

Laborator 3 – Tipuri de instrucțiuni

Obiective

După completarea acestui laborator veți dobândi următoarele cunoștințe:

- Crearea unor noi tipuri de valoare
- Utilizarea instrucțiunilor

Condiții prealabile

Înainte de a realiza acest laborator trebuie să fiți familiarizați cu următoarele concepte:

- Utilizarea funcțiilor de **input/output**
- Variabile cu tipuri de valoare (creare și utilizare)

➤ Exercițiul 1- Crearea și utilizarea unui tip **enum**

În acest exercițiu veți crea un tip de date enumerație pentru a reprezenta diferite tipuri de conturi bancare (**Cec** și **Deposit**). Veți crea două variabile de acest tip enumerație și veți seta valorile lor. După aceea, veți printa cele două variabile utilizând funcția de output **System.Console.WriteLine()**.

- Creați un fișier **Enumeration.cs** și adăugați un tip **enum** numit **AccountType** înainte de definiția clasei ce conține metoda **Main()**. O variabilă de tip **AccountType** poate avea valorile {**Cec**, **Deposit**}.
- Creați o clasă numită **Enumeration** care va conține metoda **Main()**. În interiorul metodei **Main()** declarați două variabile **account1** și **account2** de tip **AccountType**.
- Inițializați cele două variabile astfel: **account1** cu valoarea **AccountType.Cec**, **account2** cu valoarea **AccountType.Deposit**.
- Afișați valorile celor două variabile utilizând funcția specificată în enunț.
- Compilați și rulați programul.

➤ Exercițiul 2 – Crearea și utilizarea unui tip **struct**

În acest exercițiu veți defini o structură ce poate fi folosită pentru a reprezenta un cont bancar. Veți utiliza variabile pentru a păstra: numărul de cont (de tip **long**), dobânda (de tip **int**), tipul contului (enumerația care a fost creată la Exercițiul 1). După aceea, veți crea o variabilă de acest tip, veți atribui anumite valori câmpurilor acesteia și le veți afișa la consolă.

- Creați un fișier **Structure.cs** și definiți în interiorul acestuia enumerația de la Exercițiul 1.
- Definiți o structură publică (**public struct**) numită **BankAccount** care conține câmpurile:



Nume Variabilă	Tip Variabilă
accNumber	long
accInterest	int
accType	AccountType

- Creați o clasă numită **Structure** în interiorul căreia definiți metoda **Main()**. În **Main()** declarați o variabilă numită **account**, de tip **BankAccount**.
- Inițializați câmpurile variabilei **account** cu următoarele valori:

Nume Variabilă	Valoare
accNumber	123
accInterest	3200
accType	AccountType.Cec

- Afișați la consolă valorile câmpurilor variabilei **account**.
- Compilați și rulați programul.

➤ Exercițiul 3 – Adăugarea funcționalităților **input/output**

În acest exercițiu veți modifica programul scris la Exercițiul 2. În loc să atribuiți un anumit număr de cont câmpului corespunzător, veți cere utilizatorului să introducă unul de la tastatură. În final, veți printa informațiile despre contul introdus.

Observație!

Veți avea nevoie să folosiți metoda **long.Parse()** pentru a converti șirul de caractere citit cu instrucțiunea **Console.ReadLine()** într-o valoare de tip **long**.

- Deschideți fișierul **Structure.cs** și ștergeți atribuirea câmpului **accNumber** al variabilei **account** (**account.accNumber**). În locul acestei atribuirii, printați un mesaj la consolă prin care îi cereți utilizatorului să introducă un număr de cont.
- Citiți valoarea introdusă de utilizator de la tastatură și asigurați-o câmpului **account.accNumber**.
- Compilați și rulați programul. Introduceți o valoare numerică atunci când se va afișa mesajul corespunzător.



➤ Exercițiul 4 – Transformarea din bază 10 în bază 16

În acest exercitiu, veți scrie un program ce va cere utilizatorului să introducă un număr în baza 10. Acest număr va fi parsat și apoi transformat în baza 16. De exemplu, dacă utilizatorul introduce numărul 168, rezultatul afișat de program va trebui să fie A8.

Pași pentru rezolvarea exercițiului

- Adăugați un nou fișier `Conversion.cs`
- Creați o clasă numită **Conversion**, în interiorul căreia definiți metoda **Main()**
- Afișați un mesaj prin care utilizatorului i se spune să introducă un număr în baza 10
- Citiți și parsați numărul introdus de utilizator
- Pentru a realiza transformarea în bază 16, realizați împărțiri succesive la 16; realizați împărțirile până când numărul inițial ajunge să fie 0;
- La fiecare pas al împărțirii, rețineți restul. Pentru că transformarea se va face în baza 16, pe lângă cifrele 0-9 se vor folosi și literele A-F(pentru resturile 10-15). Din acest motiv, numărul în baza 16 se va asambla într-un string.
- **Obs:** pentru a adăuga caractere la începutul unui string, folosiți cod de forma:

```
string exemplu = "salut";  
exemplu = "Formula de" + exemplu;  
Console.WriteLine(exemplu);
```

va afișa "Formula de salut"

- Încercați să transformați din resturile 10-15 în A-F

