**Colegiul Național „Mihail Kogalniceanu”**

**Galați**

**Lucrare pentru atestarea competențelor profesionale la disciplina informatică**

***Algoritmi de sortare***

***Îndrumător:* *Elev:***

**Prof. Neagu Violeta Boc Mario-Adrian**

***Profil*: Real *Clasa*: a XII-a F**

***Specializare*: Matematică-Informatică**

**2023**

**CUPRINS**

**Tema proiectului…………………………………………1**

**Introducere……………………………………………….2**

**Prezentarea si utilizarea aplicației………………………3**

**Cod sursă………………………………………………….9**

**Bibliografie……………………………………………….24**

**TEMA PROIECTULUI**

Să se realizeze o aplicație software implementată în limbajul C#, prin intermediul mediului de programare Microsoft Visual Studio, folosindu-se librăria grafică Windows Forms, care sa integreze algoritmi de sortare.

**INTRODUCERE**

Aplicația realizată are în vedere reprezentarea grafică a 3 metode de sortare : Selection Sort, Bubble Sort și Insertion Sort. Motivul pentru care am ales această temă este redat de valoarea educațională care poate fi extrasă din vizualizarea procesului de ordonare în timp real. Elevii vor capăta o înțelegere mai bună a metodelor de sortare, aplicația având în componență și informații utile despre fiecare algoritm în parte.

Proiectul a fost realizat folosind limbajul de programare Visual C#, implementat în mediul de programare Microsoft Visual Studio, utilizând librăria grafică Windows Forms.

C# este un limbaj de programare orientat pe obiecte, conceput de Microsoft la sfârșitul anilor 90. A fost conceput ca un concurent pentru limbajul Java. Ca și acesta, C# este un derivat al limbajului de programare C++. C# simplifică mult scrierea de programe pentru sistemul de operare Windows, iOS, Android etc. Este un limbaj de programare cross-platform.

Microsoft Visual Studio include un set complet de instrumente de dezvoltare pentru generarea de aplicații ASP.NET, Servicii Web XML, aplicații desktop și aplicații mobile. Visual Basic, Visual C++, Visual C# și Visual J# toate folosesc același mediu de dezvoltare integrat (IDE) care le permite partajarea instrumentelor și facilitează crearea de soluții folosind mai multe limbaje de programare. Windows Forms este numele interfeței grafice care este inclusă în cadrul .NET , oferind acces prin intermediul codului gestionat la API-ul Windows . Odată cu lansarea .NET 3.0 în 2006, Windows Forms este programat pentru înlocuire cu un alt sistem GUI, Windows Presentation Foundation . Pentru dezvoltatori , este mai puțin complex de utilizat decât MFC (bazat pe C ++ ), dar nu oferă o paradigmă comparabilă cu Model-View-Controller . Au fost create biblioteci software terță parte pentru a adăuga această funcționalitate. Cel mai utilizat este blocul de aplicații pentru procesarea interfeței utilizator .

**PREZENTAREA ȘI UTILIZAREA APLICAȚIEI**

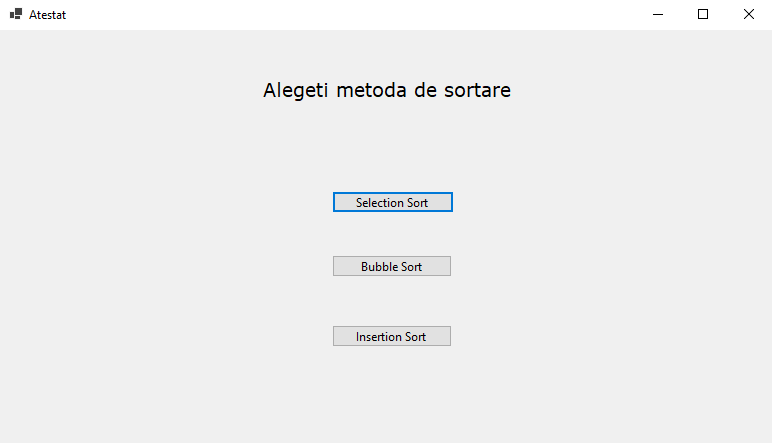
Imediat dupa accesarea aplicației, utilizatorul este întâmpinat de meniul principal.



Prima etapă constă în completarea celor 2 câmpuri cu marimea, respectiv valorile vectorului care trebuie sa fie ordonate. Valorile trebuie sa fie pozitive, cele negative fiind convertite în valori pozitive.

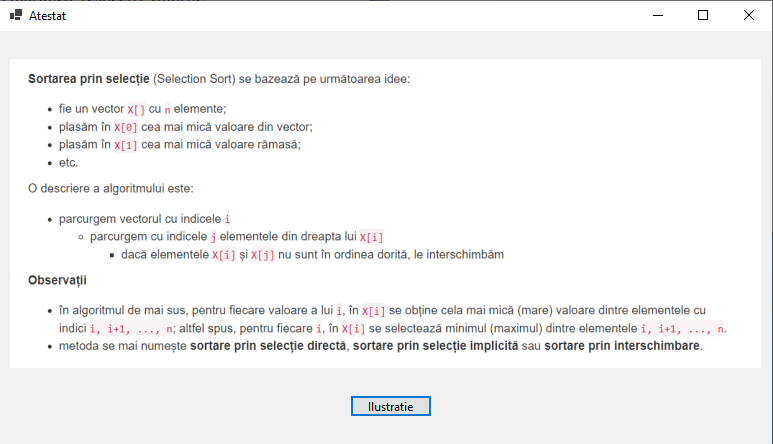


A doua etapă este redată de alegerea metodei de sortare, dupa apăsarea butonului “Alege metoda de sortare” din meniul principal. În noua fereastră deschisă, utilizatorul trebuie sa aleagă dintre opțiunile de ordonare.

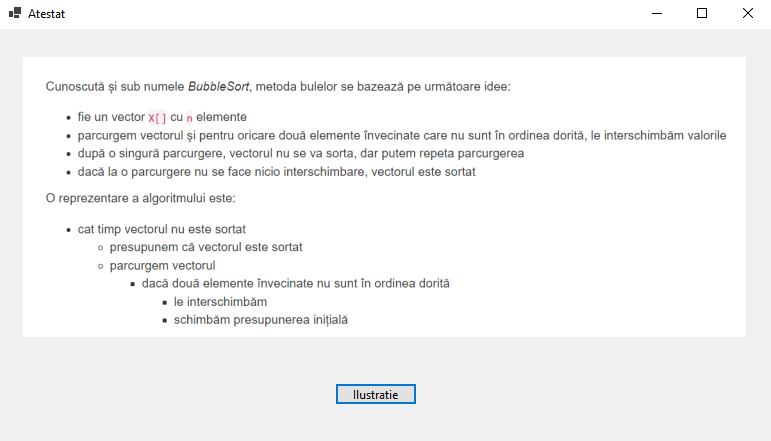


În funcție de alegere, utilizatorul va fi întâmpinat de o nouă fereastră grafică, ce informează utilizatorul cu privire la algoritmul de sortare ales.

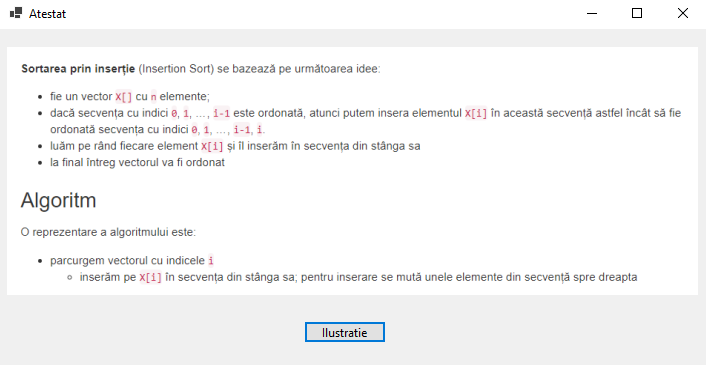
Pentru fereastra “Selection Sort”:



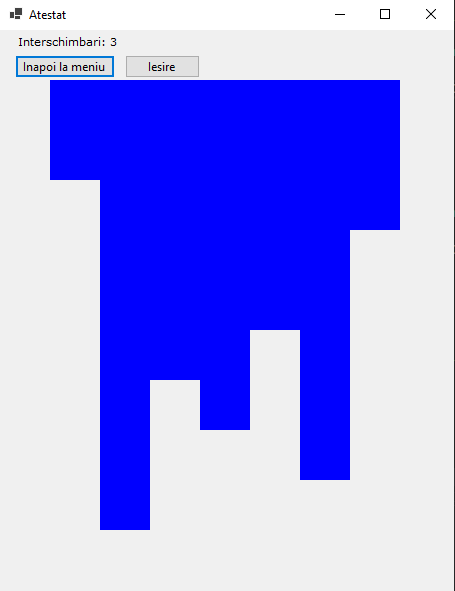
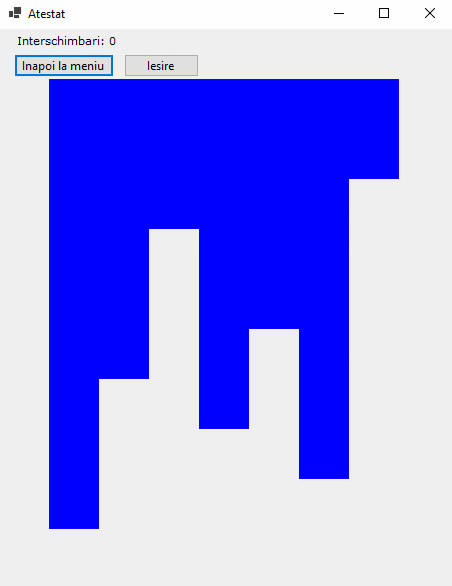
Pentru fereastra “Bubble Sort”:

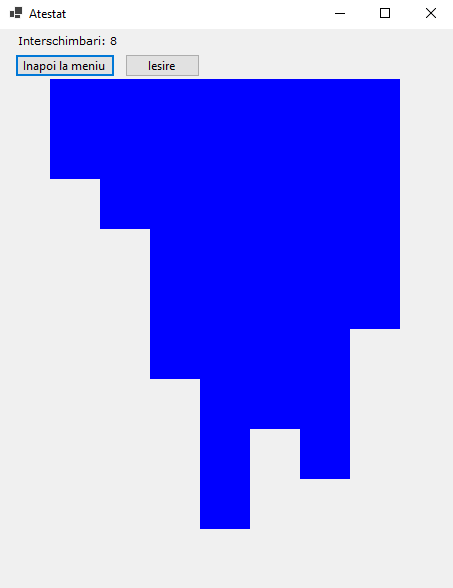
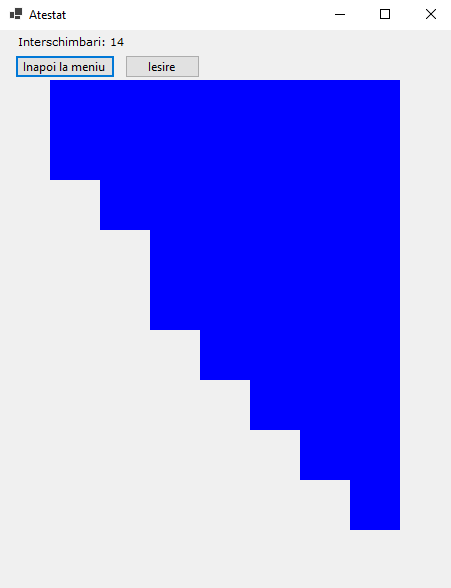


Pentru fereastra “Insertion Sort”:

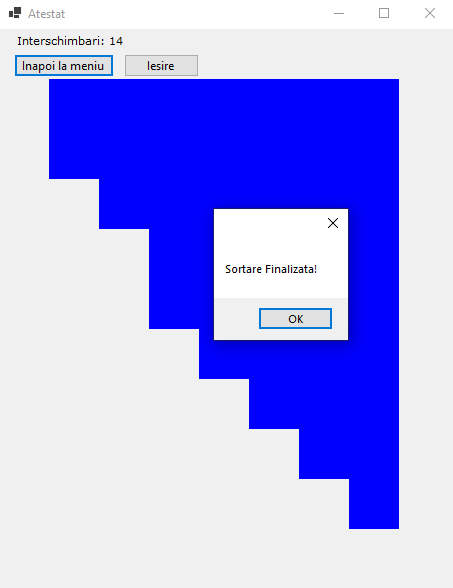


A treia etapă are la bază vizualizarea ilustrării grafice a vectorului introdus, dupa modelul algoritmului selectat, în timp real.

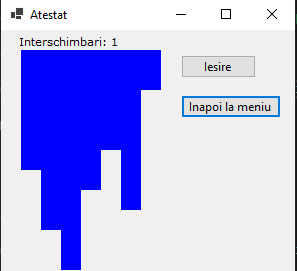


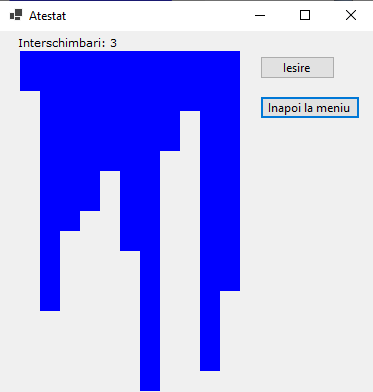
 

La finalul operațiunii de sortare, utilizatorul este întâmpinat de un mesaj ce anunță terminarea procesului de ordonare.



Observație: Din moment ce fereastra responsabilă pentru ilustrarea grafică este dimensionată dupa valorile introduse de utilizator, fereastra poate sa devină mai mare decât rezoluția monitorului datorită valorilor, motiv pentru care marimea se actualizează după introducere, formând o fereastră de dimeniuni mai mici, pentru a observa în același timp și valorile mai mari decât 10.





**CODUL SURSĂ**

Meniul principal este format din Form-ul intitulat “Form1”, ce se ocupă de prelucrarea datelor oferite de utilizator, după apăsarea butonului “Alege metoda de sortare”.

private void button\_alegeMetoda\_Click(object sender, EventArgs e)

{

string[] cuv;

//stergem spatiile care nu sunt necesare din input-ul textbox-urilor---------

string s1 = textbox\_marimeVector.Text.Trim();

textbox\_marimeVector.Text = s1;

string s2 = textbox\_valoriVector.Text.Trim();

textbox\_valoriVector.Text = s2;

//---------------------------------------------------------------------------

//verificam daca utilizatorul a completat datele necesare--------------------

if (string.IsNullOrEmpty(textbox\_marimeVector.Text) || string.IsNullOrEmpty(textbox\_valoriVector.Text))

{

MessageBox.Show("Nu ai completat datele!");

}

else

{

//stocam inputul din textbox-uri in variabilele globale--------------

Variables.marimeVector = textbox\_marimeVector.Text;

Variables.n = Convert.ToInt32(Variables.marimeVector);

Variables.valoriVector = textbox\_valoriVector.Text;

//-------------------------------------------------------------------

//stocam valorile si maximul vectorului cu care vom lucra------------

cuv = Variables.valoriVector.Split(' ');

for (int i = 0; i < Variables.n; i++)

{

Variables.v[i] = Convert.ToInt32(cuv[i]);

if (Variables.v[i] < 0)

Variables.v[i] = Math.Abs(Variables.v[i]);

}

Variables.maxi = Variables.v.Max();

//--------------------------------------------------------------------

//setam marimile necesare in caz ca valorile sunt prea mari-----------

if (Variables.maxi > 10)

{

Variables.lat = 20;

}

//--------------------------------------------------------------------

//trecem in urmatoarea fereastra--------------------------------------

Form2 f2 = new Form2();

this.Hide();

f2.Show();

//--------------------------------------------------------------------

}

//---------------------------------------------------------------------------

}

În Form-ul inițial mai sunt prezente si funcții auxiliare care se ocupă cu închiderea programului și verificarea input-ului.

//daca utilizatorul inchide aplicatia, ne asiguram ca toate ferestrele se inchid-

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

Environment.Exit(0);

}

//-------------------------------------------------------------------------------

//ne asiguram ca sunt tastate doar numere in textbox-uri-------------------------

private void textbox\_marimeVector\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) && (e.KeyChar != ' '))

{

e.Handled = true;

}

}

private void textbox\_valoriVector\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) && (e.KeyChar != ' '))

{

e.Handled = true;

}

}

//-------------------------------------------------------------------------------

Meniul care se ocupă cu ilustrarea alegerilor, intitulat “Form2”, are la baza doar funcții ce fac legătura între celelalte meniuri și care se ocupă cu închiderea aplicației.

//schimbam ferestrele in functie de butonul apasat-------------------------------

private void button\_selectionSort\_Click(object sender, EventArgs e)

{

SelectionSortInfo ssi = new SelectionSortInfo();

this.Hide();

ssi.Show();

}

private void button\_bubbleSort\_Click(object sender, EventArgs e)

{

BubbleSortInfo bsi = new BubbleSortInfo();

this.Hide();

bsi.Show();

}

private void button\_quickSort\_Click(object sender, EventArgs e)

{

InsertionSortInfo qsi = new InsertionSortInfo();

this.Hide();

qsi.Show();

}

//-------------------------------------------------------------------------------

//daca utilizatorul inchide aplicatia, ne asiguram ca toate ferestrele se inchid-

private void Form2\_FormClosing(object sender, FormClosingEventArgs e)

{

Environment.Exit(0);

}

//-------------------------------------------------------------------------------

În cazul alegerii opțiunii “Selection Sort”, meniul cu informații inserat imediat după apăsarea butonului, intitulat “SelectionSortInfo”, este format doar dintr-o funcție ce face legătura cu meniul ilustrarii și o funcție care închide aplicația.

//trecem in urmatoarea fereastra-------------------------------------------------

private void button\_ilustratie\_Click(object sender, EventArgs e)

{

SelectionSortIlustratie s = new SelectionSortIlustratie();

s.Show();

this.Hide();

}

//-------------------------------------------------------------------------------

//ne asiguram ca inchidem toate ferestrele la inchiderea aplicatiei--------------

private void SelectionSortInfo\_FormClosing(object sender, FormClosingEventArgs e)

{

Environment.Exit(0);

}

//-------------------------------------------------------------------------------

Meniul care se ocupă cu afișarea ilustrării sortării vectorului după metoda Selection Sort este intitulat “SelectionSortIlustrație” și are la bază:

-variabilele locale folosite

//variabilele locale utilizate---------------------------------------------------

PictureBox[] pictureBoxes = new PictureBox[Variables.n];

public int i = 0;

public int j = 0;

public Point intermPoint;

public Size intermSize;

public int interschimbari = 0;

//-------------------------------------------------------------------------------

-funcții care creează și afișează vectorul

//crearea vectorului de imagini--------------------------------------------------

public void creeareVector()

{

for(int i = 0; i < Variables.n; i++)

{

pictureBoxes[i] = new PictureBox();

pictureBoxes[i].Name = "pbBox" + i;

pictureBoxes[i].Tag = "pbbox";

pictureBoxes[i].BackColor = Color.Blue;

Controls.Add(pictureBoxes[i]);

pictureBoxes[i].Visible = false;

}

}

//-------------------------------------------------------------------------------

//afisarea vectorului de imagini-------------------------------------------------

public void afisareVector()

{

for (int c = 0; c < Variables.n; c++)

{

pictureBoxes[c].Location = new Point(Variables.lat \* c + Variables.lat, Variables.lat);//setarea coordonatelor

pictureBoxes[c].Size = new System.Drawing.Size(Variables.lat, Variables.v[c] \* Variables.lat);//setarea marimii

pictureBoxes[c].Visible = true;

}

}

//-------------------------------------------------------------------------------

-funcția inițială care este chemată la încărcarea ferestrei și seteaza valorile inițiale

private void SelectionSortIlustratie\_Load(object sender, EventArgs e)

{

interschimbari = 0;

//deblocarea butoanelor-------------------------------------------

buttonExit.Visible = true;

buttonExit.Enabled = true;

buttonMeniu.Visible = true;

buttonMeniu.Visible = true;

//----------------------------------------------------------------

//deblocarea si setarea timer-ului--------------------------------

timerSelectionSort.Enabled = true;

timerSelectionSort.Interval = Variables.interval;

//----------------------------------------------------------------

i = 0;

j = i + 1;

//setarea initiala a coordonatelor butoanelor---------------------

buttonMeniu.Top = 25;

buttonMeniu.Left = 15;

buttonExit.Top = 25;

buttonExit.Left = 25 + buttonMeniu.Width;

//----------------------------------------------------------------

//setarea label-ului----------------------------------------------

labelInterschimbari.Text = "Interschimbari: 0";

labelInterschimbari.Top = 5;

labelInterschimbari.Left = 15;

//----------------------------------------------------------------

creeareVector();

afisareVector();

//actualizarea coordonatelor butoanelor in cazul valorilor mari----

if (Variables.lat != 50)

{

buttonExit.Location = new Point(pictureBoxes[Variables.n - 1].Left + pictureBoxes[Variables.n - 1].Width + 20,

25);

buttonMeniu.Location = new Point(pictureBoxes[Variables.n - 1].Left + pictureBoxes[Variables.n - 1].Width + 20,

buttonExit.Top + 40);

}

//-------------------------------------------------------------------

//setarea dimensiunii ferestrei dupa valorile din vector-------------

this.Size = new Size(Math.Max(Variables.lat \* (Variables.n + 2) + 20, Math.Max(buttonMeniu.Left + buttonMeniu.Width + 30, 250)),

Variables.lat \* (Variables.maxi + 3));

//-------------------------------------------------------------------

}

-funcția care se ocupă cu sortarea vectorului de imagini și a celui de valori

//sortarea valorilor din vectorul de imagini si din cel de valori----------------

public void sortareVector()

{

if (Variables.v[i] > Variables.v[j])

{

//actualizarea label-ului si a numarului de interschimbari--------

interschimbari++;

labelInterschimbari.Text = "Interschimbari: " + interschimbari;

//----------------------------------------------------------------

//interschimbarea imaginilor--------------------------------------

intermPoint = pictureBoxes[i].Location;

intermSize = pictureBoxes[i].Size;

pictureBoxes[i].Location = pictureBoxes[j].Location;

pictureBoxes[i].Size = pictureBoxes[j].Size;

pictureBoxes[j].Location = intermPoint;

pictureBoxes[j].Size = intermSize;

//----------------------------------------------------------------

//interschimbarea valorilor---------------------------------------

int aux = Variables.v[i];

Variables.v[i] = Variables.v[j];

Variables.v[j] = aux;

//----------------------------------------------------------------

}

}

//-------------------------------------------------------------------------------

-funcția timer-ului, care se execută odată la fiecare jumătate de secundă

private void timerSelectionSort\_Tick(object sender, EventArgs e)

{

//sortarea vectorului la fiecare pas prin timer-----------------------

sortareVector();

afisareVector();

//--------------------------------------------------------------------

//parcurgerea vectorului dupa algoritm--------------------------------

if(i < Variables.n - 1 - 1)

{

if (j < Variables.n - 1)

{

j++;

}

else

{

i++;

j = i + 1;

}

}

//--------------------------------------------------------------------

else

{

timerSelectionSort.Enabled = false;//incheierea sortarii

MessageBox.Show("Sortare Finalizata!");

}

}

-restul funcțiilor auxiliare care se ocupă cu legătura dintre fereastră și alte meniuri sau cu închiderea aplicației

//trecerea in fereastra initiala-------------------------------------------------

private void buttonMeniu\_Click(object sender, EventArgs e)

{

Form1 f1 = new Form1();

f1.Show();

this.Dispose();

}

//-------------------------------------------------------------------------------

//inchiderea tuturor ferestrelor la iesirea din aplicatie------------------------

private void buttonExit\_Click(object sender, EventArgs e)

{

Environment.Exit(0);

}

private void SelectionSortIlustratie\_FormClosing(object sender, FormClosingEventArgs e)

{

Environment.Exit(0);

}

//-------------------------------------------------------------------------------

În cazul alegerii opțiunii “Bubble Sort”, meniul cu informații inserat imediat după apăsarea butonului, intitulat “BubbleSortInfo”, este format doar dintr-o funcție ce face legătura cu meniul ilustrarii și o funcție care închide aplicația.

//trecem in urmatoarea fereastra-------------------------------------------------

private void button\_ilustratie\_Click(object sender, EventArgs e)

{

this.Hide();

BubbleSortIlustratie bbsi = new BubbleSortIlustratie();

bbsi.Show();

}

//-------------------------------------------------------------------------------

//ne asiguram ca toate ferestrele sunt inchise la inchiderea aplicatiei----------

private void BubbleSortInfo\_FormClosing(object sender, FormClosingEventArgs e)

{

Environment.Exit(0);

}

//-------------------------------------------------------------------------------

Meniul care se ocupă cu afișarea ilustrării sortării vectorului după metoda Bubble Sort este intitulat “BubbleSortIlustrație” și are la bază:

-variabilele locale folosite

//variabilele locale folosite----------------------------------------------------

PictureBox[] pictureBoxes = new PictureBox[100];

public int i = 0;

public int j = 0;

public Point intermPoint;

public Size intermSize;

public int interschimbari = 0;

//-------------------------------------------------------------------------------

-funcții care creează și afișează vectorul

//crearea vectorului de imagini--------------------------------------------------

public void creeareVector()

{

int i;

for (i = 0; i < Variables.n; i++)

{

pictureBoxes[i] = new PictureBox();

pictureBoxes[i].Name = "pbBox" + i;

pictureBoxes[i].Tag = "pbbox";

pictureBoxes[i].BackColor = Color.Blue;

Controls.Add(pictureBoxes[i]);

pictureBoxes[i].Visible = false;

}

}

//-------------------------------------------------------------------------------

//afisarea vectorului de imagini-------------------------------------------------

public void afisareVector()

{

for (int c = 0; c < Variables.n; c++)

{

pictureBoxes[c].Location = new Point(Variables.lat \* c + Variables.lat, Variables.lat);//pozitie

pictureBoxes[c].Size = new System.Drawing.Size(Variables.lat, Variables.v[c] \* Variables.lat);//marime

pictureBoxes[c].Visible = true;

}

}

//-------------------------------------------------------------------------------

-funcția inițială care este chemată la încărcarea ferestrei și seteaza valorile inițiale

private void BubbleSortIlustratie\_Load(object sender, EventArgs e)

{

interschimbari = 0;

//deblocarea butoanelor---------------------------------------------------

buttonExit.Visible = true;

buttonExit.Enabled = true;

buttonMeniu.Visible = true;

buttonMeniu.Visible = true;

//------------------------------------------------------------------------

//activarea si setarea timerului------------------------------------------

timerBubbleSort.Enabled = true;

timerBubbleSort.Interval = Variables.interval;

//------------------------------------------------------------------------

i = 0;

j = 0;

//setarea initiala a coordonatelor butoanelor-----------------------------

buttonMeniu.Top = 25;

buttonMeniu.Left = 15;

buttonExit.Top = 25;

buttonExit.Left = 25 + buttonMeniu.Width;

//------------------------------------------------------------------------

//setarea coordonatelor label-ului----------------------------------------

labelInterschimbari.Text = "Interschimbari: 0";

labelInterschimbari.Top = 5;

labelInterschimbari.Left = 15;

//------------------------------------------------------------------------

//crearea initiala a vectorului de imagini-------------------------------

creeareVector();

afisareVector();

//------------------------------------------------------------------------

//actualizarea coordonatelor butoanelor daca valorile sunt prea mari------

if (Variables.lat != 50)

{

buttonExit.Location = new Point(pictureBoxes[Variables.n - 1].Left + pictureBoxes[Variables.n - 1].Width + 20,

25);

buttonMeniu.Location = new Point(pictureBoxes[Variables.n - 1].Left + pictureBoxes[Variables.n - 1].Width + 20,

buttonExit.Top + 40);

}

//-------------------------------------------------------------------------

//setarea dimensiunilor ferestrei------------------------------------------

this.Size = new Size(Math.Max(Variables.lat \* (Variables.n + 2) + 20, Math.Max(buttonMeniu.Left + buttonMeniu.Width + 30, 250)),

Variables.lat \* (Variables.maxi + 3));

//-------------------------------------------------------------------------

}

-funcția care se ocupă cu sortarea vectorului de imagini și a celui de valori

//sortarea vectorului de valori si a vectorului de imagini-----------------------

public void sortareVector()

{

if (Variables.v[j] > Variables.v[j + 1])

{

//actualizarea numarului de interschimbari----------------------------

interschimbari++;

labelInterschimbari.Text = "Interschimbari: " + interschimbari;

//--------------------------------------------------------------------

//interschimbarea imaginilor------------------------------------------

intermPoint = pictureBoxes[j].Location;

intermSize = pictureBoxes[j].Size;

pictureBoxes[j].Location = pictureBoxes[j + 1].Location;

pictureBoxes[j].Size = pictureBoxes[j + 1].Size;

pictureBoxes[j + 1].Location = intermPoint;

pictureBoxes[j + 1].Size = intermSize;

//--------------------------------------------------------------------

//interschimbarea valorilor-------------------------------------------

int aux = Variables.v[j];

Variables.v[j] = Variables.v[j + 1];

Variables.v[j + 1] = aux;

//--------------------------------------------------------------------

}

}

//-------------------------------------------------------------------------------

-funcția timer-ului, care se execută odată la fiecare jumătate de secundă

private void timerBubbleSort\_Tick(object sender, EventArgs e)

{

//sortarea vectorului la fiecare pas---------------------------------------

sortareVector();

afisareVector();

//-------------------------------------------------------------------------

//parcurgerea vectorului dupa algoritm-------------------------------------

if (i < Variables.n - 1)

{

if (j < Variables.n - i - 1)

{

j++;

if (j == Variables.n - i - 1)

j = 0;

}

else

{

i++;

j = 0;

}

}

//-------------------------------------------------------------------------

else

{

timerBubbleSort.Enabled = false;//finalizarea sortarii

MessageBox.Show("Sortare Finalizata!");

}

}

-restul funcțiilor auxiliare care se ocupă cu legătura dintre fereastră și alte meniuri sau cu închiderea aplicației

//trecerea in fereastra initiala-------------------------------------------------

private void buttonMeniu\_Click(object sender, EventArgs e)

{

Form1 f1 = new Form1();

f1.Show();

this.Dispose();

}

//-------------------------------------------------------------------------------

//iesirea din aplicatie----------------------------------------------------------

private void buttonExit\_Click(object sender, EventArgs e)

{

Environment.Exit(0);

}

private void BubbleSortIlustratie\_FormClosing(object sender, FormClosingEventArgs e)

{

Environment.Exit(0);

}

//-------------------------------------------------------------------------------

În cazul alegerii opțiunii “Insertion Sort”, meniul cu informații inserat imediat după apăsarea butonului, intitulat “InsertionSortInfo”, este format doar dintr-o funcție ce face legătura cu meniul ilustrarii și o funcție care închide aplicația.

//trecem in urmatoarea fereastra-------------------------------------------------

private void button\_ilustratie\_Click(object sender, EventArgs e)

{

InsertionSortIlustratie isi = new InsertionSortIlustratie();

this.Hide();

isi.Show();

}

//-------------------------------------------------------------------------------

//ne asiguram ca toate ferestrele sunt inchise la iesirea din aplicatie----------

private void InsertionSortInfo\_FormClosing(object sender, FormClosingEventArgs e)

{

Environment.Exit(0);

}

//-------------------------------------------------------------------------------

Meniul care se ocupă cu afișarea ilustrării sortării vectorului după metoda Insertion Sort este intitulat “InsertionSortIlustrație” și are la bază:

-variabilele locale folosite

//variabilele locale folosite----------------------------------------------------

PictureBox[] pictureBoxes = new PictureBox[100];

public int i = 0;

public int j = 0;

public int p;

public Point intermPoint;

public Size intermSize;

public int interschimbari = 0;

//-------------------------------------------------------------------------------

-funcții care creează și afișează vectorul

//crearea vectorului de imagini--------------------------------------------------

public void creeareVector()

{

int i;

for (i = 0; i < Variables.n; i++)

{

pictureBoxes[i] = new PictureBox();

pictureBoxes[i].Name = "pbBox" + i;

pictureBoxes[i].Tag = "pbbox";

pictureBoxes[i].BackColor = Color.Blue;

Controls.Add(pictureBoxes[i]);

pictureBoxes[i].Visible = false;

}

}

//-------------------------------------------------------------------------------

//afisarea vectorului de imagini-------------------------------------------------

public void afisareVector()

{

for (int c = 0; c < Variables.n; c++)

{

pictureBoxes[c].Location = new Point(Variables.lat \* c + Variables.lat, Variables.lat);//pozitie

pictureBoxes[c].Size = new System.Drawing.Size(Variables.lat, Variables.v[c] \* Variables.lat);//marime

pictureBoxes[c].Visible = true;

}

}

//-------------------------------------------------------------------------------

-funcția inițială care este chemată la încărcarea ferestrei și seteaza valorile inițiale

private void InsertionSortIlustratie\_Load(object sender, EventArgs e)

{

interschimbari = 0;

//deblocarea butoanelor---------------------------------------------------

buttonExit.Visible = true;

buttonExit.Enabled = true;

buttonMeniu.Visible = true;

buttonMeniu.Visible = true;

//------------------------------------------------------------------------

//activarea si setarea timer-ului-----------------------------------------

timerInsertionSort.Enabled = true;

timerInsertionSort.Interval = Variables.interval;

//------------------------------------------------------------------------

i = 1;

p = i;

//setarea initiala a coordonatelor butoanelor------------------------------

buttonMeniu.Top = 25;

buttonMeniu.Left = 15;

buttonExit.Top = 25;

buttonExit.Left = 25 + buttonMeniu.Width;

//-------------------------------------------------------------------------

//setarea label-ului--------------------------------------------------------

labelInterschimbari.Text = "Interschimbari: 0";

labelInterschimbari.Top = 5;

labelInterschimbari.Left = 15;

//--------------------------------------------------------------------------

//afisarea initiala a vectorului--------------------------------------------

creeareVector();

afisareVector();

//--------------------------------------------------------------------------

//actualizarea coordonatelor butoanelor in cazul valorilor mari-------------

if (Variables.lat != 50)

{

buttonExit.Location = new Point(pictureBoxes[Variables.n - 1].Left + pictureBoxes[Variables.n - 1].Width + 20,

25);

buttonMeniu.Location = new Point(pictureBoxes[Variables.n - 1].Left + pictureBoxes[Variables.n - 1].Width + 20,

buttonExit.Top + 40);

}

//---------------------------------------------------------------------------

//setarea marimii ferestrei--------------------------------------------------

this.Size = new Size(Math.Max(Variables.lat \* (Variables.n + 2) + 20, Math.Max(buttonMeniu.Left + buttonMeniu.Width + 30, 250)),

Variables.lat \* (Variables.maxi + 3));

//---------------------------------------------------------------------------

}

-funcția care se ocupă cu sortarea vectorului de imagini și a celui de valori

//sortarea vectorului------------------------------------------------------------

public void sortareVector()

{

if (Variables.v[p] < Variables.v[p - 1])

{

//actualizarea numarului de interschimbari---------------------------

interschimbari++;

labelInterschimbari.Text = "Interschimbari: " + interschimbari;

//-------------------------------------------------------------------

//interschimbarea imaginilor-----------------------------------------

intermPoint = pictureBoxes[p].Location;

intermSize = pictureBoxes[p].Size;

pictureBoxes[p].Location = pictureBoxes[p - 1].Location;

pictureBoxes[p].Size = pictureBoxes[p - 1].Size;

pictureBoxes[p - 1].Location = intermPoint;

pictureBoxes[p - 1].Size = intermSize;

//-------------------------------------------------------------------

//interschimbarea valorilor------------------------------------------

int aux = Variables.v[p];

Variables.v[p] = Variables.v[p - 1];

Variables.v[p - 1] = aux;

//-------------------------------------------------------------------

}

}

//-------------------------------------------------------------------------------

-funcția timer-ului, care se execută odată la fiecare jumătate de secundă

private void timerInsertionSort\_Tick(object sender, EventArgs e)

{

//parcurgerea vectorului dupa algoritm---------------------------------------

if (i < Variables.n)

{

if (p > 0)

{

sortareVector();

p--;

afisareVector();

}

else

{

i++;

p = i;

}

}

//---------------------------------------------------------------------------

else

{

afisareVector();

timerInsertionSort.Enabled = false;//finalizarea sortarii

MessageBox.Show("Sortare Finalizata!");

}

}

-restul funcțiilor auxiliare care se ocupă cu legătura dintre fereastră și alte meniuri sau cu închiderea aplicației

//trecerea in fereastra initiala---------------------------------------------------------

private void buttonMeniu\_Click(object sender, EventArgs e)

{

Form1 f1 = new Form1();

f1.Show();

this.Dispose();

}

//-------------------------------------------------------------------------------

//iesirea din aplicatie----------------------------------------------------------

private void buttonExit\_Click(object sender, EventArgs e)

{

Environment.Exit(0);

}

//-------------------------------------------------------------------------------

**BIBLIOGRAFIE**

https://ro.wikipedia.org/wiki/C\_sharp

https://ro.wikipedia.org/wiki/Microsoft\_Visual\_Studio

<https://www.pbinfo.ro/articole/5605/sortarea-prin-selectie>

<https://www.pbinfo.ro/articole/5589/metoda-bulelor>

https://www.pbinfo.ro/articole/5609/sortarea-prin-insertie