Приложение WPF

«Игра: Математико»

Бобровников Никита, группа ИСпП-21-2

Микрюкова Анастасия, группа ИСпП-21-2

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc163314666)

[ОПИСАНИЕ РАБОТЫ ПРИЛОЖЕНИЯ 4](#_Toc163314667)

[БЛОК-СХЕМА 5](#_Toc163314668)

# ВВЕДЕНИЕ

Игра "Математико" — это итальянская логическая игра для нескольких игроков. Правила игры довольно просты и выводятся на экран (или в отдельное окно) во время работы программы. В игре реализована возможность игры от одного до четырех человек.

Имеется набор из 52 карточек, на которых записаны числа от 1 до 13, причем карточки с каждым из этих чисел встречаются четырежды. У каждого игрока имеется квадратное поле с 25 клетками. Программа случайным образом извлекает какую-либо из имеющихся карточек и выдает записанное на ней число. Каждый игрок заносит это число в одну из клеток квадрата на своем поле. После того, как число записано, перемещать его в другую клетку запрещается. Затем программа случайным образом извлекает следующую карточку и т.д. Так продолжается до тех пор, пока не будут заполнены все клетки квадрата.

Результат игрока оценивается числом набранных им очков, зависящим от способа размещения чисел в клетках квадрата. Победителем считается тот, кто набирает наибольшее количество очков. Подсчет очков производится по следующей таблице:

|  |  |  |
| --- | --- | --- |
| **Комбинации чисел** | **В ряду или столбце** | **По диагонали** |
| За 2 одинаковых числа | 10 очков | 20 очков |
| За 2 пары одинаковых чисел | 20 очков | 30 очков |
| За 3 одинаковых числа | 40 очков | 50 очков |
| За 3 одинаковых числа и два других одинаковых числа | 80 очков | 90 очков |
| За 4 одинаковых числа | 160 очков | 170 очков |
| За 5 последовательных чисел, но не обязательно по порядку расположенных | 50 очков | 60 очков |
| За три раза по 1 и два раза по 13 | 100 очков | 110 очков |
| За числа 1, 13, 12, 11 и 10, но не обязательно по порядку расположенных | 150 очков | 160 очков |
| За 4 единицы | 200 очков | 210 очков |

# ОПИСАНИЕ РАБОТЫ ПРИЛОЖЕНИЯ

Программа представляет собой игру под названием "Математико", реализованную в виде приложения для Windows с использованием технологии WPF (Windows Presentation Foundation).

Основные компоненты программы:

MainWindow: Главное окно приложения, в котором отображается кнопка "Start Game" для запуска игры.

GameWindow: Окно игры, в котором пользователь играет против компьютера (бота).

Функциональность программы:

* инициализация: при запуске игры генерируется игровое поле, колода карт и начальные очки игрока и бота;
* ход игрока: игрок поочередно размещает карты на игровом поле, выбирая ячейки с помощью кликов мыши;
* ход бота: после каждого хода игрока бот автоматически делает свой ход, выбирая случайную ячейку на игровом поле;
* оценка очков: после размещения каждой карты оцениваются очки игрока и бота на основе комбинаций карт на игровом поле;
* завершение игры: игра завершается, когда все ячейки на игровом поле заполнены, после чего определяется победитель (игрок, бот или ничья) и отображается соответствующее сообщение;
* рестарт игры: после завершения игры пользователь может нажать кнопку "restart", чтобы начать игру заново, сбросив очки и очищая игровое поле;
* возврат в главное меню: пользователь может вернуться в главное меню, нажав кнопку "back".

Программа обеспечивает интерактивное и увлекательное игровое окружение, где пользователь может соревноваться с ботом, стараясь собрать наибольшее количество очков.

# БЛОК-СХЕМА

Основной ход программы:

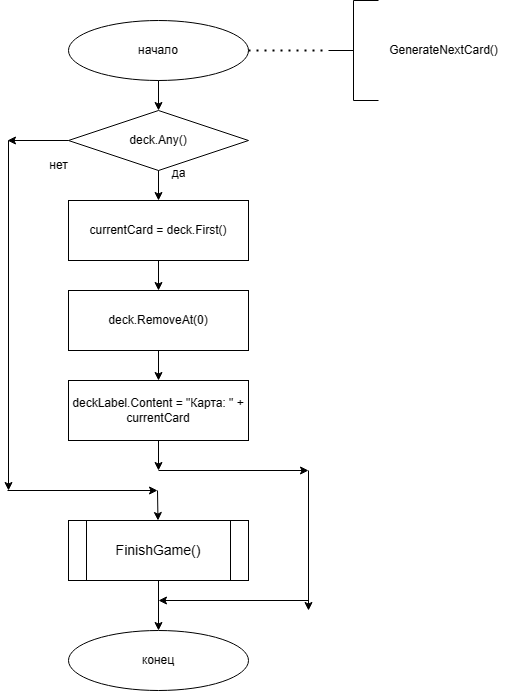


Функция `GenerateNextCard()` в программе отвечает за генерацию следующей игровой карты из колоды карт. Каждый раз, когда игрок или бот делает ход, они берут следующую карту из колоды. Если колода пуста, это означает, что все карты были использованы, и игра завершается.

Конкретные шаги функции `GenerateNextCard()` включают:

* проверка, есть ли еще карты в колоде (`deck`);
* если в колоде есть карты, извлечение первой карты из колоды;
* удаление извлеченной карты из колоды;
* обновление метки на игровом интерфейсе, отображающей текущую карту.

Эта функция выполняется автоматически при старте игры и после каждого хода игрока или бота, чтобы обеспечить постоянное предоставление новых карт для игры.

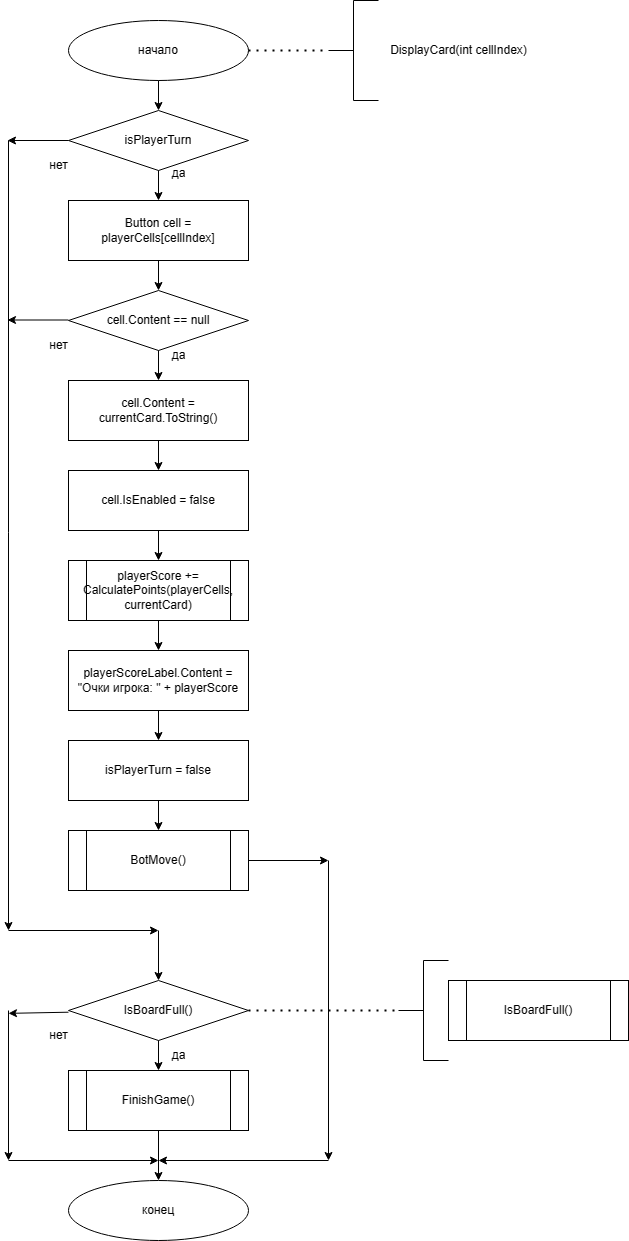


Функция DisplayCard(int cellIndex) в программе отвечает за отображение карты в указанной ячейке игрового поля. Когда игрок кликает на свободную ячейку, эта функция вызывается для отображения карты в выбранной ячейке и выполнения необходимых действий, таких как подсчет очков и изменение текущего игрового состояния.

Конкретные шаги функции DisplayCard() включают:

* проверка, чей сейчас ход (игрока или бота);
* проверка, свободна ли выбранная ячейка для размещения карты;
* отображение карты в выбранной ячейке игрового поля;
* вычисление и обновление счета игрока (если ходит игрок);
* передача хода боту (если ходит игрок);
* проверка на заполненность игрового поля;
* завершение игры, если поле полностью заполнено.

Эта функция активируется при клике игрока на определенную ячейку игрового поля для размещения карты.

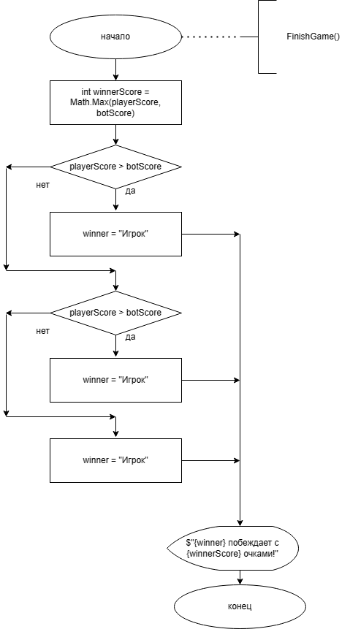


Функция FinishGame() в программе отвечает за завершение игры и определение победителя. После того как все ячейки на игровом поле заполнены, вызывается эта функция для определения победителя и вывода сообщения о завершении игры.

Основные шаги функции FinishGame() включают:

* определение победителя на основе набранных очков игрока и бота;
* определение наивысшего счета и идентификация победителя или объявление ничьей;
* вывод сообщения о завершении игры с указанием победителя и его очков;
* завершение работы игры.

Эта функция активируется при заполнении всех ячеек на игровом поле и используется для определения победителя и завершения игры.

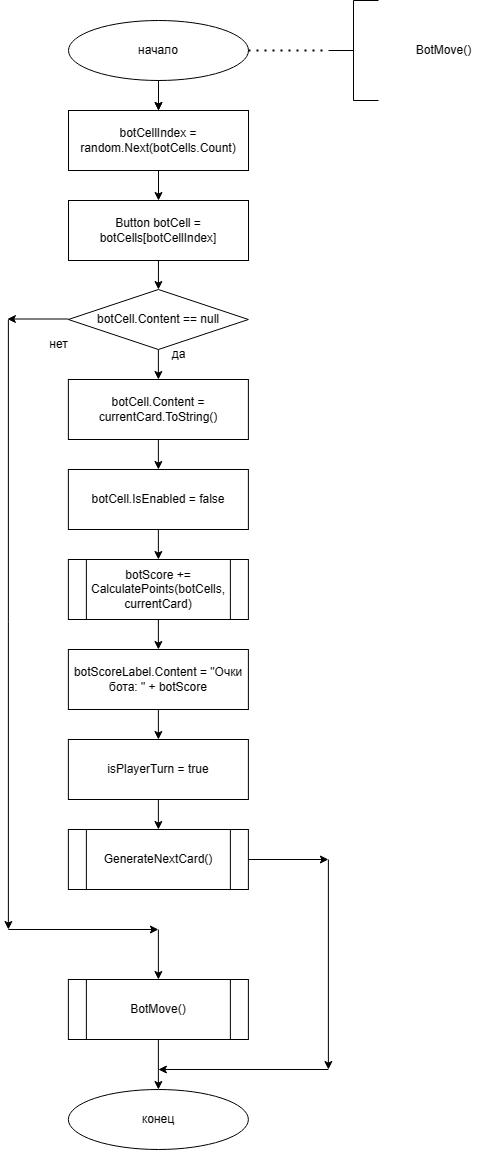


Функция BotMove() в программе отвечает за ход компьютерного противника (бота) в игре. Когда наступает очередь бота ходить, эта функция вызывается для выполнения хода бота и размещения карты на игровом поле.

Основные шаги функции BotMove() включают:

* выбор случайной свободной ячейки на игровом поле;
* проверка, свободна ли выбранная ячейка для размещения карты;
* отображение карты в выбранной ячейке игрового поля;
* вычисление и обновление счета бота;
* передача хода игроку после завершения хода бота;
* генерация следующей карты после завершения хода бота.

Эта функция активируется после того, как игрок сделал свой ход, и отвечает за выполнение хода бота в игре.



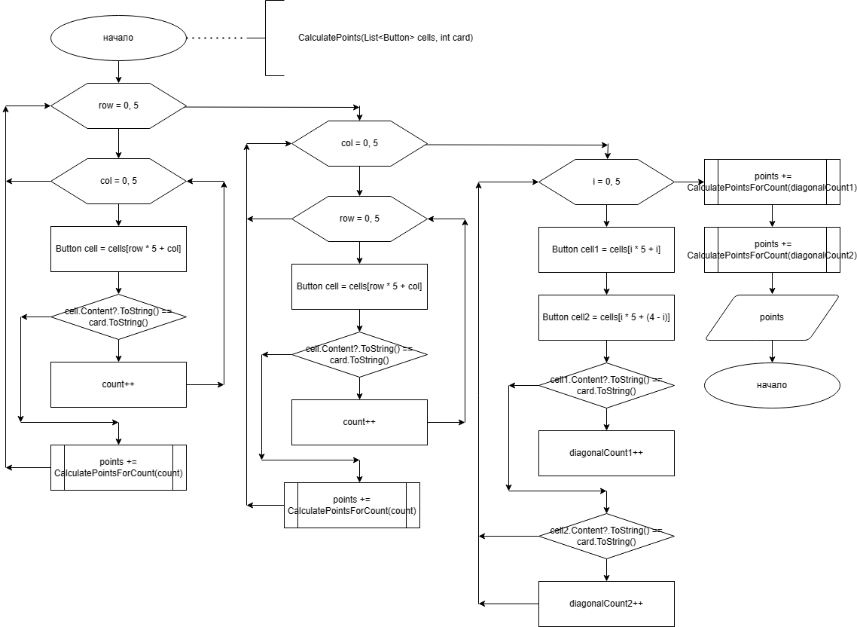
Функция CalculatePoints(List<Button> cells, int card) в программе отвечает за вычисление очков игрока или бота в зависимости от комбинаций карт на игровом поле.

Основные шаги функции CalculatePoints() включают:

* перебор всех комбинаций карт на игровом поле;
* проверка каждой строки, столбца и диагонали на наличие определенного количества карт, совпадающих с текущей картой;
* вычисление очков за каждую комбинацию с учетом количества карт и их значения;
* суммирование полученных очков.

Функция использует вспомогательную функцию CalculatePointsForCount(int count), чтобы определить количество очков за каждую комбинацию карт.

Эта функция вызывается при каждом ходе игрока или бота для оценки текущего состояния игрового поля и назначения очков за обнаруженные комбинации карт.

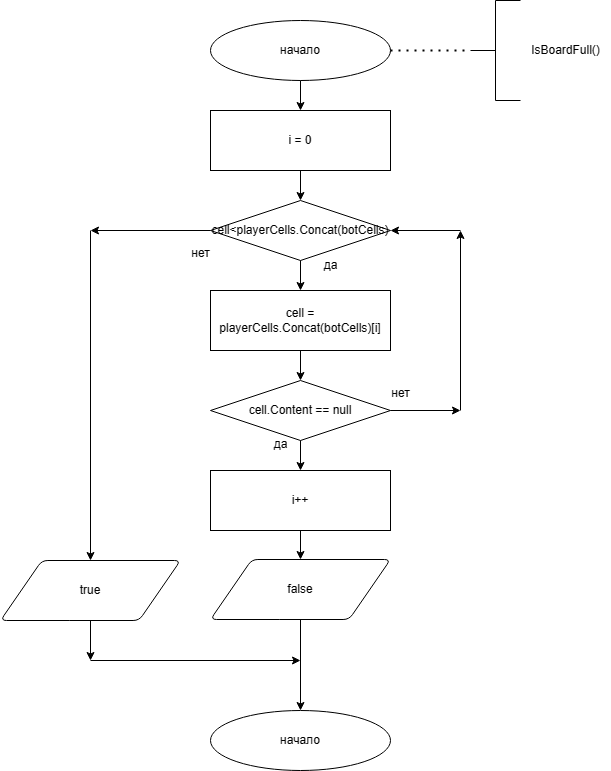


Функция IsBoardFull() проверяет, заполнено ли игровое поле картами полностью.

Основные шаги функции включают:

* перебор всех клеток на игровом поле, как для игрока, так и для бота;
* проверка каждой клетки на наличие карты;
* если хотя бы одна клетка пуста, функция возвращает false, что означает, что игровое поле еще не заполнено полностью;
* если все клетки на игровом поле заполнены картами, функция возвращает true, указывая на то, что игровое поле полностью заполнено.

Эта функция используется для определения, когда игра должна завершиться из-за полного заполнения игрового поля без возможности совершения дальнейших ходов.



Функция CalculatePointsForCount вычисляет количество очков, полученных за определенное количество карт одного номинала.

Основные шаги функции включают:

* принимает на вход количество карт одного номинала;
* в зависимости от этого количества, определяет количество очков, которые игрок или бот получают за эти карты. например, за 2 карты - 10 очков, за 3 - 40 очков, за 4 - 160 очков, за 5 - 50 очков;
* если количество карт не соответствует ни одному из этих вариантов, функция возвращает 0 очков.

Эта функция используется внутри другой функции CalculatePoints, которая подсчитывает общее количество очков для определенной комбинации карт.

