

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



**МОСКОВСКИЙ  
ПОЛИТЕХ**

**КУРСОВАЯ РАБОТА**

По дисциплине «Разработка мобильных приложений»

**Тема курсовой работы**

**«Разработка приложения Быки и коровы»**

Группа	221-329
Студент	Никитин Олег Евгеньевич
Дата	08.04.2025
Преподаватель	<u>Васильев Денис Борисович</u>

2025 г

## ОГЛАВЛЕНИЕ

<b>ВВЕДЕНИЕ.....</b>	<b>3</b>
<b>1 ТРЕБОВАНИЯ К РАЗРАБАТЫВАЕМОЙ ПРОГРАММЕ.....</b>	<b>4</b>
1.1 Функциональные требования.....	4
1.2 Нефункциональные требования.....	4
<b>2 РЕАЛИЗАЦИЯ ПРОГРАММЫ.....</b>	<b>6</b>
2.1 Код программы.....	6
<b>3 ПРИМЕР РАБОТЫ ПРОГРАММЫ.....</b>	<b>9</b>
3.1 Контрольный пример работы программы.....	9
<b>ЗАКЛЮЧЕНИЕ.....</b>	<b>11</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....</b>	<b>12</b>

## **ВВЕДЕНИЕ**

Цель курсовой работы: разработать приложение «Клеточный автомат Игра Жизнь».

Задачи курсовой работы:

- a) сформулировать требования к программе;
- b) реализовать код программы.
- c) привести пример работы программы;

# **1 ТРЕБОВАНИЯ К РАЗРАБАТЫВАЕМОЙ ПРОГРАММЕ**

Разрабатываемая программа должна соответствовать последующим функциональным и нефункциональным требованиям

## **1.1 Функциональные требования**

Программа должна предоставлять следующий функционал:

### **1. Ввод и проверка догадок:**

- Пользователь вводит попытку (число фиксированной длины).
- Программа проверяет корректность ввода (необходимая длина, только цифры).

### **2. Игровая логика:**

- Генерация случайного числа без повторяющихся цифр (первая цифра не равна 0).
- Подсчёт количества «быков» (совпадение цифр на своих позициях) и «коров» (цифры присутствуют, но на других позициях) для каждой попытки.

### **3. Вывод истории попыток:**

- Отображение всех введенных пользователем вариантов с результатами проверки.

### **4. Таймер попытки:**

- Отсчёт времени, прошедшего с момента начала текущей попытки.
- Сброс таймера при каждой новой попытке и при запуске новой игры.

### **5. Управление игрой:**

- Возможность начать новую игру с генерацией нового случайного числа.
- Автоматический сброс текущего состояния игры и истории попыток после верного ответа.

## **1.2 Нефункциональные требования**

Программа должна удовлетворять следующие нефункциональные требования:

1. Необходимо использовать язык программирования C# и фреймворк .NET 8.0.
2. Необходимо создать графический интерфейс с использованием WPF.

## 2 РЕАЛИЗАЦИЯ ПРОГРАММЫ

### 2.1 Код программы

Полный код программы находится на [Github](#)

Листинг 1 – Генерация случайного числа

```
private string GenerateSecretNumber()
{
    Random rnd = new Random();
    int[] digits = Enumerable.Range(0, 10).ToArray();
    // Перемешиваем массив цифр
    digits = digits.OrderBy(x => rnd.Next()).ToArray();

    // Если первая цифра равна 0, меняем её с первой ненулевой цифрой
    if (digits[0] == 0)
    {
        int nonZeroIndex = Array.FindIndex(digits, x => x != 0);
        (digits[0], digits[nonZeroIndex]) = (digits[nonZeroIndex], digits[0]);
    }

    return string.Join("", digits.Take(NumberLength));
}
```

## Листинг 2 – Обработчик нажатий кнопки «Проверить»

```
private void CheckGuess_Click(object sender, RoutedEventArgs e)
{
    string guess = GuessTextBox.Text.Trim();

    // Проверка корректности ввода: число нужной длины и только цифры
    if (guess.Length != NumberLength || !guess.All(char.IsDigit))
    {
        MessageBox.Show($"Пожалуйста, введите {NumberLength}-значное число.",
            "Ошибка ввода", MessageBoxButton.OK, MessageBoxImage.Error);
        return;
    }

    int bulls = 0, cows = 0;
    for (int i = 0; i < NumberLength; i++)
    {
        if (guess[i] == secretNumber[i])
        {
            bulls++;
        }
        else if (secretNumber.Contains(guess[i]))
        {
            cows++;
        }
    }

    TimeSpan attemptTime = DateTime.Now - attemptStartTime;

    ResultTextBlock.Text += $"\\nПопытка: {guess} - Быки: {bulls}, Коровы: {cows} (время: {attemptTime.ToString(@"mm:ss")}");

    // Если число угадано, выводим сообщение о победе и начинаем новую игру
    if (bulls == NumberLength)
    {
        MessageBox.Show($"Поздравляем! Вы угадали число: {secretNumber} за {attemptTime.ToString(@"mm:ss")}",
            "Победа", MessageBoxButton.OK, MessageBoxImage.Information);
        StartNewGame();
    }
    else
    {
        GuessTextBox.Clear();
    }
}
```

### Листинг 3 – Реализация интерфейса

```
<Window x:Class="appl.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:appl"
        mc:Ignorable="d"
        Title="Быки и Коровы" Height="350" Width="400">
    <Grid>
        <StackPanel Margin="10">
            <TextBlock Text="Введите ваш вариант:" FontSize="14"/>
            <TextBox x:Name="GuessTextBox" Margin="0,5" FontSize="14"/>
            <Button Content="Проверить" Click="CheckGuess_Click" Margin="0,5" FontSize="14"/>
            <ScrollViewer Height="150" VerticalScrollBarVisibility="Auto" Margin="0,10">
                <TextBlock x:Name="ResultTextBlock" Text="Результаты игры:" FontSize="14" TextWrapping="Wrap"/>
            </ScrollViewer>
            <Button Content="Начать заново" Click="RestartGame_Click" Margin="0,5" FontSize="14"/>
        </StackPanel>

        <TextBlock x:Name="TimerTextBlock"
            HorizontalAlignment="Right"
            VerticalAlignment="Bottom"
            Margin="10"
            FontSize="14"
            Foreground="DarkBlue"
            Text="00:00"/>
    </Grid>
</Window>
```



### 3 ПРИМЕР РАБОТЫ ПРОГРАММЫ

#### 3.1 Контрольный пример работы программы

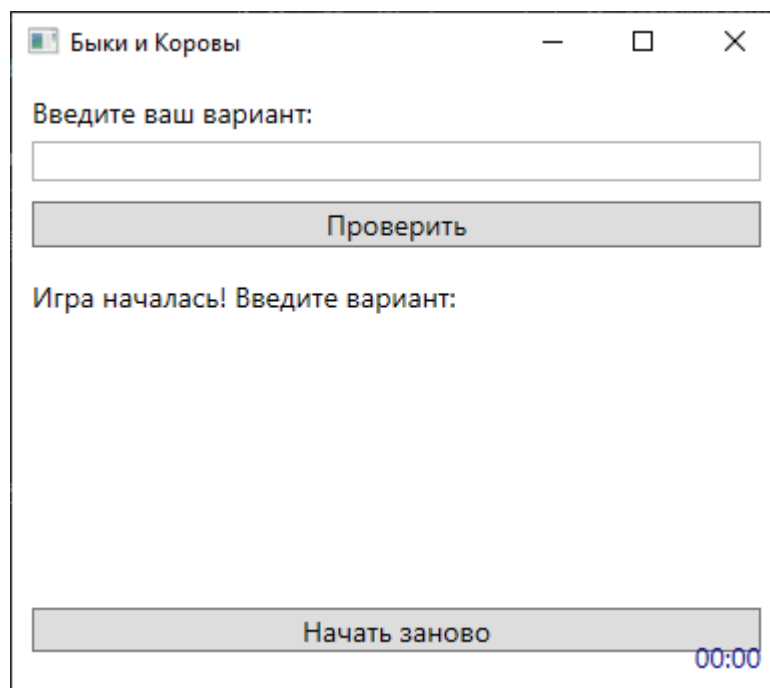


Рисунок 1 – Начальное состояние окна игры

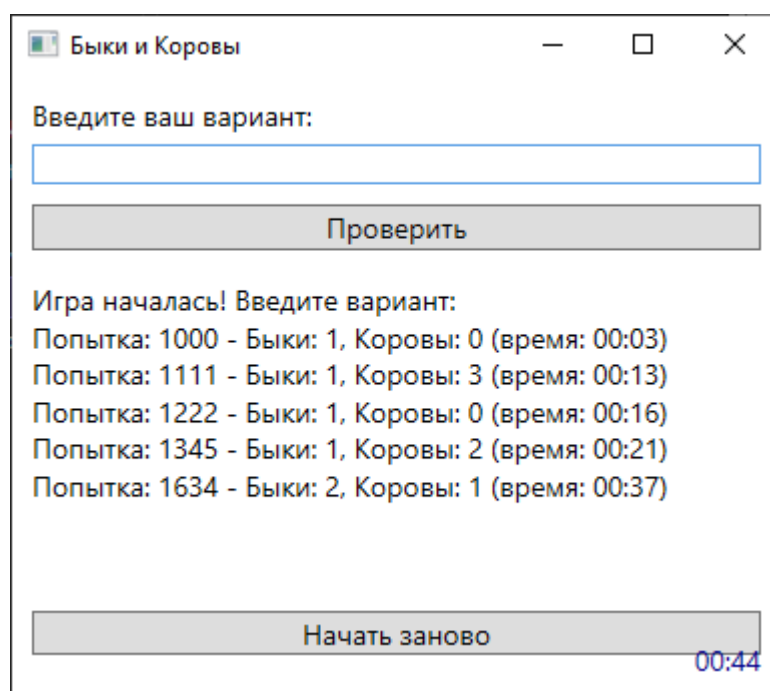


Рисунок 2 – Процесс игры

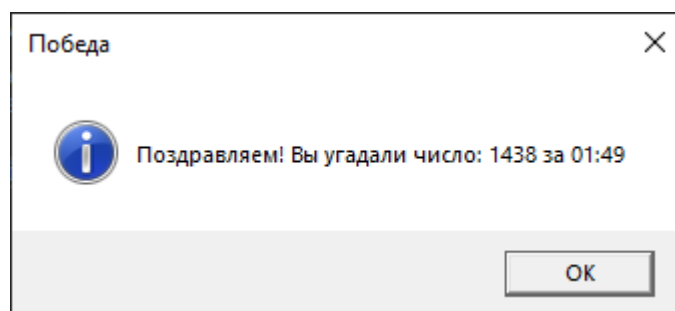


Рисунок 3 – Конец игры

## **ЗАКЛЮЧЕНИЕ**

В ходе разработки приложения «Быки и коровы» использовались современные технологии .NET 8.0 и WPF для создания удобного, компактного и функционального решения. Приложение реализует основные механизмы игры: генерацию случайного числа, проверку вариантов пользователя, подсчёт количества «быков» и «коров», а также отображение истории попыток с возможностью прокрутки. Дополнительно, реализован таймер, показывающий время, затраченное на текущую попытку, что позволяет пользователю оценить скорость своих догадок. Вся функциональность сосредоточена в единой структуре кода, что делает приложение простым для понимания, дальнейшей отладки и возможного расширения функционала.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Адам Натан. WPF 4. Подробное руководство. – Символ-Плюс, 2011. – 880 с. – ISBN 978-5-93286-196-7.
2. Мартин, Р. Чистая архитектура. Искусство разработки программного обеспечения / Р. Мартин. – Санкт-Петербург: Питер, 2021. – 352 с. – ISBN 978-5-4461-0772-8.