

**COLEGIUL NAȚIONAL DE INFORMATICĂ  
„GRIGORE MOISIL” BRAȘOV**

**LUCRARE PENTRU ATESTAREA  
COMPETENȚELOR PROFESIONALE**

**SUDOKU**

**Elev**

**Bucur Robert-Adrian**

**Profesori îndrumători**

**Șerban Manuela-Corina**

**Burcuță Maria-Simona**

**Clasa**

**a XII-a B**

**Mai 2018**

## Cuprins

Motivarea alegerii temei .....	3
Prezentarea aplicației .....	4
Detalii de implementare .....	7
Posibilități de dezvoltare .....	9
Resurse hardware și software necesare .....	9
Bibliografie .....	10

## Motivarea alegerii temei

Am ales să realizez aplicația de față, deoarece dintotdeauna mi-a plăcut să îmi pun mintea la încercare prin diverse jocuri. Sudoku este unul dintre ele.

Este foarte interesant cum mintea umană poate procesa informațiile, de aceea consider că este extrem de util și benefic să ne antrenăm creierul prin tot felul de joculețe de memorie, logică și atenție distributivă. Gândirea logică este foarte prețioasă în ziua de azi în toate domeniile, mai ales în informatică. Realizarea unei aplicații presupune o privire de ansamblu, generală, dar mai ales abstractă, de aceea alegerea mea în privința temei aplicației a fost una destul de ușoară.

Sudoku este un joc în formă de grilă inventat în 1979 și inspirat de pătratul latin și de problema celor 36 ofițeri a lui Leonard Euler. Scopul jocului este de a umple această grilă cu cifrele de la 1 la 9 respectând anumite condiții, cu unele cifre fiind de la început dispuse în grilă. Grila jocului este un pătrat de nouă pe nouă căsuțe, subdivizat în tot atâtea pătrate identice, numite regiuni. Regula jocului este simplă: fiecare rând, coloană sau regiune nu trebuie să conțină decât o dată cifrele de la unu la nouă. Formulată altfel, fiecare ansamblu trebuie să conțină cifrele de la unu la nouă o singură dată.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

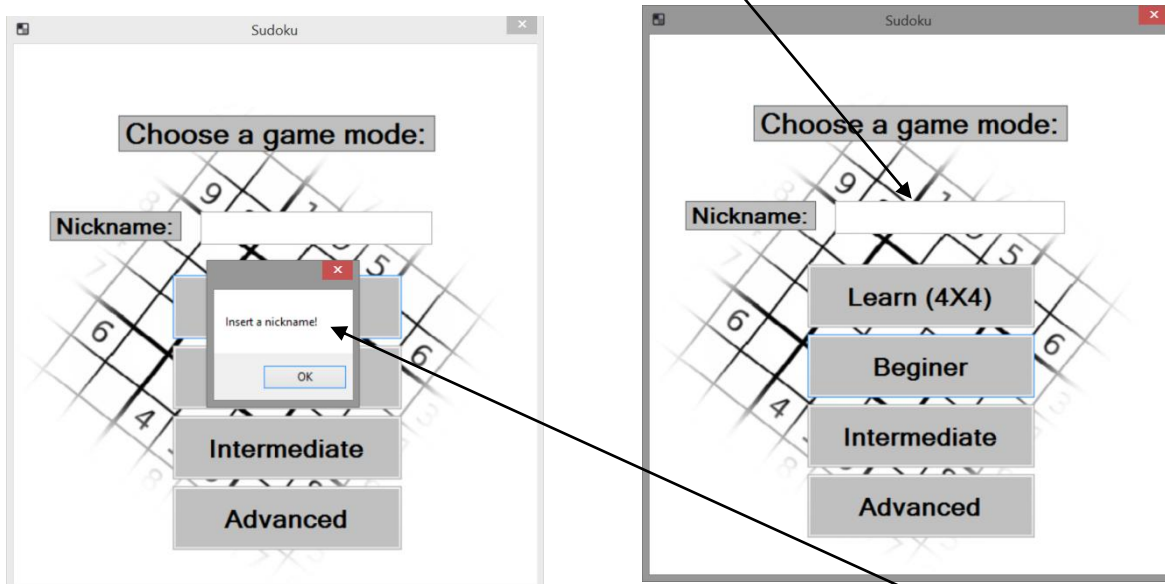
Realizarea aplicației Sudoku permite experimentarea unei astfel de gândiri logice, generale, deoarece trebuie luate în considerare multe aspecte, ca de exemplu "Dacă utilizatorul introduce litere în loc de cifre?" sau "Dacă utilizatorul introduce mai multe caractere nu numai unul?". Pentru a evita erorile logice, legate de introducerea unor parametri greșiți, trebuie limitat accesul utilizatorului la anumite lucruri prin setările din 'background'. Blocându-i accesul la căsuțele în care sunt introduse cifrele ajutoare pentru completarea tablei, spre exemplu, se realizează o reducere a erorilor în verificarea corectitudinii aranjamentului de cifre pe tablă.

## Prezentarea aplicației

Aplicația este concepută astfel încât oricine să poată juca fără probleme binecunoscutul joc.

Pentru utilizatorii care nu știu "cu ce se mănâncă" și nu au mai jucat până acum, am creat o secțiune separată pe care am numit-o 'Learn'. Este o varianta simplificată de la grila de 9x9 la o grilă de 4x4, astfel fiind ușor de învățat regulile.

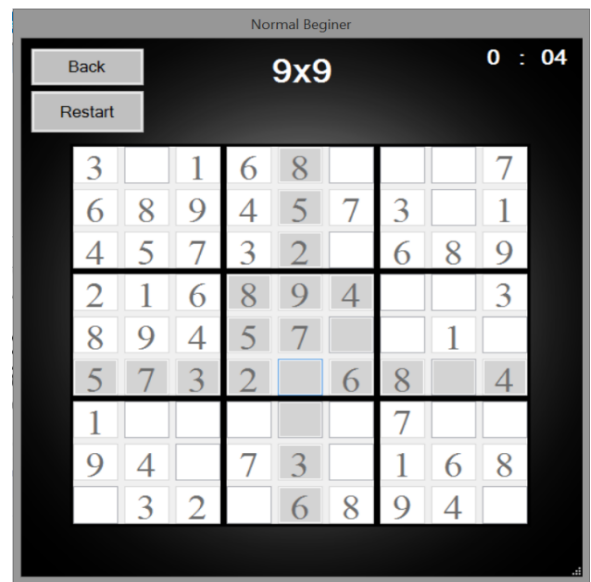
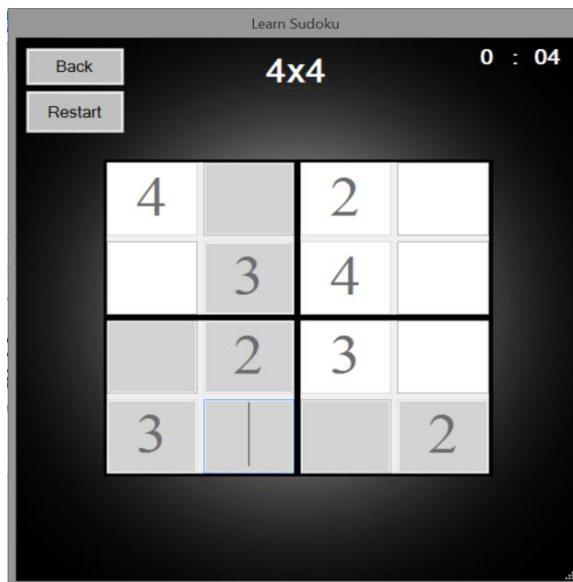
Dar să începem cu începutul. La rularea aplicației ne întâmpină o pagină de start cu patru butoane și un câmp de completare a poreclei. Pentru a putea continua și a juca în oricare dintre nivele trebuie introdusă o poreclă aici.



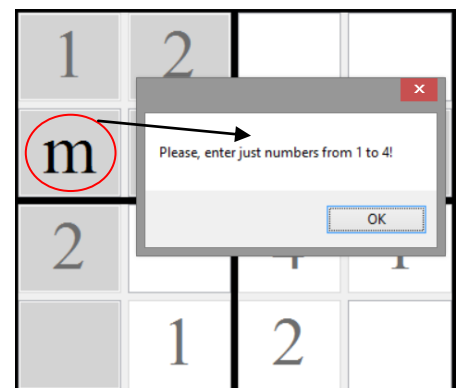
Dacă nu este introdus un text în câmpul specificat mai sus ne va apărea un mesaj prin care ne este cerut să completăm câmpul, deoarece scorul obținut în oricare dintre nivele este înregistrat într-o bază de date și este necesară interogarea acesteia pentru a actualiza scorul sau pentru a-l insera pentru un utilizator nou. Scorul este dat de timpul în care acesta reușește să completeze tabla cu toate cifrele puse în ordine corectă. În cazul utilizatorilor care au mai folosit aplicația și au fost înregistrați în baza de date, scorul va fi actualizat doar dacă noul scor obținut ulterior este mai bun decât cel înregistrat.

Aplicația dispune de patru nivele de dificultate:

- **Learn** - este cel mai simplu mod de a învăța sudoku; în loc de o grilă de 9x9, am realizat o tablă mai simplă de 4x4 cu un ajutor de a înțelege în ce mod trebuie completate căsuțele prin colorarea liniei, coloanei și a regiunii din care face parte căsuța selectată;
- **Beginner** - modul de începător reprezintă o tabla de joc normală de 9x9, cu un număr mai mare de soluții, cifre afișate de la început pentru ajutor, și prin colorarea liniei, coloanei și regiunii exact ca la modul 'Learn';
- **Intermediate** - este un nivel mediu ca dificultate, care nu prezintă ajutorul prin colorarea căsuțelor;
- **Advanced** - cel mai greu nivel, nu prezintă ajutorul prin colorarea căsuțelor, iar numărul de cifre afișate este minim.



Câștigarea jocului presupune completarea cu cifre corespunzătoare a tablei. Cifrele completate trebuie să fie poziționate corespunzător, astfel încât să nu se repete pe linie, coloană sau într-o regiune. În cazul în care sunt introduse alte caractere decât cele corespunzătoare, pe ecran va apărea un mesaj care va indica faptul că trebuie introduse doar cifre de la 1 la 9, în cazul nivelului beginner, intermediate și advanced, și respectiv cifre de la 1 la 4 în cazul nivelului learn. După ce jucătorul apasă OK, căsuța în care a introdus caracterul necorespunzător va fi resetată.



La câștigarea jocului, pe ecran va apărea un mesaj de felicitare prin care este specificat faptul ca utilizatorul a câștigat, iar în cazul în care ordinea cifrelor nu corespunde sau sunt repetate caractere va apărea un mesaj care sugerează jucătorului să reîncece fără a i se reseta timpul sau chiar tabla.



show

Restart Exit

	Id	Nickname	Mins	Sec	Date	Level
▶	1	marius	0	8	4/13/2018 1:27 PM	learn
	3	raul	0	10	4/13/2018 1:27 PM	learn
	6	mao	0	13	4/13/2018 1:27 PM	learn
	13	mary	0	15	4/13/2018 1:27 PM	learn
	5	gicu	0	15	4/13/2018 1:27 PM	learn
	16	rusu	0	15	4/13/2018 1:27 PM	learn
	19	romeo	0	15	4/13/2018 1:27 PM	learn
	14	florin	0	16	4/13/2018 1:27 PM	learn
	21	mircea	0	16	4/22/2018 8:32 PM	learn
	18	mirica	0	18	4/13/2018 1:27 PM	learn
*						

Apăsând butonul 'OK' sau 'X' aplicația va afișa utilizatorului top 10 cei mai buni jucători ai nivelului respectiv înregistrați cu timpul obținut și data la care au obținut acest timp.

Butonul 'Restart' al fiecărui nivel de joc, va reseta timpul si tabla de sudoku prin ștergerea tuturor caracterelor introduse automat și de către utilizator și re completarea acestuia cu numărul necesar de cifre plasate în diferite poziții.

## Detalii de implementare

Codul sursă l-am scris în limbajul de programare al celor de la Microsoft, și anume C#, iar pentru conexiunea la baza de date și interogarea acestora am folosit SQL.



Pentru realizarea tablei 'șablon' am creat o clasă C# care generează o matrice standard, folosind numere de la 1 la n (n având valoarea 4, în cazul nivelului 'learn', sau 9 pt celelalte nivele), și o actualizează mutând aleatoriu cifrele generate pe anumite poziții fixate la început pe alte poziții astfel încât în fond să fie tot o tablă rezolvată de sudoku.

```
class SudokuClass
{
    public int[,] numbers = new int[9, 9];
    private int N;
    public SudokuClass(int n)
    {
        N = (int)System.Math.Sqrt(n);
        Generate(n);
        Update(n);
    }
    private void Generate(int n)
    {
        for (int i = 0; i < n; i++)
            for (int j = 0; j < n; j++)
                numbers[i, j] = (i * N + i / N + j) % n + 1;
    }
    private void ChangeCells(int v1, int v2, int n)
    {
        int x1 = 0, y1 = 0, x2 = 0, y2 = 0;
        for (int i = 0; i < n; i += N)
            for (int j = 0; j < n; j += N)
            {
                for (int k = 0; k < N; k++)
                    for (int l = 0; l < N; l++)
                    {
                        if (numbers[i + k, j + l] == v1)
                        {
                            x1 = i + k;
                            y1 = j + l;
                        }
                        if (numbers[i + k, j + l] == v2)
                        {
                            x2 = i + k;
                            y2 = j + l;
                        }
                    }
                numbers[x1, y1] = v2;
                numbers[x2, y2] = v1;
            }
    }
    private void Update(int n)
    {
        for (int i = 0; i < n+1; i++)
        {
            var rnd1 = new Random(Guid.NewGuid().GetHashCode());
            var rnd2 = new Random(Guid.NewGuid().GetHashCode());
            ChangeCells(rnd1.Next(1, n + 1), rnd2.Next(1, n + 1), n);
        }
    }
}
```

Constructorul are parametrul n prin care primește nr. de cifre al matricei care trebuie generată; acest nr. este 4 sau 9 în funcție de nivel.

Funcția 'Generate' creează matricea standard, care va fi memorată și verificată prin corelație cu matricea pe care o completează utilizatorul.

Prin parametrii funcției 'ChangeCells' sunt transmise valorile alese aleatoriu de variabilele de tip Random din funcția 'Update' de unde este de altfel și apelat subprogramul de schimbare a celulelor, mai exact a cifrelor din celule.

```

private void beginner_Load(object sender, EventArgs e)
{
    Font font = new Font("Times New Roman", 28.0f);
    for (int i = 0; i < table.RowCount; i++)
    {
        for (int j = 0; j < table.ColumnCount; j++)
        {
            matrix[i, j] = new TextBox();
            matrix[i, j].Parent = table;
            matrix[i, j].Location = new Point(i * 80, j * 80);
            matrix[i, j].AutoSize = false;
            matrix[i, j].Width = 80;
            matrix[i, j].Height = 80;
            matrix[i, j].Visible = true;
            matrix[i, j].MaxLength = 1;
            matrix[i, j].TextAlign = HorizontalAlignment.Center;
            matrix[i, j].Font = font;
            matrix[i, j].BackColor = Color.White;
            matrix[i, j].TextChanged += new EventHandler(matrix_TextChanged);
            matrix[i, j].Click += new EventHandler(matrix_Click);
        }
    }
    AddNumbers();
    timer.Start();
}

private SudokuClass b = new SudokuClass(9);
private void AddNumbers()
{
    int ok = 0;
    while (ok < 56)
    {
        var rnd1 = new Random(Guid.NewGuid().GetHashCode());
        var rnd2 = new Random(Guid.NewGuid().GetHashCode());
        int x1 = rnd1.Next(0, 9);
        int x2 = rnd2.Next(0, 9);
        if (matrix[x1, x2].Text == "")
        {
            matrix[x1, x2].Text = b.numbers[x1, x2].ToString();
            matrix[x1, x2].Enabled = false;
            ok++;
        }
    }
}

private void verify(object sender, EventArgs e)
{
    int ok = 1;
    for (int i = 0; i < table.RowCount; i++)
        for (int j = 0; j < table.ColumnCount; j++)
            if (matrix[i, j].Text == "")
                return;
            else
                if (matrix[i, j].Text == b.numbers[i, j].ToString())
                    ok = 1;
                else
                    ok = 0;

    if (ok == 1)
    {
        timer.Stop();
        MessageBox.Show("Congratulations! You won!");
        this.Hide();
        if (yes_no == 0)
        {
            cmd = new SqlCommand("SELECT * FROM board", con);
            SqlDataReader reader = cmd.ExecuteReader();
            List<string> l1 = new List<string>();
            while (reader.Read())
            {
                l1.Add(reader.GetString(1));
            }
            reader.Close();
            string b = @"INSERT INTO board(Id,Nickname,Mins,Sec,Date,Level) VALUES(" + l1.Count + ", '" + nickname + "', '" + mins.ToString() + "', '" + sec.ToString() + "', GETDATE(), '" + lvl + "')";
            reader.Close();
            cmd = new SqlCommand(b, con);
            cmd.ExecuteNonQuery();
        }
        else
        {
            string b = @"UPDATE board SET Mins = '" + mins.ToString() + "', Sec = '" + sec + "', Date = GETDATE()";
            cmd = new SqlCommand(b, con);
            cmd.ExecuteNonQuery();
        }
        this.Hide();
        show form = new show(lvl);
        form.ShowDialog();
        restart_Click(sender, e);
    }
    else
        MessageBox.Show("Retry!");
}

```

În momentul în care o fereastră a unui nivel este încărcată, este generată o matrice de **TextBox**-uri cărora le sunt setate anumite proprietăți din cod, cum ar fi poziția, lungimea maximă a textului care poate fi introdus, fontul și acțiunile.

Funcția 'AddNumbers' adaugă un anumit număr de cifre din matricea șablon generată cu ajutorul clasei **SudokuClass** și transmite ferestrelor nivelelor prin intermediul obiectului 'b'.

Funcția 'verify' verifică dacă tabla completată de jucător coincide cu cea șablon, iar în cazul în care matricea corespunde se va afișa mesajul de felicitare și se va actualiza baza de date: dacă utilizatorul este nou (se realizează o interogare a bazei de date după poreclă) va fi inserat cu scorul obținut, data și toate datele necesare, iar dacă este un utilizator înregistrat scorul îi va fi actualizat.



## **Posibilități de dezvoltare**

1. Punerea unui sunet de fundal și a unui sunet la completare a tablei.
2. Adăugarea unei înregistrări prin intermediul unei parole și a unui nume de utilizator.
3. Stilizarea afișării topului utilizatorilor prin selectarea nivelului.
4. Interogarea bazei de date prin căutarea unui jucător după numele de utilizator.
5. Adăugarea unei ferestre de informare legată de regulile jocului și a unei ferestre de sfaturi și sugestii despre cum se rezolvă simplu și repede o tablă de sudoku.

## **Resurse hardware și software necesare**

- 1.6 GHz sau mai mare
- 1 GB RAM sau mai mare
- 1 GB ROM sau mai mare
- 5,400 RPM HDD sau mai mare
- Placă video care rulează la o rezoluție de  $1024 \times 768$  sau mai mare

Aplicația Sudoku a fost realizată pe un ASUS F550V cu procesor Intel Core i7 de 3.5 GHz, memorie RAM de 8 GB, memorie HDD de 1 TB .

Programul Microsoft Visual Studio 2012 versiunea 11.050727.42 de x32Bit a făcut posibilă implementarea codului sursă, a interfeței și a bazei de date prin folosirea unor biblioteci speciale cu funcții și metode necesare compilării și executării codului scris în SQL.

## **Bibliografie**

- <https://www.w3schools.com/>
- <https://stackoverflow.com/>
- <https://msdn.microsoft.com/>
- <https://www.codecademy.com/>
- <https://www.tutorialspoint.com/csharp/>
- <http://csharp.net-tutorials.com/>
- <http://www.learncs.org/>
- <https://www.sololearn.com/Course/CSharp/>
- <https://www.youtube.com/watch?v=HwA-tNvU3hw&feature=youtu.be>
- <https://www.youtube.com/watch?v=lisiwUZJXqQ>
- <https://www.youtube.com/watch?v=GcFJjpMFJvI>