

Laborator 2 – Folosirea colecțiilor

Obiective

După completarea acestui laborator veți dobândi următoarele cunoștințe:

- Folosirea colecției generice Dictionary pentru a stoca perechi cheie/valoare puternic tipate
- Folosiți colecția generică List pentru a stoca liste de obiecte puternic tipate
- Folosiți colecția specializată NameValueCollection pentru a stoca un dicționar ce are chei de tip vector și valori multi-string-uri

Scenariul de laborator

Ești dezvoltator pentru o organizație care implementează aplicații .NET Framework. Ai fost rugat să realizezi o aplicație Windows Forms numită AccountManager astfel încât aceasta va folosi clasele colecție oferite de platforma .NET

➤ Exercițiul 1- Utilizarea colecției generice Dictionary

În acest exercițiu vă veți familiariza cu aplicația AccountManager, după care veți modifica aplicația astfel încât să folosească o colecție Dictionary pentru a stoca conturile. Cheile dicționarului vor fi de tip întreg, reprezentând numărul contului. Valorile vor fi obiecte de tipul Account.

- Înainte de a începe efectiv codarea, familiarizați-vă cu aplicația AccountManager.
- Deschideți fișierul MainForm și selectați să vizualizați codul paginii
- Identificați în clasa MainForm locul unde trebuie să definiți o clasă Dictionary. Definiți o variabilă de tip Dictionary cu numele **_accounts** și instanțiați-o. Specificați ca tip pentru cheie int, iar pentru valoare Account
- Identificați metoda AddAccountButton_Click și scrieți codul necesar pentru a adăuga obiectul **acc** dicționarului creat. Specificați **_nextAccountNumber** drept cheie pentru noua intrare în dicționar
- Identificați metoda DisplayAccounts. Scrieți codul necesar pentru a itera prin intrările din dicționar. Pentru fiecare intrare, adăugați o linie nouă în controlul AccountsDataGridView pentru a afișa numărul contului în prima coloană și detaliile acestuia în a doua.
- Localizați metoda AccountsDataGridView_CellDoubleClick și modificați codul existent. Căutarea obiectului Account trebuie făcută în dicționar. Specificați **accNum** drept cheie și depuneți obiectul găsit în variabila **theAccount**
- Rulați și testați aplicația

➤ Exercițiul 2- Folosirea colecției generice List

Clasa Account menține o colecție List pentru obiectele de tip Transaction. Fiecare obiect Transaction reprezintă un depozit sau o retragere și indică suma și data la care a avut loc tranzacția. În acest exercițiu, veți modifica clasa Account astfel încât aceasta să sorteze elementele din List în funcție de data și de sumă. Operațiile de sortare vor folosi următoarele clase, ambele implementări ale interfeței generice IComparer

- TransactionDateComparer va compara obiectele Transaction după dată
- TransactionAmountComparer va compara obiectele Transaction după sumă

Veți defini de asemenea metodele **IsDeposit** și **IsWithdrawal** pentru clasa Account pentru a indica dacă un anumit obiect Transaction reprezintă un depozit sau o retragere. Veți folosi apoi aceste metode drept predicat pentru metoda List.FindAll pentru a găsi toate obiectele Transaction ce sunt depozite sau, respectiv, retrageri.

Definirea claselor ce implementează interfață generică IComparer

- Definiți o clasă numită **TransactionDateComparer** ce implementează interfața **IComparer<Transaction<double>>**
- În clasa TransactionDateComparer, implementați metoda **Compare** astfel încât să compare proprietatea **Timestamp** a celor două obiecte **Transaction<double>**
- Același lucru pentru clasa **TransactionAmountComparer**

Sortarea obiectelor din colecția List folosind clasele create

- Localizați metoda **GetTransactionsByDate** în clasa Account. Scopul acestei metode este de a popula cei 3 parametri List după cum urmează:
 - Popularea parametrul **all** cu lista de tranzacții sortată după dată
 - Popularea parametrul **deposits** cu lista tranzacțiilor tip depozit sortată după dată
 - Popularea parametrul **withdrawals** cu lista tranzacțiilor tip retragere sortată după dată
- Modificați instrucțiunea ce setează parametrul **all** astfel încât să creați o listă nouă care este o copie a instanței variabilei **_allTransactions**
- Sortați lista **all** apelând metoda **Sort** pasând ca parametru un obiect **TransactionDateComparer**
- Localizați metoda **GetTransactionByAmount**. Aceasta este similară precedentei, doar că va sorta elementele după sumă
- Modificați instrucțiunea de setare a parametrului **all** al acestei metode ca mai sus. Sortați apoi lista obținută după sumă

Rulați și testați aplicația

Identificarea obiectelor dintr-o colecție List folosind predicate

- La finalul clasei Account, identificați comentariul **TODO: Define IsDeposit method**. Implementați metoda **IsDeposit** astfel încât aceasta să primească ca parametru un obiect **Transaction<double>** și să întoarcă o valoare Booleană pentru a indica dacă valoarea tranzacției este 0 sau mai mare

- Identificați comentariul **TODO: Define IsWithdrawal method**. Implementați metoda **IsWithdrawal** astfel încât aceasta să primească ca parametru un obiect **Transaction<double>** și să întoarcă o valoare Booleană pentru a indica dacă valoarea tranzacției este mai mică de 0
- În metoda **GetTransactionByDate**, modificați instrucțiunile astfel:
 - Folosind metoda **FindAll** și predicatul **IsDeposit** asupra listei **all**, obțineți în **deposits** lista tranzacțiilor de tip depozit. Sortați apoi această listă apelând metoda **Sort**.
 - Asemănător, pentru parametrul **withdrawals**, obțineți lista tranzacțiilor de tip retragere folosind predicate și sortați-o
- Realizați aceleași operații pentru metoda **GetTransactionByAmount**

Rulați și testați aplicația

➤ Exercițiul 3 – Folosirea colecției specializate **NameValueCollection**

În acest exercițiu veți spori funcționalitatea clasei **Account** astfel încât aceasta să reține anumite note pentru cont. Veți folosi un obiect **NameValueCollection** pentru a stoca notele. Acest obiect va conține 3 chei, ce specifică următoarele:

- Telephone notes
- E-mail notes
- Letter notes

Fiecare dintre aceste chei va avea asociate mai multe valori string. De exemplu, intrarea pentru notele de e-mail va conține mai multe valori string ce reprezintă toate discuțiile purtate cu deținătorul contului prin e-mail. Pașii în realizarea acestei funcționalități sunt:

- Crearea și inițializarea colecției **NameValueCollection**
- Afișarea cheilor pentru colecția **NameValueCollection**
- Adăugarea și afișarea de valori pentru o anumită cheie din colecție

Crearea și inițializarea colecției **NameValueCollection**

- În fișierul **Account**, adăugați o directivă **using** pentru a putea folosi colecții specializate
- Identificați comentariul **TODO: Define a NameValueCollection instance variable**. Definiți aici o variabilă de acest tip cu numele **_notes**. Instanțiați variabila.
- În constructorul clasei **Account**, adăugați cele 3 chei colecției, setând pe **null** valorile pentru fiecare dintre acestea.
- Definiți o proprietate **read-only** ce întoarce colecția **_notes**.

Afișarea cheilor pentru colecția **NameValueCollection**

- Deschideți **NotesForm** din **Soluțion Explorer**. În această fereastră:

- Controlul **KeysComboBox** va afișa cheile colecției **Notes** ale unui cont. Cheile sunt: **Telephone notes, E-mail notes** și **Letter notes**
 - Butonul **AddButton** va adăuga o nouă valoare pentru cheia selectată
 - Controlul **NotesListBox** va afișa toate valorile pentru cheia selectată
- În constructorul clasei **NotesForm**, localizați comentariul **TODO: Display the Notes keys in the combo box**. Adăugați o instrucțiune ce întoarce toate cheile colecției **_theAccount.Notes**, legând apoi aceste chei de controlul **KeysComboBox**. (Hint: **KeysComboBox.DataSource**)
Adăugarea și afișarea de valori pentru o anumită cheie din colecție
- Identificați metoda **AddButton_Click** și observați codul existent. Adăugați valoare obținută în colecție în locul indicat de comentariu.
- Identificați metoda **DisplayNotes** și observați codul existent. Înlocuiți instrucțiunea marcată de comentariul **TODO: Get values for key** cu o alta ce obține toate valorile asociate cheii curente din colecția **_theAccount.Notes**.

Rulați și testați aplicația.