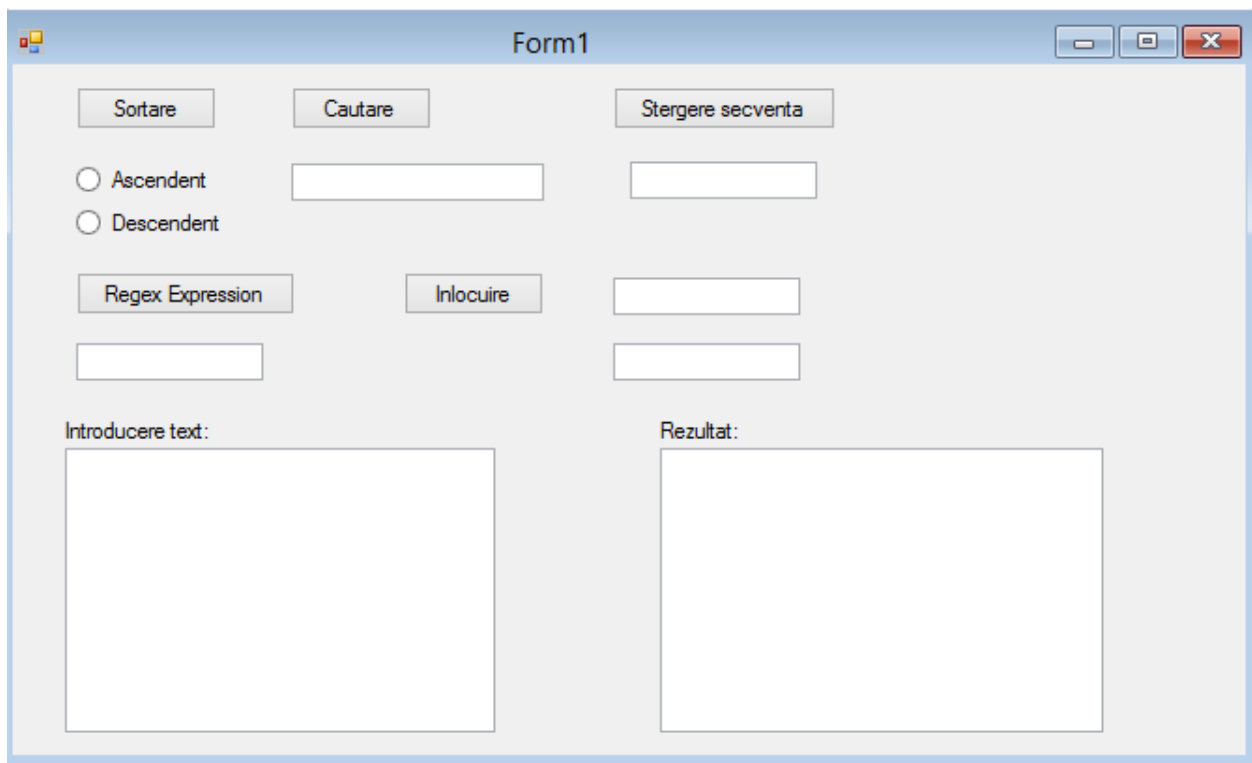


Manipularea textelor

In cadrul acestui laborator vom invata cum sa manipulam texte in diferite moduri, cum ar fi: cautare, inlocuire, sortare etc.

Pentru a incepe vom crea un nou proiect de tipul Windows forms si il vom seta ca in figura urmatoare:



Pentru a realiza sorta rea ascendent sau descendent va trebui prima data sa transformam string-ul din textbox-ul de introducere text intr-un vector de string-uri, iar acest lucru se face prin folosirea fuctie Split care se va apela dupa fiecare spatiu. Apoi pentru a sorta ascendent folosim functi Sort din biblioteca Array, apoi pentru a reface string-ul initial vom folosi metoda Join pentru care vom adauga un spatiu dupa fiecare element din vectorul creat initial.

Pentru sortarea descendenta vom folosi aceeasi tehnica de sortare, numai ca de data aceasta vom folosi metoda Reverse din biblioteca Array, care va inversa vectorul sortat.

```

private void button1_Click(object sender, EventArgs e)
{
    string[] tempString = textBox1.Text.Split(' ').ToArray<string>();
    if (radioButton1.Checked)
    {
        Array.Sort(tempString);
        textBox2.Text = String.Join(" ", tempString);
    }
    else
    {
        Array.Sort(tempString);
        Array.Reverse(tempString);
        textBox2.Text = String.Join(" ", tempString);
    }
}

```

Rezultatul ar trebui sa fie ca in cele 2 figuri de mai jos:

The screenshot shows a Windows application window titled "Form1". The interface contains several controls:

- Buttons: "Sortare", "Cautare", "Stergere secventa", "Regex Expression", and "Inlocuire".
- Radio buttons: "Ascendent" (selected) and "Descendent".
- Text boxes: Two for input and two for output.
- Text area: "Introducere text:" containing the text "In acest laborator vom invata manipularea textelor folosind diferite metode".
- Text area: "Rezultat:" containing the text "acest diferite folosind In invata laborator manipularea metode textelor vom".

Pentru a cauta un cuvânt într-un text, vom face aceeași împărțire într-un vector, după care vom folosi metoda `Contains` pentru a verifica dacă textul căutat există în vector, iar dacă există vom afișa la ce index se află folosind metoda `Array.IndexOf`, metoda care ia ca parametru un vector și o valoare și returnează indexul la care acea valoare se află, iar dacă nu vom afișa că nu am găsit cuvântul căutat. De precizat că `Array.IndexOf` trebuie incrementat cu 1, pentru că începe număratoarea de la 0.

```
private void button2_Click(object sender, EventArgs e)
{
    string[] tempString = textBox1.Text.Split(' ').ToArray<string>();
    if (tempString.Contains(textBox3.Text)){
        textBox2.Text = "este al " + (Array.IndexOf(tempString, textBox3.Text) + 1) + "-lea cuvânt în text";
    }
    else
    {
        textBox2.Text = "Cuvântul nu a fost găsit";
    }
}
```

Rezultatul ar trebui să fie ca în figura următoare:

Pentru stergerea unui cuvint dintr-un text dat, prima data vom folosi principiul de la cautarea unui cuvint, dupa care vom folosi `Array.Clear`, care va primi 3 parametri (1. Un vector, 2. Indexul de la care pornim stergerea, 3. Cate pozitii din vector vom sterge).

```
private void button4_Click(object sender, EventArgs e)
{
    string[] tempString = textBox1.Text.Split(' ').ToArray<string>();
    if (tempString.Contains(textBox7.Text))
    {
        Array.Clear(tempString, Array.IndexOf(tempString, textBox7.Text), 1);
        textBox2.Text = String.Join(" ", tempString);
    }
    else
    {
        textBox2.Text = "Cuvantul nu exista in acest text";
    }
}
```

Rezultatul va fi ca in imaginea urmatoare:

Inlocuirea unui text se va face prin luarea unui cuvânt, cautarea lui în textul dat și înlocuirea cu un alt text la alegere. Singura diferență față de căutare, este aceea că luăm indexul elementului și la acel index în vector asignăm noua valoare introdusă.

```
private void button3_Click(object sender, EventArgs e)
{
    string[] tempString = textBox1.Text.Split(' ').ToArray<string>();
    if (tempString.Contains(textBox5.Text))
    {
        int indexElement = Array.IndexOf(tempString, textBox5.Text);
        tempString[indexElement] = textBox6.Text;
        textBox2.Text = String.Join(" ", tempString);
    }
    else
    {
        textBox2.Text = "Cuvantul ce trebuie inlocuit nu exista in acest text";
    }
}
```

Rezultatul va fi ca în imaginea următoare:

The screenshot shows a Windows application window titled "Form1". At the top, there are three buttons: "Sortare", "Cautare", and "Stergere secventa". Below "Sortare" are two radio buttons: "Ascendent" and "Descendent". To the right of these are two empty text boxes. Below "Cautare" is a "Regex Expression" button and an empty text box. To the right of "Regex Expression" is a blue-bordered "Inlocuire" button. Below "Inlocuire" are two text boxes containing the text "laborator" and "curs". At the bottom, there are two large text areas. The left one is labeled "Introducere text:" and contains the text "In acest laborator vom invata manipularea textelor folosind diferite metode". The right one is labeled "Rezultat:" and contains the text "In acest curs vom invata manipularea textelor folosind diferite metode".

Ultima tehnica de manipulare a textelor in cadrul acestui laborator va fi Regex (Regular Expression Language). Regex foloseste un set de patter-uri bine stabilite pentru a cauta intr-un text dat si a returna o valoare sau mai multe din cadrul textului dat. Regex este folosit in special in cautari in baze de date unde performanta nu este o problema, ci usurinta cu care cautam este scopul principal.

Pentru o lista completa cu patter-urile pe care Regex le foloseste in C# puteti accesa urmatorul link: [https://msdn.microsoft.com/en-us/library/az24scfc\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/az24scfc(v=vs.110).aspx)

Pentru a folosi Regex trebui pentru inceput sa introducem namespace-ul unde aceasta clasa se gaseste, acesta fiind: `using System.Text.RegularExpressions;`

Apoi vom crea o noua instanta a clasei Regex care primeste ca parametru textul introdus de noi. Dupa care se creaza o noua instanta a clasei Match, care va cauta in textul nostru pattern-ul introdus, iar daca acest patter a fost gasit cu success, vom afisa rezultatul.

```

private void button6_Click(object sender, EventArgs e)
{
    Regex regex = new Regex(textBox4.Text);
    Match match = regex.Match(textBox1.Text);
    if (match.Success)
    {
        textBox2.Text = match.Value;
    }
    else
    {
        textBox2.Text = "Patternul a fost introdus gresit sau nu exista in text";
    }
}

```

De exemplu daca avem un numar de telefon si dorim sa aflam prefixul care este din 3 cifre vom folosi urmatorul pattern: `^\d{3}` care va lua inceputul unui string din care va alege primele 3 cifre si le va afisa.

The screenshot shows a Windows application window titled "Form1". The interface contains several controls:

- Buttons: "Sortare", "Cautare", "Stergere secventa", "Inlocuire", and "Regex Expression".
- Radio buttons: "Ascendent" and "Descendent".
- Text boxes: One for the regex expression containing `^d{3}`, and another empty one next to the "Inlocuire" button.
- Text area: "Introducere text:" containing the string "901-333-452".
- Text area: "Rezultat:" containing the string "901".

Un exemplu pe text ar fi folosirea pattern-ului `a.e`, care va returna toate secventele de text din 3 litere care incep cu a si se termina cu e.

Form1

Sortare Cautare Stergere secventa

Ascendent

Descendent

Regex Expression Inlocuire

a.e

Introducere text: Rezultat:

In acest laborator vom invata manipularea
textelor folosind diferite metode

ace