Laborator 5 – Vectori. Excepții

Obiective

După completarea acestui laborator veți dobândi următoarele cunoștințe:

- Crearea și folosirea vectorilor de tipuri valoare
- Trimiterea de argumente metodei **Main**
- Crearea și folosirea unor vectori cu dimensiuni calculate la run-time
- Folosirea vectorilor de diferite ranguri

Condiţii prealabile

Înaintea de a realiza acest laborator trebuie să fiţi familiarizaţi cu următoarele concepte:

- Folosirea instrucțiunilor C#
- Folosirea metodelor C#

➢ Exerciţiul 1- Lucrul cu vectori de tipuri valoare

În acest exercițiu, veți scrie un program ce primește un nume de fișier text sau mai multe ca argument al metodei Main. Programul va sumariza conținutul fișierelor text. Va citi conținutul într-un vector de caractere și apoi va parcurge acest vector, numărând consoanele și vocalele. La final, programul va afișa numărul total de consoane, vocale și numărul de linii din fișiere.

Pentru a citi numele fișierelor primite ca argumente

- Creați un proiect nou, de tipul **ConsoleApplication**
- Adăugați un vector de string-uri numit **args** ca parametru al metodei **Main.**
- Parcurgeți vectorul și afișați la consolă parametrii primiți din linia de comandă. Verificați limitele vectorului în momentul în care faceți parcurgerea.
- Salvați programul;
- Compilați și rulați.

Citirea conținutului fișierului text într-un vector

- Pentru fiecare element din vectorul de parametrii, realizați următoarele operații.
- Declarați o variabilă de tip string **fileName** în care copiați valoarea parametrului curent.
- Definiți o variabilă de tipul **FileStream** pentru deschiderea de fișiere text. Instanțiați această variabilă, oferind ca parametrii numele fișierului și **FileMode.Open**.
  Observați excepțiile pe care le pot arunca metodele folosite și prindeți-le!
- Definiți o variabilă **StreamReader** pentru a citi conținutul fișierului. Instanțiați această variabilă, oferind ca parametru variabila de tipul FileStream definită la pasul anterior.

- Afișați dimensiunea în bytes a fișierului deschis, folosind proprietatea **Length** a variabilei de tip FileStream.

- Codul ar trebui să arate așa:

```
string fileName = args[0];
FileStream stream = new FileStream(fileName, FileMode.Open);
StreamReader reader = new StreamReader(stream);
Console.WriteLine((int)stream.Length);
```

- Conținutul fișierului va fi citit într-un vector de char. Declarați acest vector; luați în considerare dimensiunea fișierului la instanțiere.
- Pentru fiecare caracter din fișierul deschis, apelați metoda Read a variabilei **StreamRead** pentru a obține caracterul curent. Salvați-l în vectorul de caractere declarat.
- Parcurgeți vectorul de caractere la final, afișând conținutul fișierului deschis.
- Salvați programul.
- Compilați și testați funcționarea sa.

Clasificarea și afișarea conținutului fișierului

- Definiți o metodă statică numită **Summarize** în clasa curentă. Metoda nu va întoarce nimic și primește ca parametru vectorul de caractere ce conține fișierul citit.
- Parcurgeți acest vector în metoda **Summarize** și inspectați fiecare element. Numărați numărul de consoane, vocale și caractere NewLine ('\n') reținând rezultatele în variabile separate. Afișați la finalul parcurgerii valorile obținute.
- Apelați metoda Summarize din metoda Main, trimițând ca parametru vectorul ce conține fișierul citit.
- Salvați programul
- Compilați și testați funcționarea sa.

➢ Exercițiul 2- Înmulțirea matricelor

În acest exercițiu, veți scrie un program ce va înmulți două matrice. Programul va citi de la consolă dimensiunea matricelor ce urmează să fie înmulțite precum și valorile elementelor acestora. Se va aplica apoi algoritmul de înmulțire a matricelor pentru a se obține rezultatul, ce va fi apoi afișat.

Pași pentru rezolvarea exercițiului

- Creați un proiect nou, de tipul **Console Application**
- Declarați 3 vectori de tipul int, bidimensionali, pentru a reține cele două matrice și rezultatul
- Afișați un mesaj prin care utilizatorului i se spune să introducă dimensiunile primei matrice

- Instanțiați prima matrice și apoi parcurgeți toate elementele acesteia, pentru fiecare citind de la tastatură valoarea pe care utilizatorul dorește să o introducă și parsând-o la o valoare întreagă.
- Faceți același lucru și pentru a doua matrice.
- Verificați că numărul de coloane a primei matrice este egal cu numărul de linii din a doua. Aceasta este condiția ca înmulțirea să se poată executa.
- Aplicați algoritmul de înmulțire a matricelor pentru a afla rezultatul. Întrebați instructorul pentru a exemplifica algoritmul de înmulțire de matrice.

➢ Exercițiul 3- Aruncarea și prinderea excepțiilor

În acest exercițiu veți mări funcționalitatea programului scris la exercițiul anterior. Programul va verifica dacă numarul de coloane a primei matrice este egal cu numarul de linii din a doua. Dacă acest lucru nu se întâmplă, se aruncă o excepție **ArgumentException** ("Numărul de coloane din prima matrice trebuie să fie egal cu numărul de linii din a doua"). Programul va prinde această excepție într-o clauză **catch** și va afișa un mesaj de diagnosticare.

Pentru a valida numărul de linii și numărul de coloane

- Includeți toate instrucțiunile din metoda Main într-un bloc **try**.
- Adăugați un bloc **catch** la finalul blocului **try,** pentru a prinde excepțiile de tipul **System.Exception**. În blocul **catch**, afișați excepția prinsă la consolă.
- Introduceți, în Main, după citirea matricelor, instrucțiuni pentru a verifica dacă numărul de coloane a primei matrice este egal cu numarul de linii din a doua. Dacă nu este îndeplinită condiţia, aruncați o excepție de tipul **System.ArgumentOutOfRangeException**. Adăugați un mesaj la crearea obiectului excepție.
- Salvați programul.
- Compilați și testați. Verificați pentru numere ce ar trebui să provoace excepții.