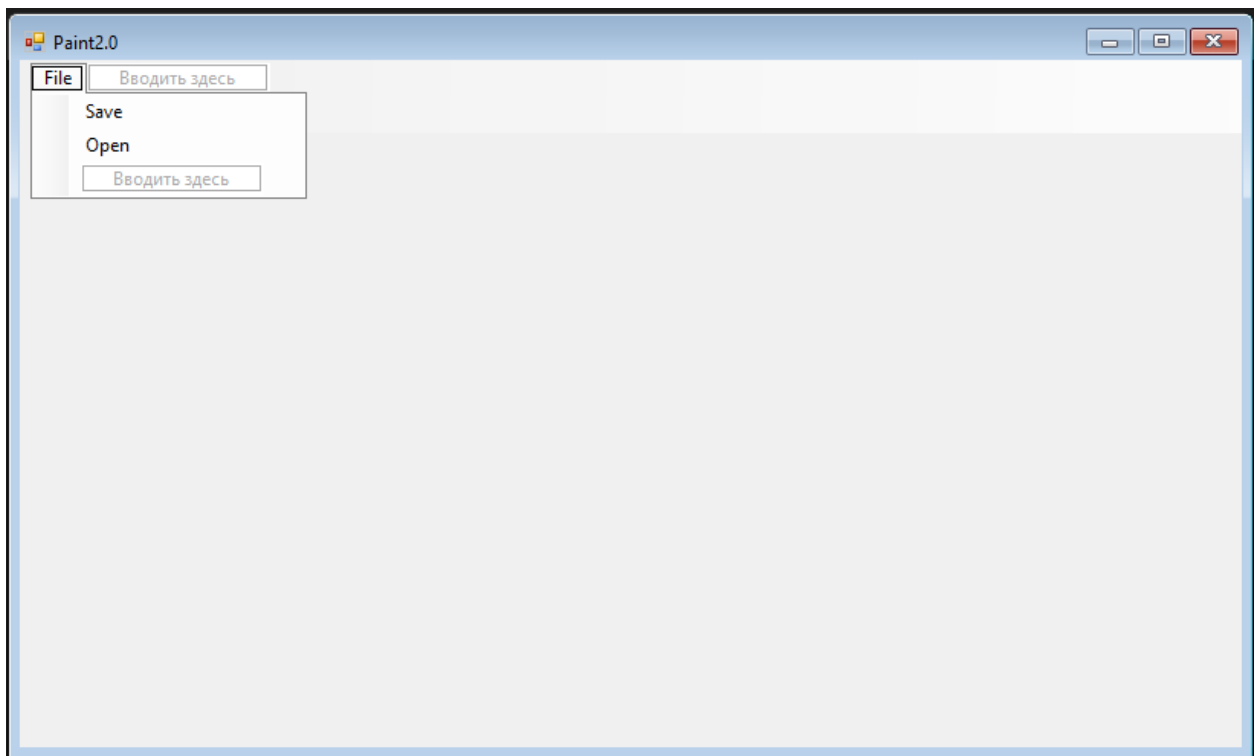


Рисунок 3. Создание элементов меню

Далее в форму добавляем панель инструментов «ToolStrip» и создаем там несколько кнопок, необходимых для выбора цвета пера, и 1 выпадающий список с выбором размера пера (рис.4).

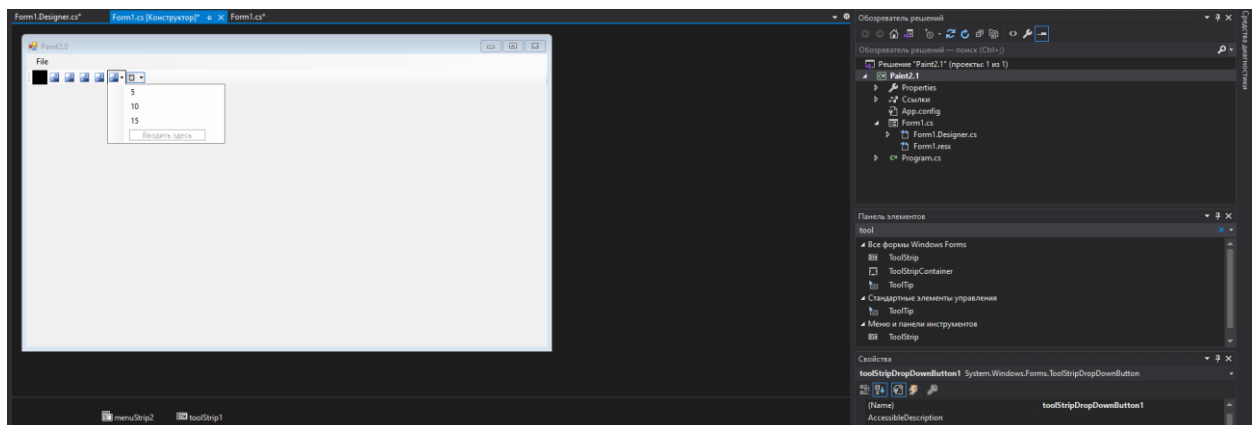


Рисунок 4. Создание панели инструментов

Панель инструментов представляет из себя кнопки и список с непонятными цифрами. Для исправления необходимо очистить изображение кнопки, поменять цвет в «BackColor» на другой, изменить авторедктирование размера, и изменить ширину на 50. По аналогии редактируем оставшиеся кнопки. (рис.5-9).

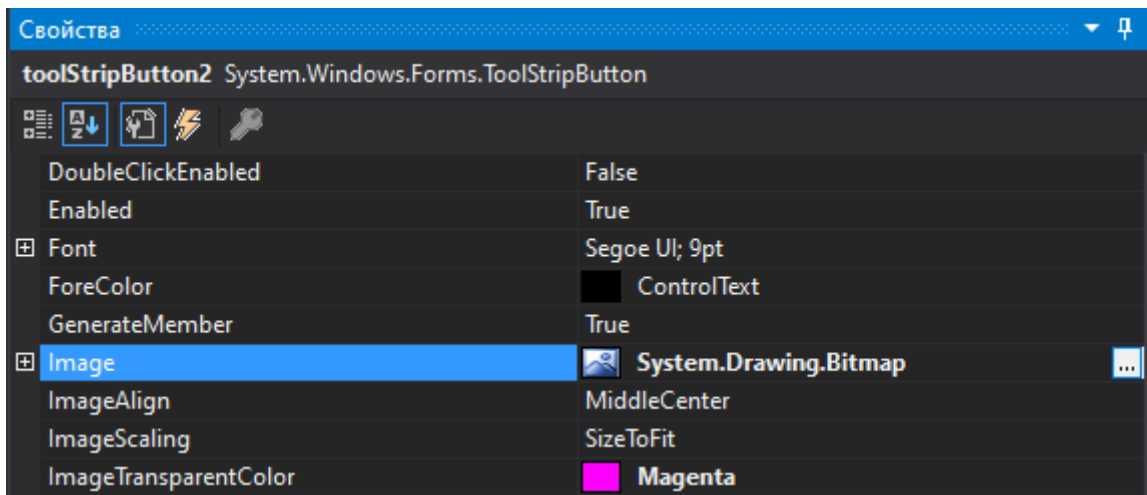


Рисунок 5. Удаление изображения с кнопки

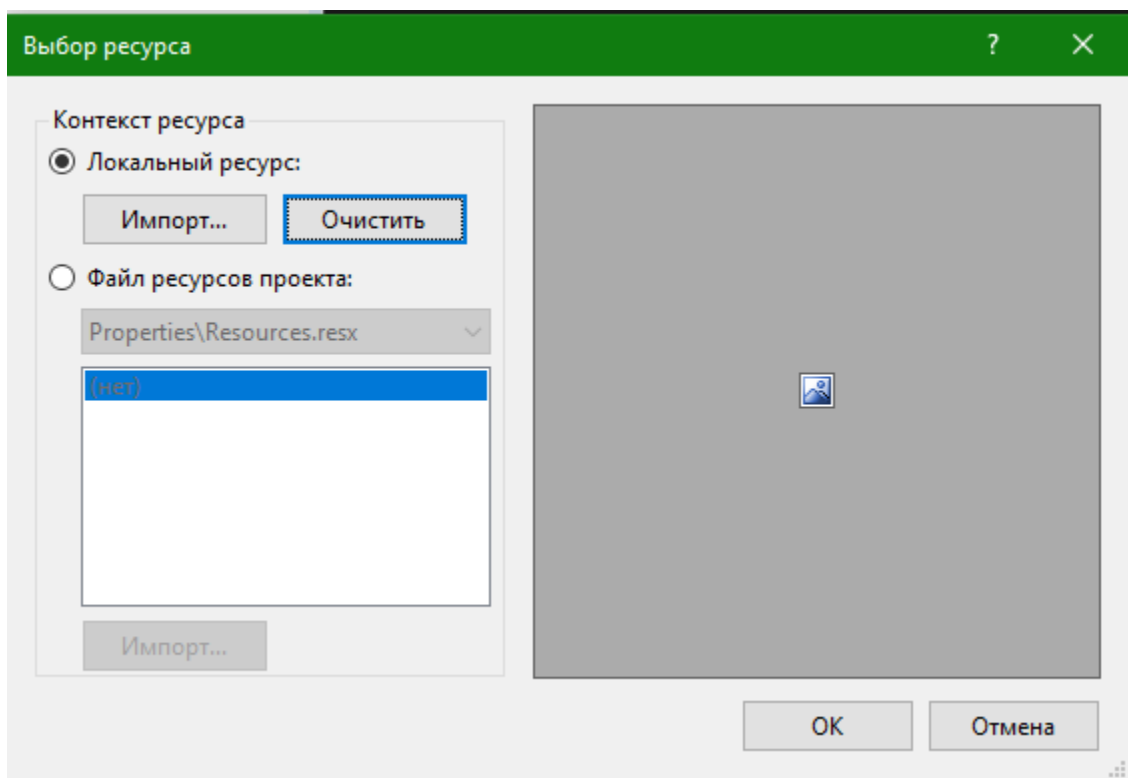


Рисунок 6. Удаление изображения с кнопки

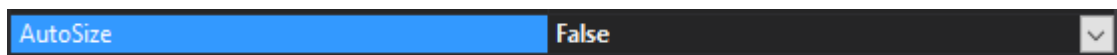


Рисунок 7. Отключение авто редактирования размера



Рисунок 8. Изменение ширины кнопки



Рисунок 9. Изменение цвета кнопки

У выпадающего списка в свойствах контент меняем на текст и пишем «PenSize» и убираем изображение (рис.10-11).

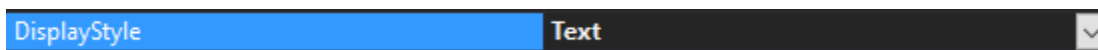


Рисунок 10. Редактирование контента

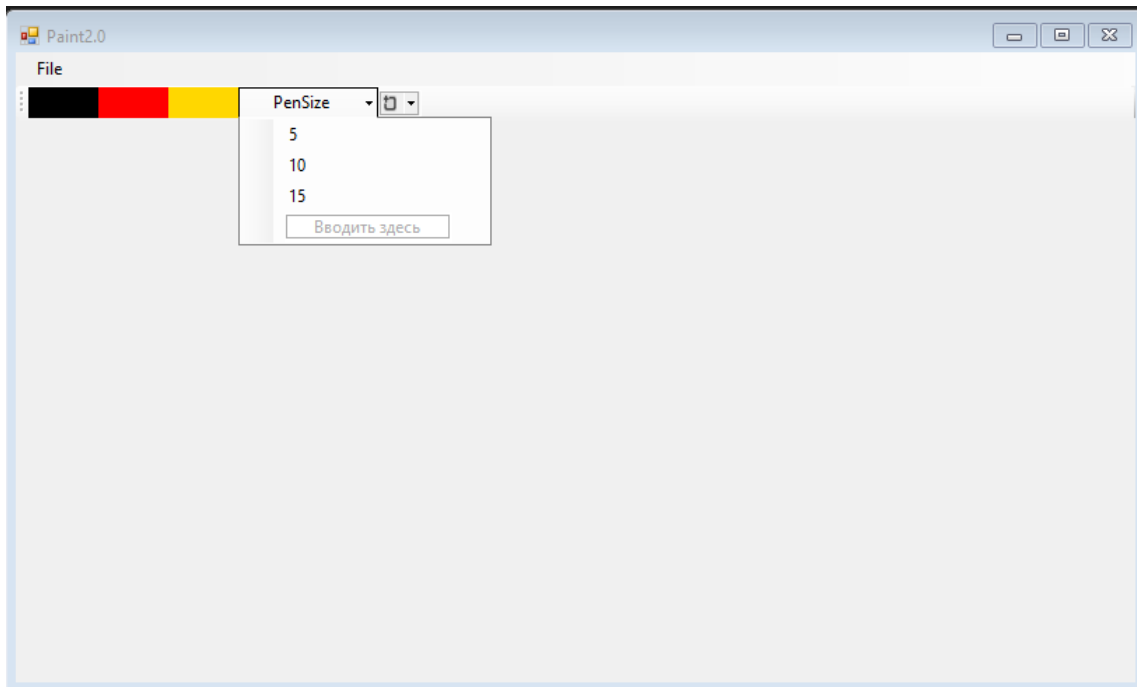


Рисунок 11. Готовый результат

Далее в качестве полотна добавляем «PictureBox» и посредством растягивания делаем его на все рабочее пространство (рис.12).

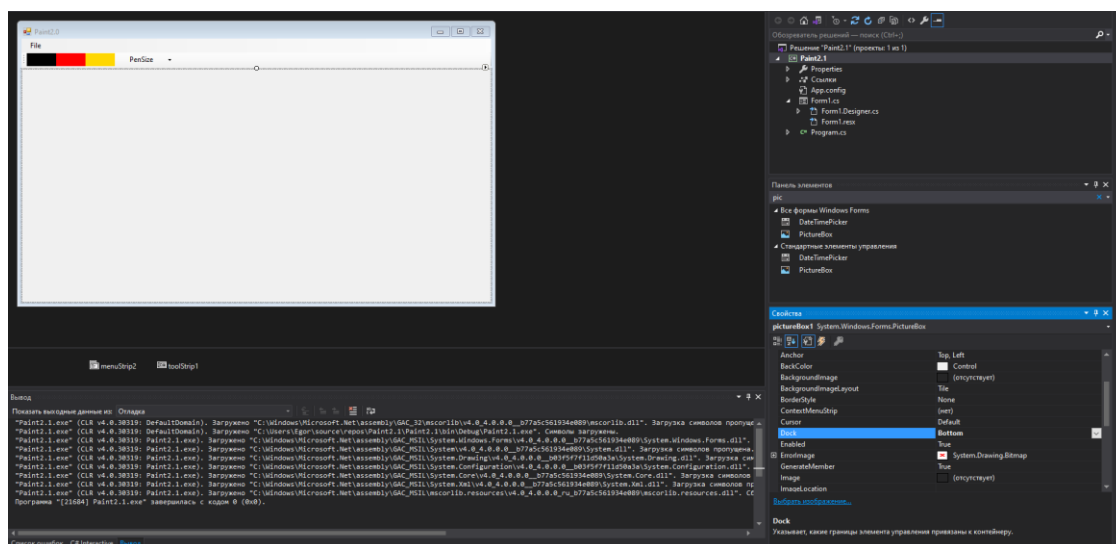


Рисунок 12. Добавление «PictureBox»

Далее необходимо изменить наименования кнопок и полотна для легкой инициализации в коде (рис.10).

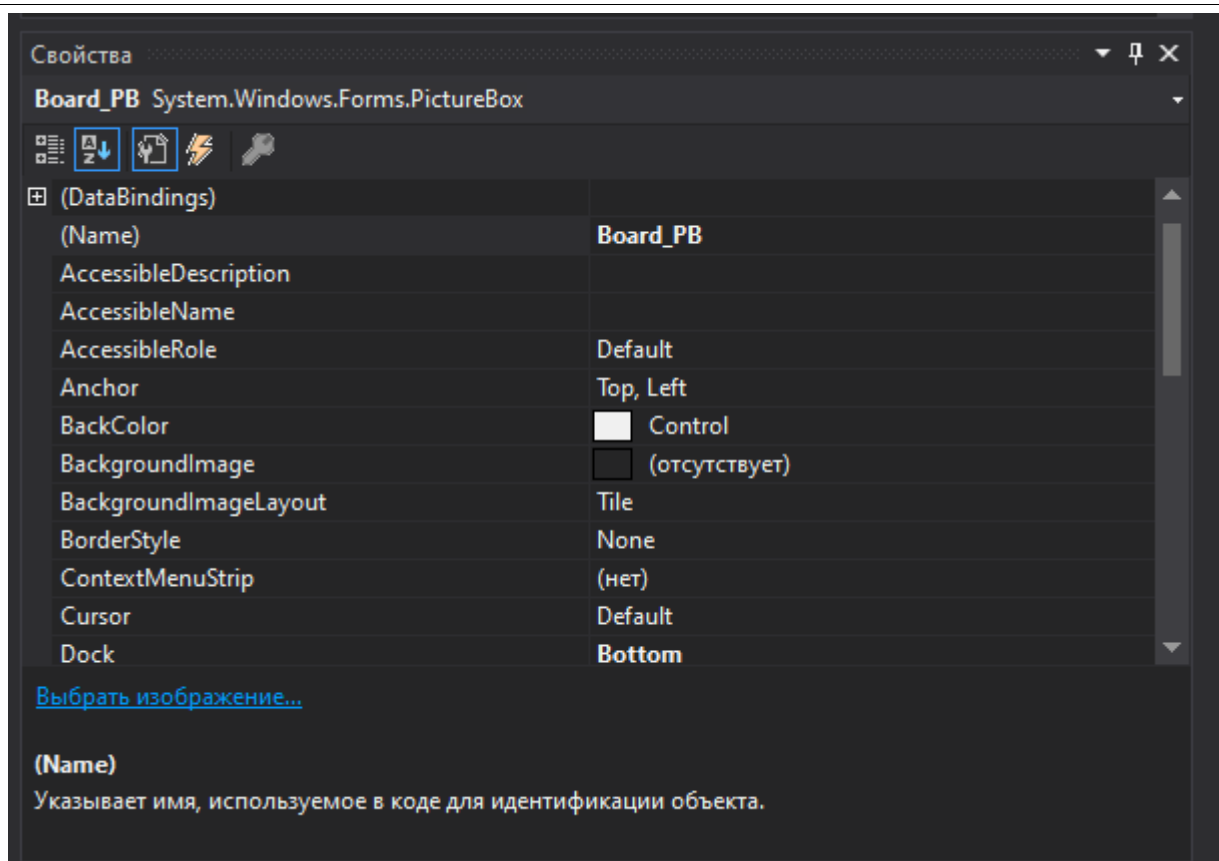


Рисунок 13. Изменение наименования элементов

Переходим к написанию программной логики. Начинаем с создания трех переменных. «Bitmap» для обозначения размера полотна, «Pen» для инициализации стандартного пера, и булеву переменную для проверки клика мышки (рис.14).

```
Bitmap bitmap = new Bitmap(1920, 1080);  
Pen pen = new Pen(Color.Black, 5);  
bool isDrawing = false;
```

Рисунок 14. Создание переменных

Далее необходимо добавить события, когда мышка двигается вниз и вверх (рис.15).

```
Ссылка: 2  
private void Board_PB_MouseDown(object sender, MouseEventArgs e)  
{  
    if (isDrawing == true)  
    {  
        isDrawing = false;  
    }  
    else  
    {  
        isDrawing = true;  
    }  
}
```

Рисунок 15. Обработка движения мыши вниз

Дублируем событие на движение мыши вверх (рис.16).

MouseDown	Board_PB_MouseDown
MouseEnter	
MouseHover	
MouseLeave	
MouseMove	Board_PB_MouseMove
MouseUp	Board_PB_MouseDown

Рисунок 16. Добавление события движение мыши вверх

Теперь создаем событие движение мыши, то есть добавляем функцию самого рисования посредством рисования небольших прямоугольников с длиной в 3 единицы и шириной в 1 единицу (рис.17).

```
ссылка: 1
private void Board_PB_MouseMove(object sender, MouseEventArgs e)
{
    if (isDrawing == false)
    {
        Graphics graphics = Graphics.FromImage(Bitmap);
        graphics.DrawRectangle(pen, e.X, e.Y, 3, 1);
        Board_PB.Image = Bitmap;
    }
}
```

Рисунок 17. Добавление функции рисования

Главная функция приложения уже работает, теперь необходимо активизировать кнопки смены цвета и размера пера (рис.18).

```
ссылка: 1
private void Black_Btn_Click(object sender, EventArgs e)
{
    pen.Color = Color.Black;
}

ссылка: 1
private void Red_Btn_Click(object sender, EventArgs e)
{
    pen.Color = Color.Red;
}

ссылка: 1
private void Yellow_Btn_Click(object sender, EventArgs e)
{
    pen.Color = Color.Gold;
}
```

Рисунок 18. Функция смены цвета пера

Для работоспособности кнопки изменения размера пера необходимо создать новую переменную «ThePenSize» типа «int» и задать начальное значение в 5 единиц (рис.19).



```
Bitmap Bitmap = new Bitmap(1920, 1080);  
Pen pen = new Pen(Color.Black, 5);  
int ThePenSize = 5;  
bool isDrawing = false;
```

Рисунок 19. Инициализация новой переменной

Изменяем в функции рисования размера пера на только что созданную переменную (рис. 20).

```
ссылка: 1  
private void Board_PB_MouseMove(object sender, MouseEventArgs e)  
{  
    if (isDrawing == false)  
    {  
        Graphics graphics = Graphics.FromImage(Bitmap);  
        graphics.DrawRectangle(pen, e.X, e.Y, ThePenSize, ThePenSize);  
        Board_PB.Image = Bitmap;  
    }  
}
```

Рисунок 20. Редактирование пера

Далее создаем события на кнопках изменения размера пера (рис. 21).

```
ссылка: 1  
private void PenSize_5_Click(object sender, EventArgs e)  
{  
    ThePenSize = 5;  
}  
  
ссылка: 1  
private void PenSize_10_Click(object sender, EventArgs e)  
{  
    ThePenSize = 10;  
}  
  
ссылка: 1  
private void PenSize_15_Click(object sender, EventArgs e)  
{  
    ThePenSize = 15;  
}
```

Рисунок 21. Создание событий изменения размера пера

Осталось только добавить функции сохранения и открытия изображения, для добавления функции сохранения добавляем элемент «saveFileDialog» на рабочую область и добавляем событие (рис.22-23).

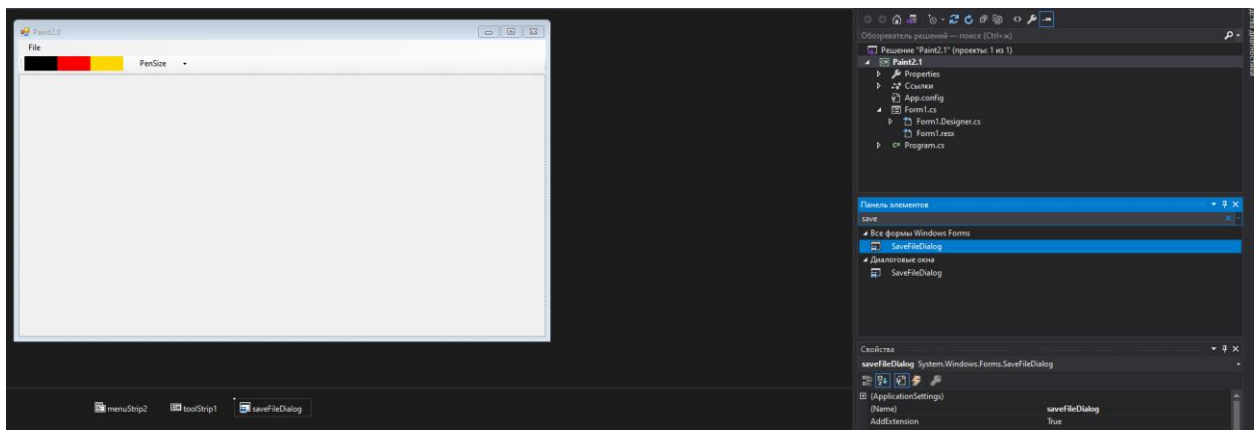


Рисунок 22. Добавление элемента «saveFileDialog»

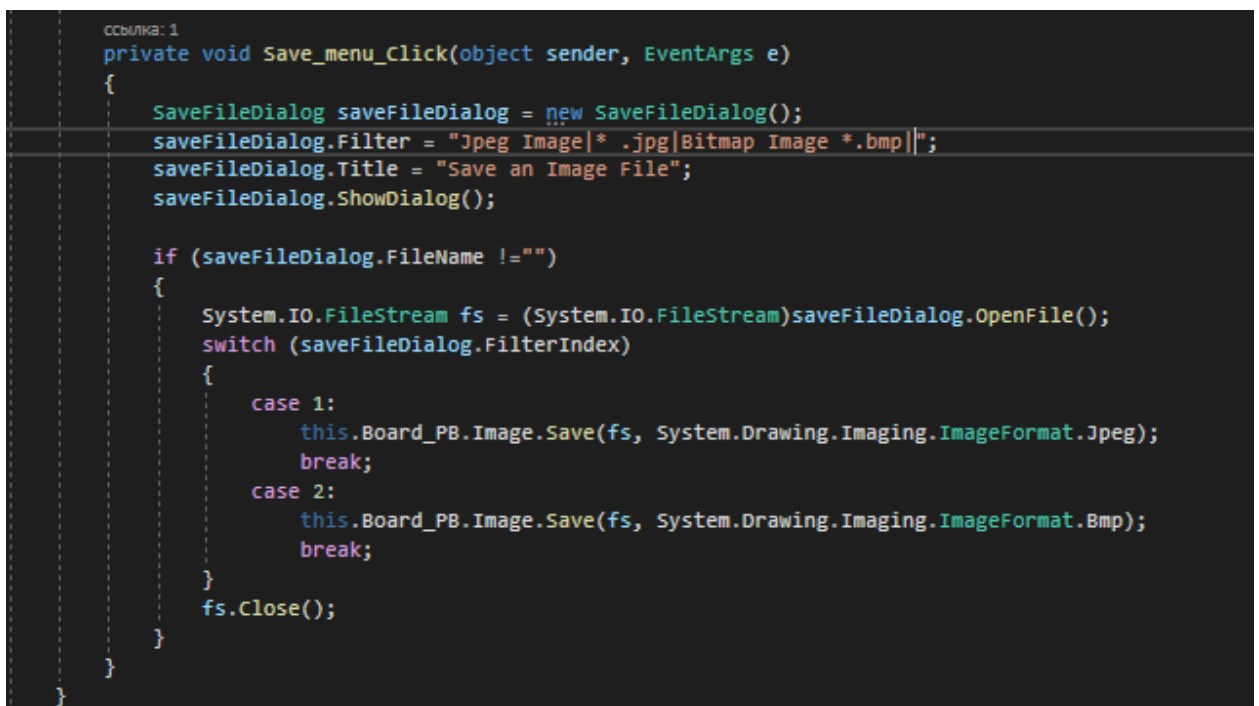


Рисунок 23. Добавление события

Для добавления функции открытия необходимо добавить новую переменную, и также событие (рис.24-25).

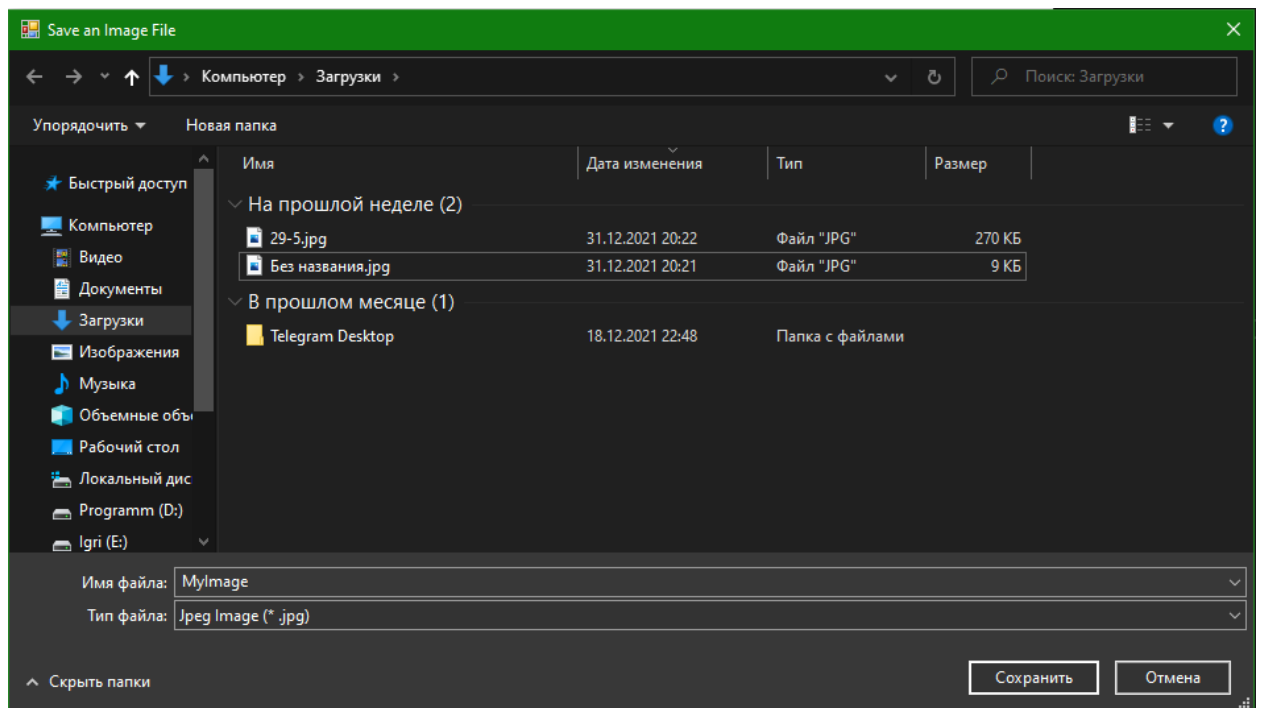


Рисунок 24. Добавление переменной

```
ССЫЛКА: 1
private void Open_menu_Click(object sender, EventArgs e)
{
    OpenFileDialog Op = new OpenFileDialog();
    DialogResult dr = Op.ShowDialog();
    if (dr == DialogResult.OK)
    {
        OpenedFile = Image.FromFile(Op.FileName);
        Board_PB.Image = OpenedFile;
    }
}
```

Рисунок 25. Добавление события

Осталось только проверить программу (рис.26-28).

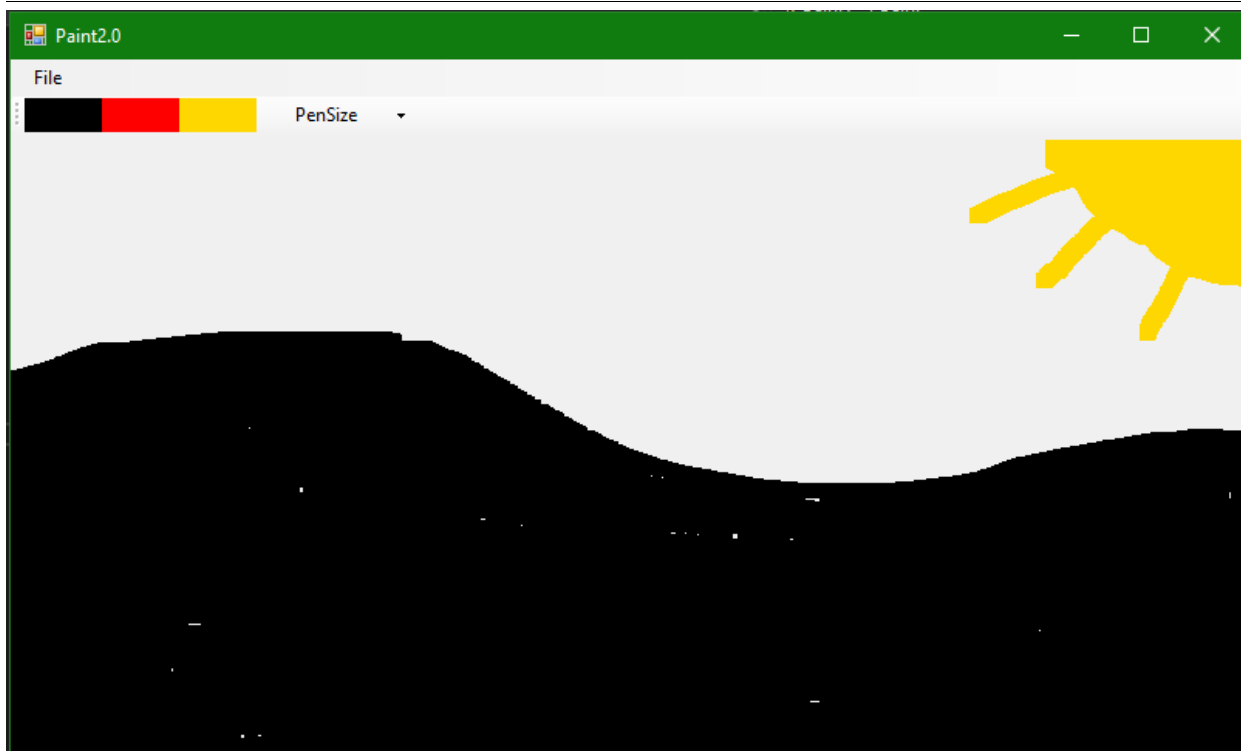


Рисунок 26. Основная функция рисования

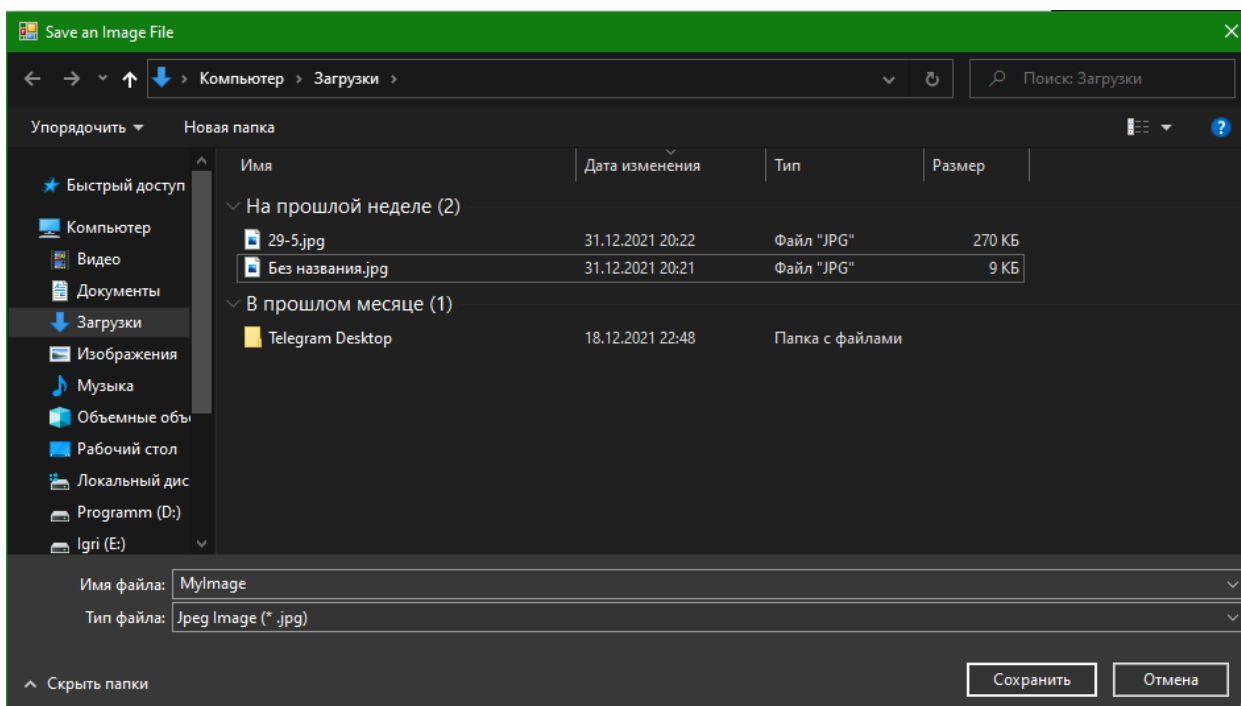


Рисунок 27. Функция сохранения изображения

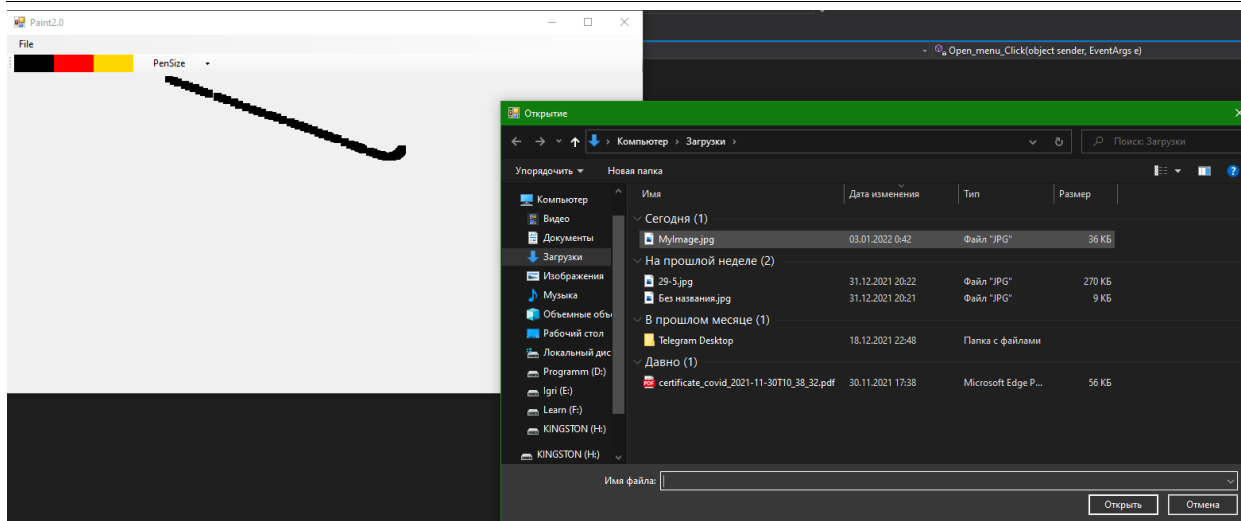


Рисунок 28. Функция открытия изображений

#### 4 Выводы

Были проанализированы существующие аналоги и методы разработки, а также выбрана среда разработки. Для реализации поставленной задачи отлично подошла разработка с помощью Visual Studio и языка программирования C#. Такой выбор заметно упростило разработку приложения, так как в интернете имеется достаточное количество документации. Во время создания приложения был полученный ценный опыт работы с этим средством разработки. В итоге было разработано приложения для рисования с базовым функционалом. Тесты приложения прошло хорошо, были исправлены незначительные ошибки. Созданное приложения имеет потенциал к развитию, а именно: добавление новых функций и улучшение интерфейса.

#### Библиографический список

1. Додобоев Н. Н., Кукарцева О. И., Тынченко Я. А. Современные языки программирования //Современные технологии: актуальные вопросы, достижения и инновации. 2014. №5. С. 81-85.
2. Магомадова З. С. Языки программирования высокого уровня //Разработка и применение наукоёмких технологий в эпоху глобальных трансформаций. 2020. №8. С. 94-96.
3. Просвирнина И. Ю., Егунова А. И., Аббакумов А. А. Среда разработки Microsoft Visual Studio на примере создания игры "Морской бой" //Интеграционные процессы в науке в современных условиях. 2017. С. 123-125.
4. Патюченко Ф.В., Слащев И.С., Клименко А.В., Трегубенко Л.А. Windows Form или Windows Presentation Foundation // Modern science. 2019. №7-2. С. 318-320.
5. Хасаева Д.З., Демин А.Ю. Разработка графической библиотеки для визуализации объектов робототехники на основе технологии Windows presentation foundation // XXVII Международная инновационно-

- 
- ориентированная конференция молодых ученых и студентов. 2015. С. 536-539.
6. Li J., Liu Q., Su H. Virtual Reality Method of Portal Slewing Crane Based on WPF // *Advances in Mechanical Engineering*. 2013. С. 1-5.
  7. Torsten S., Toshiyuki S., Derek M. Improving Sample Analysis Throughput and Quality with a .NET™-based, Real-Time QC Decision Support System // *Journal of the Association for Laboratory Automation*. 2003. №8. С. 107-112.
  8. WFP URL: <https://docs.microsoft.com/ru-ru/visualstudio/designers/getting-started-with-wpf?view=vs-2022>