UNIVERSITATEA “STEFAN CEL MARE”, SUCEAVA

FACULTATEA DE INGINERIE ELECTRICA SI STIINTA CALCULATOARELOR

SPECIALIZAREA CALCULATOARE

**PROIECT DISCIPLINA PIU**

**Programare interfete utilizator**

Autor: Maxemiuc Alina

Grupa: 3122B

An de studiu: II

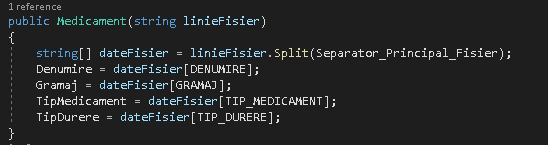
**Gestiunea unei farmacii**

In cadrul proiectului de laborator la disciplina PIU (Programare Interfete Utilizator) am ales tema de gestiune a unei farmacii.

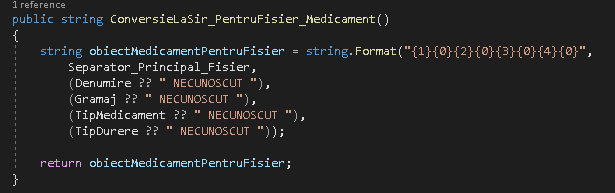
Voi prezenta pe scurt clasele folosite si rostul acestora:

1. **Clasa Medicament** (aflata in namespace-ul StocareMedicamente): contine denumirea, gramajul, tipul medicamentului (pilule, capsule, unguent, picaturi, injectie) si tipul de durere pe care il combate medicamentul ( cap, spate, ochi, dinti etc.). Pentru inceput voi initializa constantele, voi seta proprietarile auto-implemented, iar mai apoi voi folosi un constructor implicit in care voi seta valoarea elementelor amintite mai sus la Empty(am golit sirurile), trecand astfel la un constructor cu parametri unde voi crea alte variabile pe care le voi egala cu proprietatile auto-implemented.

Functia din aceasta imagine ia datele de care avem nevoie din fisierul creat.

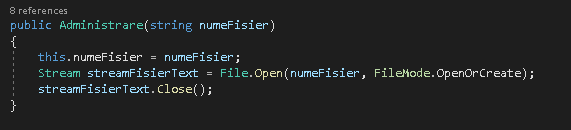


Functia de conversie la sir ne asigura formatul in care dorim sa apara obiectele noastre in consola.

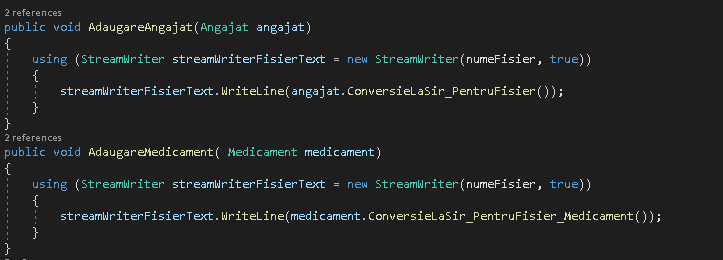


1. **Clasa Angajat** (aflata in namespace-ul AngajatiFarmacie): contine numele, prenumele, varsta si codul angajatului. Asemanator ca la clasa Medicament, aici vom regasi initializarea unor constante, proprietati auto-implemented, constructor implicit, constructor cu parametri, functie de citire, etc.
2. **Clasa Administrare** (aflata in namespace-ul AdministareFarmacie): contine initializare de constante, functia de crearea a unui fisier, de adaugare a medicamentelor si angajatilor si functiile GetAngajati si GetMedicamente.

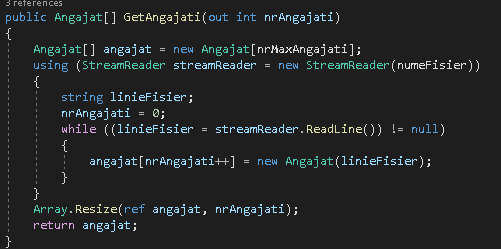
Cu ajutorul functiei urmatoare realizam citirea datelor din fisier, in cazul in care nu exista acel fisier, proprietatea OpenOrCreate, il va crea.



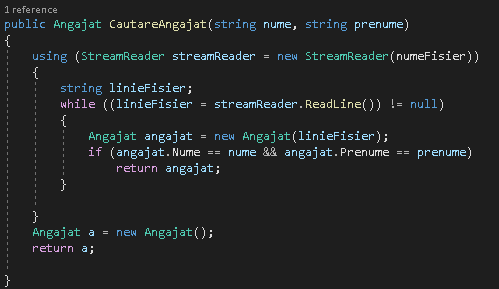
Functiile de adaugare angajat si medicament vor deschide fisierul si vor scrie in acesta detalii despre angajat sau despre medicament.



In cazul functiei de mai jos, instructiunea „using” va apela streamReader.Close(), urmand sa citeasca cate o linie si sa creeze un obiect de tip Angajat pe baza datelor din linia citita. Analog se va intampla si la functia GetMedicamente().



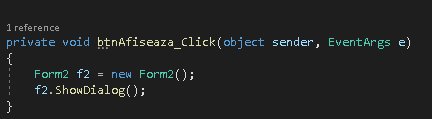
Functia de cautare angajat foloseste instructiuna „using” asemator la ca functia GetAngajati(), apoi se citeste cate o linie si se creaza un obiect de tip Angajat pe baza datelor din linia citita, ca mai apoi sa verficam daca proprietatea auto-implemented Nume este egala cu parametrul nume al functiei noastre(analog pentru prenume), iar daca este adevarat vom afisa obiectul creat.



1. **Clasa Program** (namespace-ul FarmacieGestiune): reprezinta meniul proiectului. Contine functiile de afisare angajati si medicamente.

In continuare voi vorbi despre interfata grafica. Aceasta contine 5 form-uri:

1. **Form1.cs** – este pagina de start a aplicatiei in care vom intalni 4 butoane care dupa apasarea lor vor deschide alte form-uri, iar butonul de „Iesire” va inchide aplicatia. Aici doar am deschis celelalte form-uri cu ajutorul metodei ShowDialog():



1. **Form2.cs** – se va deschide in urma apasarii butonului „Afisare angajati”. Vom avea nevoie de 4 label-uri pentru a retine numele, prenumele, varsta si codul angajatilor, constantele initializate ajuta la afisarea informatiilor pe ecran. In primul rand, in interiorul constructorului implicit Form2 vom folosi metoda de deschidere a fisierului din care dorim sa luam informatiile necesare. Am creat o functie separata de afisare a angajatilor in care am adaugat controale de tip label pentru toate detaliile unui angajat.
2. **Form3.cs** – se va deschide in urma apasarii butonului „Salveaza angajati”. Aici se regasesc 4 controale de tip label pentru informatiile angajatilor si 4 controle de tip textBox care ne vor ajuta in adaugarea acestora in fisier. Vom mai regasi 3 butoane: unul care efectueaza salvarea informatiilor in fisier, unul de anulare, iar altul de iesire din form. Asemanator ca la Form2.cs am folosit metoda de deschidere a fisierului. In cadrul functiei de afisare, am folosit un o instructiune foreach in care cu ajutorul clasei Control(clasa deja existenta in Visual Studio) vom lua pe rand butoanele existente in codul nostru. In interiorul unei instructiuni de tip if vom verifica daca tipul butoanelor noastre este la fel ca cel Button (initializeaza o noua instanta a clasei Button). Codul folosit pentru butonul de salveaza va trece pe rand prin fiecare textBox si va verifica daca informatiile sunt scrise corect (de exemplu daca la nume sunt scrise doar caractere sau la varsta doar cifre). Codul din spatele acestui buton reprezinta de fapt validarea datelor, pentru ca utilizatorul sa nu introduca cifre la nume sau prenume sau in loc de un cod angajat sa treaca caractere la intamplare.
3. **Form4.cs** – se va deschide in urma apasarii butonului „Afisare medicamente”. Aici se va intampla analog ca in Form2.cs.
4. **Form5.cs** – se va deschide in urma apasarii butonului „Salveaza medicamente”. Analog ca in Form3.cs.