**Эйнор Леонард 2-ИП**

**Отчёт по практическому заданию 1**

Необходимо реализовать игру “Жизнь”.

**В проекте задачи содержится 1 файл:**

**Game.cs**

Файл содержит основной код со всеми функциями, необходимыми для реализации данного задания.

**Глобальные переменные**

Grafica – Содержит экземпляр класса Graphics для рисования

CellSize – Содержит размер клетки

RedCells – Массив красных клеток

BlueCells – Массив синих клеток

Rows – Содержит кол-во строк

Columns – Содержит кол-во столбцов

**Описание конструктора класса:**

**Алгоритм:**

После инициализации компонентов программы корректируется местоположение компонентов.

**Описание конструктора формы:**

У формы были изменены свойства: text, name и size.

К форме привязан обработчик изменения размеров формы.

В конструкторе были созданы следующие объекты:

4 Button – Переименованные в StartButton, PauseButton, CleanButton, ExitButton. К каждой привязано событие Click и изменено св-во text.

PictureBox – Переименованный в GamePlace. Изменено свойство size. Привязано событие MouseClick.

**Описание методов и функций:**

1. **Private void StartGame()**

Функция запускает таймер, благодаря чему начинается генерация поколений.

**Алгоритм:**

Если таймер включен выйти из функции, иначе запустить таймер.

1. **Private int CountNeighbors(int x, int y, bool[,] a)**

Функция подсчитывает кол-во соседей клетки.

**Переменные:**

CountNeighbors – Хранит количество соседей

NeighboringCol – Хранит соседей в столбцах

NeighboringRow – Хранит соседей в строках

Samoproverka – Содержит логическое значение, отражающее не проверяется ли сама клетка.

IsAlive – Содержит логическое значение, отражающее жива ли клетка

**Алгоритм:**

Создаём переменную для хранения кол-ва соседей.

В двух циклах находим соседей в соседних столбцах, в соседних строках.

Проверяем на самопроверку.

Проверяем жива ли клетка.

Если клетка жива и это не самопроверка, то прибавить соседа.

Затем вернуть итоговое кол-во соседей.

1. **Private void NextGeneration()**

Функция рассчитывает следующее поколение.

**Переменные:**

NewRedCells – Массив следующего поколения красных

NewBlueCells – Массив следующего поколения синих

NeighborsRed – Содержит кол-во красных соседей

IsAliveRed – Содержит значение, отражающее красная ли живая клетка

NeighborsBlue – Содержит кол-во синих соседей

IsAliveBlue – Содержит значение, отражающее синяя ли живая клетка

**Алгоритм:**

1. **Private void TimerTick(object sender, EventArgs e)**

Каждый тик таймера рассчитывается новое поколение.

**Алгоритм:**

Запустить функцию следующего поколения.

1. **Private void PauseClick(object sender, EventArgs e)**

Инвертирует состояние таймера.

**Алгоритм:**

Включить таймер, если выключен и наоборот.