Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Омский государственный технический университет»

Факультет *информационных технологий и компьютерных систем*

Кафедра *«Автоматизированные системы обработки информации и управления»*

КУРСОВОЙ ПРОЕКТ

по дисциплине ***Программирование***

на тему ***Генератор «филвордов»***

Пояснительная записка

Шифр проекта **043-КП-09.03.04-№9/25-ПЗ**

Студента (ки) *Даниловой Александры Юрьевны*
фамилия, имя, отчество полностью

Курс *1* Группа ***ПИН-181***

Направление (специальность) 09.03.04 – Программная инженерия
код, наименование

Руководитель
ученая степень, звание

Богатов Р.Н.
фамилия, инициалы

Выполнил (а) 28.05.2019
дата, подпись студента (ки)

К защите 28.05.2019
дата, подпись руководителя

Проект (работа) защищен (а) с оценкой хор

Набранные баллы: 50 (в семестре) 30 (на защите) 80 (Итого)

Омск 2019

Реферат

Пояснительная записка по курсовому проекту 19 с., 6 ч., 6 рис., 2 таб, 1 источ., 1 прил.

ПРОГРАММИРОВАНИЕ, ЯЗЫК C#, WINDOWS FORMS, АЛГОРИТМ ГЕНЕРАЦИИ, РЕКУРСИЯ.

Объект работы: алгоритм и программа, генерирующая филворды.

Цель: Создать алгоритм генерации филвордов.

Методы: Рекурсия.

Результаты: После разработки курсового проекта была создана рабочая программа, генерирующая филворды и позволяющая задать размеры прямоугольного поля. Работу программы можно считать эффективной по времени, так как построение полей разных размеров занимает малое количество времени.

Содержание

Введение

Курсовой проект по дисциплине “Программирование”, 1 курс. В проекте использовался язык программирования C#.

Актуальность: Филворды («Венгерский кроссворд») – это разновидность головоломок, простая и понятная игра, которая популярна во многих странах.

Цель: Создать алгоритм генерации филвордов.

Задача: Разработать и реализовать алгоритм генерации «филвордов» (англ. FillWord) на основе существующего набора существительных. Пользователю предоставить возможность задать размеры прямоугольного поля и составить исходный набор существительных из случайной выборки по базе русских слов.

Методы: В программе используется рекурсия для заполнения поля словами.

1 Постановка задачи

В данном курсовом проекте, необходимо разработать и реализовать алгоритм генерации «филвордов» на основе существующего набора существительных. Пользователю предоставить возможность задать размеры прямоугольного поля и составить исходный набор существительных из случайной выборки по базе русских слов.

На выходе программы представлено поле, заполненное существительными.

2 Схемы алгоритмов

На рисунке 1 представлена основная схема работы программы.

Рисунок 1

На рисунках 2, 3 представлены подробные алгоритмы функций рекурсии и заполнения слов.

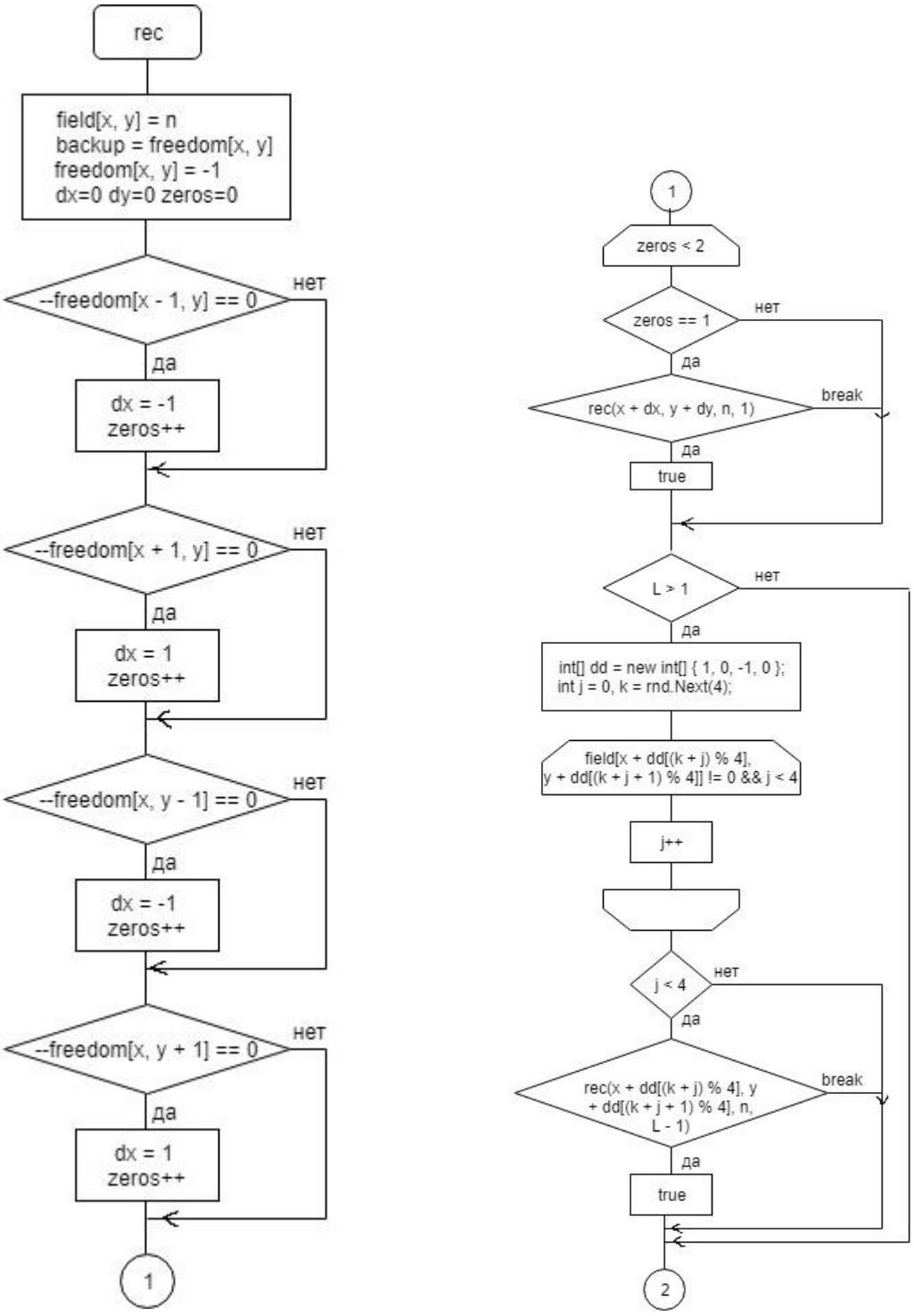

Рисунок 2

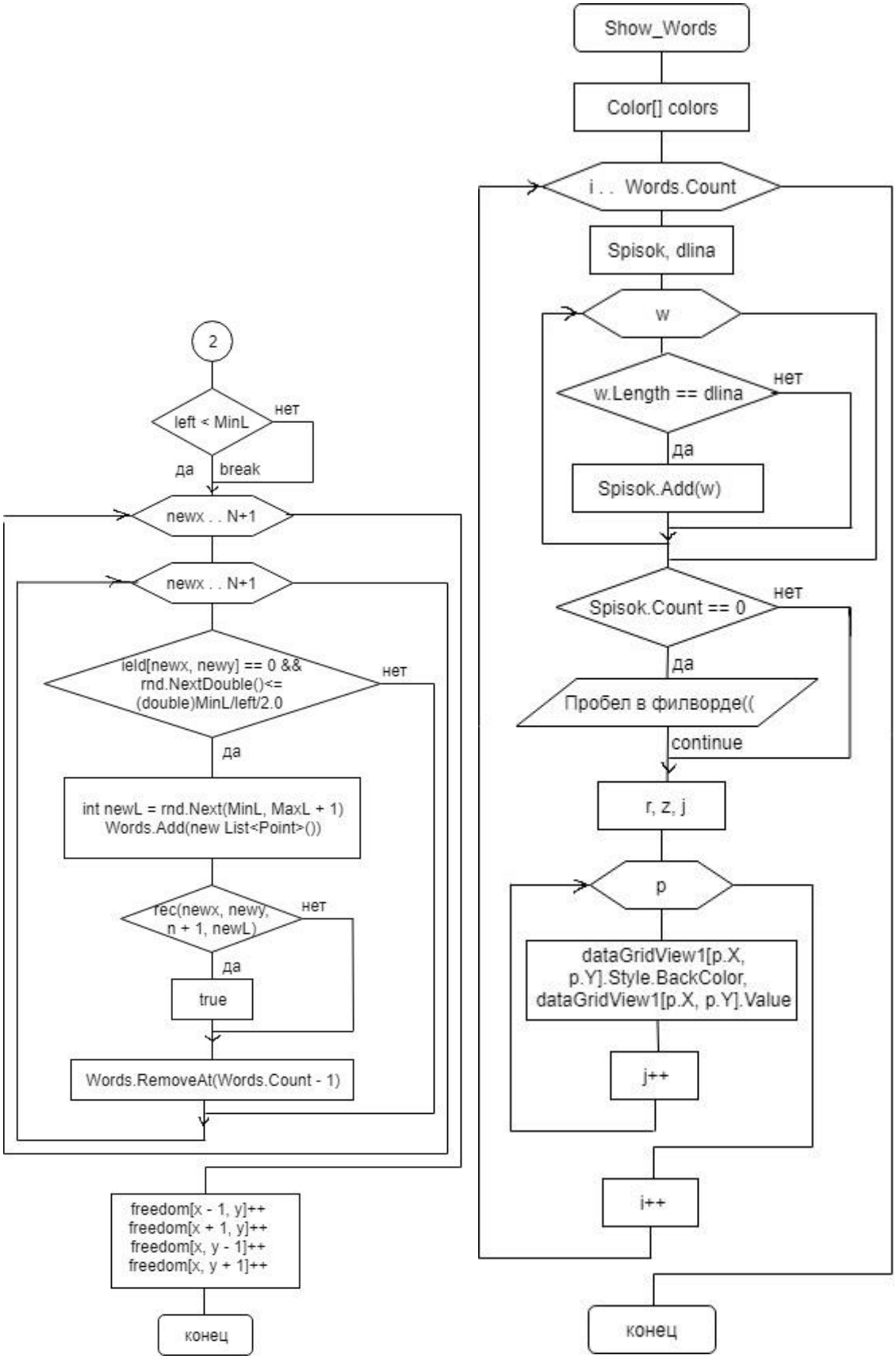
Show_Words
Color[] colors
i . . Words.Count
Spisok, dlina
w
w.Length == dlina
нет
да
Spisok.Add(w)
Spisok.Count == 0
нет
да
Пробел в филворде((
continue
r, z, j
p
dataGridView1[p.X, p.Y].Style.BackColor, dataGridView1[p.X, p.Y].Value
j++
i++
конец
2
left < MinL
нет
да
break
newx . . N+1
newx . . N+1
ield[newx, newy] == 0 && rnd.NextDouble()<= (double)MinL/left/2.0
нет
да
int newL = rnd.Next(MinL, MaxL + 1)
Words.Add(new List<Point>())
rec(newx, newy, n + 1, newL)
нет
да
true
Words.RemoveAt(Words.Count - 1)
freedom[x - 1, y]++
freedom[x + 1, y]++
freedom[x, y - 1]++
freedom[x, y + 1]++
конец

Рисунок 3

3 Структура данных

Наиболее важные переменные и прочие структуры данных использующихся в программе приведены в таблице 1 (где N – количество элементов в структуре, M – длина строки). А также объяснено, для чего они применяются.

Таблица 1

Структура	Размер (в байтах)	Назначение
int MinL, MaxL, N, left	4 x 4	Минимальные и максимальные длины слов, длина/ширина поля
int[,] freedom int[,] field	N x 4	Массивы свобод и поля
List<List<Point>> Words	12 x N	Список слов
List<string> Dict	N x M	Список существительных
List<string> Spisok	N * M	Список слов, с нужной длиной

4 Аспекты реализации на C#

Программа реализована с помощью Windows Forms (интерфейс программирования приложений, отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework).

Использовались такие элементы как Button (кнопка), DataGridView (таблица), Lable (надпись), numericUpDown (выбор числа из определенного диапазона), textBox (текстовое поле)

Элемент DatdGridView используется для создания прямоугольного поля и заполнения его словами. В numericUpDown предоставляется возможность изменять размер поля. Кнопка Button необходима для запуска построения филвода.

5 Руководство пользователя

После запуска программы в numericUpDown предоставляется возможность изменить размер поля.

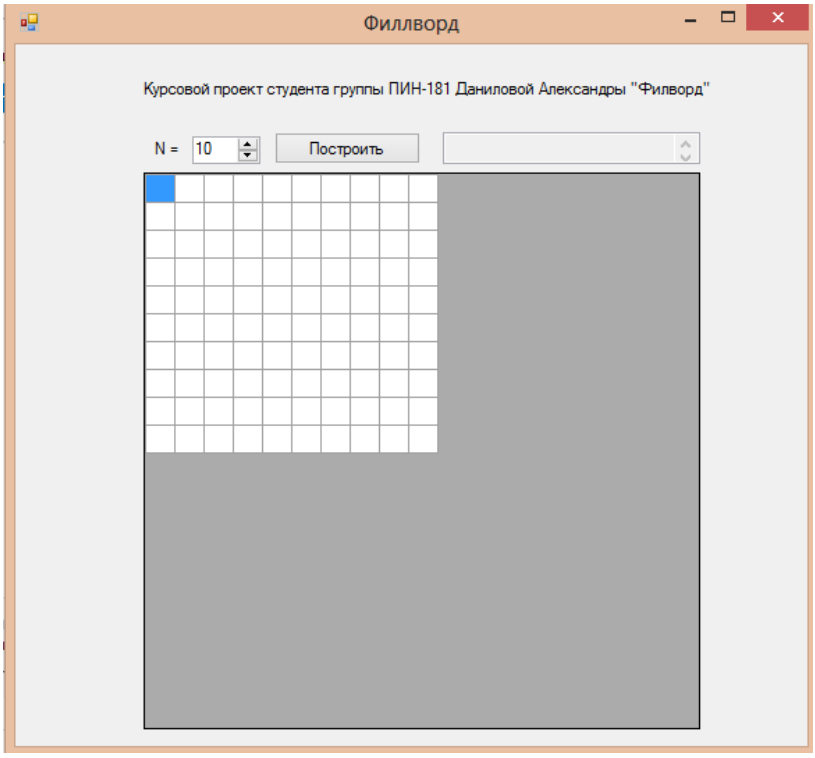

Рисунок 4

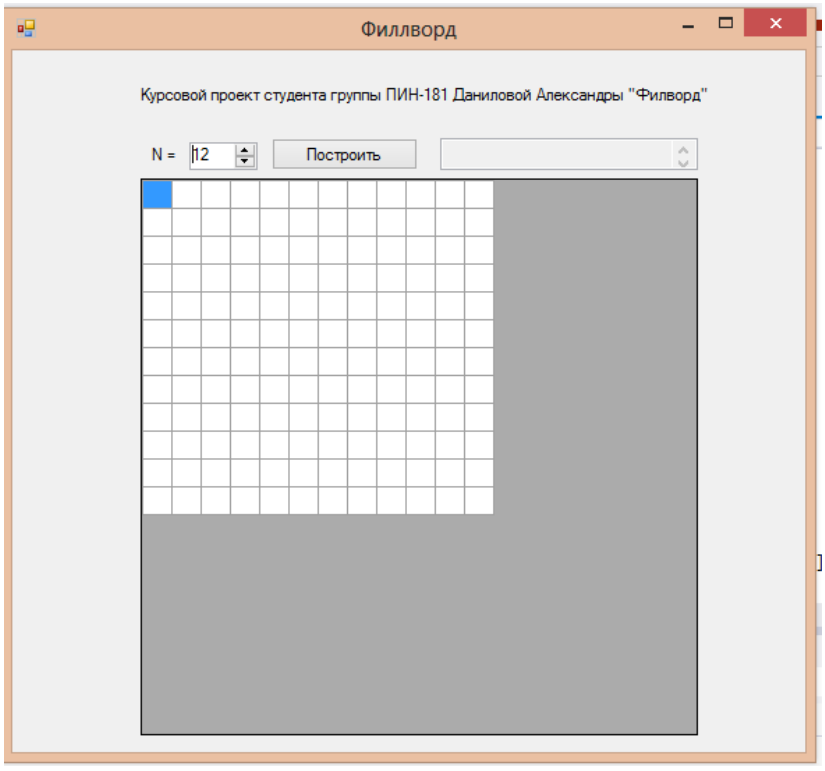

Рисунок 5

Далее после нажатия кнопки “Построить” в dataGridView формируется филворд, каждое слово которого выделено разными цветами. В textBox выводится время, за которое был построен филворд.

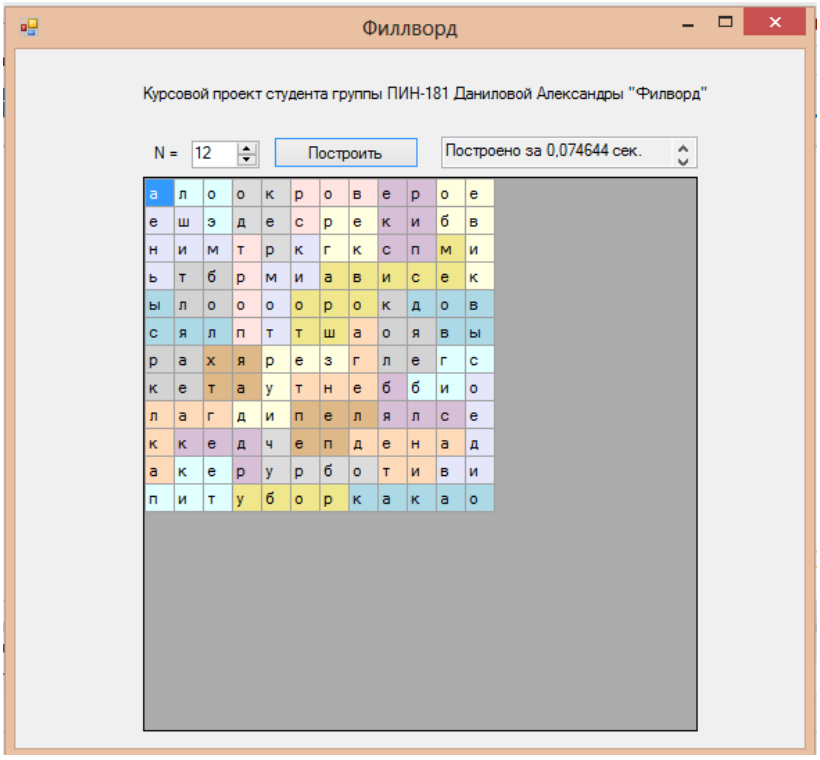

Рисунок 6

6 Методика тестирования

В ходе тестирования использовались словари с разными наборами слов.

В таблице 2 приведены результаты тестирования программы.

Таблица 2 – Результаты тестирования

№	Условие	Предполагаемый результат	Итог
1	Словарь, включающий все части речи	Вывод только существительных	+
2	Словарь, включающий существительные как нарицательные, так и собственные	Вывод и нарицательных, и собственных существительных	+
3	Маленький словарь, содержащий существительные длиной 4-5 букв	Неполное заполнение поля	+
4	Словарь, состоящий только из прилагательных	Вывод субстантивированных существительных	+
5	Словарь, состоящий только из глаголов	Заполнение поля цифрами	+

Заключение

В ходе курсового проекта удалось реализовать программу, генерирующую филворды. Главным достоинством является эффективность по времени, так как составление поля занимает небольшое количество времени. Программа понятна и проста в использовании.

Список использованных источников

1 ГОСТ 19.701-90, Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения – Стандартинформ, 04.03.2010. – 158 с

Приложение А
(обязательное)

Исходный код программы

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        const int MinL = 4, MaxL = 6;
        int N;
        Random rnd = new Random();
        int[,] freedom;
        int[,] field;
        int left;
        List<List<Point>> Words;
        List<string> Dict = new List<string>();

        public Form1()
        {
            InitializeComponent();
            numericUpDown1.Value = 10;
            string[] r1 = Properties.Resources.lemma.Split(new char[] { '\n', '\r' } ,
StringSplitOptions.RemoveEmptyEntries);
            foreach (string s in r1)
            {
                string[] r2 = s.Split(' ');
                if (r2[3]=="noun")
                    Dict.Add(r2[2]);
            }
        }

        private void numericUpDown1_ValueChanged(object sender, EventArgs e)
        {
            N = dataGridView1.ColumnCount = dataGridView1.RowCount =
(int)numericUpDown1.Value;
            dataGridView2.ColumnCount = dataGridView2.RowCount = N;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            dataGridView1.ColumnCount = dataGridView1.RowCount = 0;
            dataGridView1.ColumnCount = dataGridView1.RowCount = N;

            freedom = new int[N + 2, N + 2];
            field = new int[N + 2, N + 2];

            for (int i = 0; i < N; i++)
                for (int j = 0; j < N; j++)
                {
```

```
                    freedom[i+1, j+1] = 4 - ((i == 0 || i == N - 1) ? 1 : 0) - ((j == 0
|| j == N - 1) ? 1 : 0);
                    dataGridView2[i, j].Value = freedom[i+1, j+1];
                }

            for (int i = 0; i < N + 2; i++) field[0, i] = field[N + 1, i] = field[i, 0] =
field[i, N + 1] = -1;
            for (int i = 0; i < N + 2; i++) freedom[0, i] = freedom[N + 1, i] =
freedom[i, 0] = freedom[i, N + 1] = -1;

            left = N * N;

            System.Diagnostics.Stopwatch sw = new System.Diagnostics.Stopwatch();
            sw.Start();
            Words = new List<List<Point>>();
            for (int z = 0; z < N*N; z++)
            {
                int x, y;
                x = rnd.Next(1, N + 1);
                y = rnd.Next(1, N + 1);
                int newL = rnd.Next(MinL, MaxL + 1);
                Words.Add(new List<Point>());
                if (rec(x, y, 1, newL)) break;
                Words.RemoveAt(Words.Count - 1);
            }
            sw.Stop();
            Show_Lists();
            Show_Field();
            Show_Words();

            if (Words.Count > 0)
            {
                textBox1.Text = "Построено за " + sw.Elapsed.TotalSeconds + "
сек.\r\n\r\n" + textBox1.Text;
            }
            else
                MessageBox.Show("Не удалось построить карту слов");

        }

        bool rec(int x, int y, int n, int L)
        {
            field[x, y] = n;
            Words[Words.Count - 1].Add(new Point(x-1, y-1));
            int backup = freedom[x, y];
            freedom[x, y] = -1;

            int dx = 0, dy = 0, zeros = 0;
            if (--freedom[x - 1, y] == 0) { dx = -1; zeros++; }
            if (--freedom[x + 1, y] == 0) { dx =  1; zeros++; }
            if (--freedom[x, y - 1] == 0) { dy = -1; zeros++; }
            if (--freedom[x, y + 1] == 0) { dy =  1; zeros++; }

            if (--left == 0)
                if (Words[Words.Count - 1].Count >= MinL && Words[Words.Count - 1].Count
<= MaxL) return true;
                else zeros = 3;

            while (zeros < 2)
            {
                if (zeros == 1)
                    if (rec(x + dx, y + dy, n, 1)) return true; else break;

                if (L > 1)
```

```
                {
                    int[] dd = new int[] { 1, 0, -1, 0 };
                    int j = 0, k = rnd.Next(4);
                    while (field[x + dd[(k + j) % 4], y + dd[(k + j + 1) % 4]] != 0 && j
< 4) j++;
                    if (j < 4) if (rec(x + dd[(k + j) % 4], y + dd[(k + j + 1) % 4], n, L
- 1)) return true; else break;
                }

                if (Words[Words.Count - 1].Count < MinL || Words[Words.Count - 1].Count >
MaxL) break;

                if (left < MinL) break;

                for (int newx = 1; newx < N + 1; newx++)
                    for (int newy = 1; newy < N + 1; newy++)
                        if ( field[newx, newy] == 0 &&
rnd.NextDouble()<=(double)MinL/left/2.0 )
                        {
                            int newL = rnd.Next(MinL, MaxL + 1);
                            Words.Add(new List<Point>());
                            if (rec(newx, newy, n + 1, newL)) return true;
                            Words.RemoveAt(Words.Count - 1);
                        }
                break;
            }

            field[x, y] = 0;
            Words[Words.Count - 1].RemoveAt(Words[Words.Count - 1].Count-1);
            left++;
            freedom[x, y] = backup;

            freedom[x - 1, y]++;
            freedom[x + 1, y]++;
            freedom[x, y - 1]++;
            freedom[x, y + 1]++;

            return false;
        }

        void Show_Lists()
        {
            textBox1.Clear();

        }

        void Show_Field()
        {
            for (int x = 0; x < N; x++)
                for (int y = 0; y < N; y++)
                    dataGridView1[x, y].Value = field[x + 1, y + 1];
        }

        void Show_Words()
        {
            Color[] colors = new Color[] { Color.Khaki, Color.LightYellow,
Color.LightBlue, Color.PeachPuff, Color.LightCyan,
                Color.LightGray, Color.Lavender, Color.Thistle, Color.MistyRose,
Color.BurlyWood, Color.Gainsboro };

            for (int i = 0; i < Words.Count; i++)
            {
                Color c = colors[i % colors.Length];
```

```
                List<string> Spisok = new List<string>();

                int dlina = Words[i].Count;
                foreach (string w in Dict)
                {
                    if (w.Length == dlina)
                        Spisok.Add(w);
                }

                if (Spisok.Count == 0)
                {
                    textBox1.Text = "Пробел в филворде((";
                    continue;
                }

                int r = rnd.Next(Spisok.Count);
                string z = Spisok[r];

                int j = 0;
                foreach (Point p in Words[i])
                {
                    dataGridView1[p.X, p.Y].Value = z[j++];
                    dataGridView1[p.X, p.Y].Style.BackColor = c;
                }
            }
        }

    }
}
```