Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică şi Microelectronică

**Raport**

**Lucrarea de laborator nr. 1**

PAD

**Tema:** Agent de mesagerie

**A efectuat:** st. gr*.* SI-141 Chetrușca Anastasia

**A verificat:**  lect. sup. A. Gavrișco

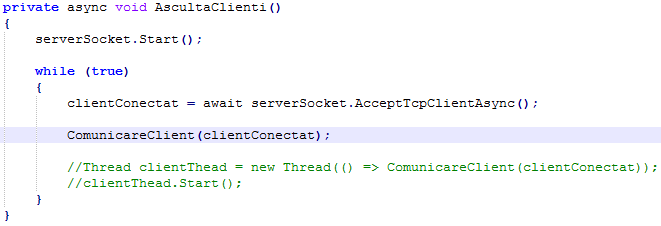
Chişinău 2017

**Scopul lucrarii:**

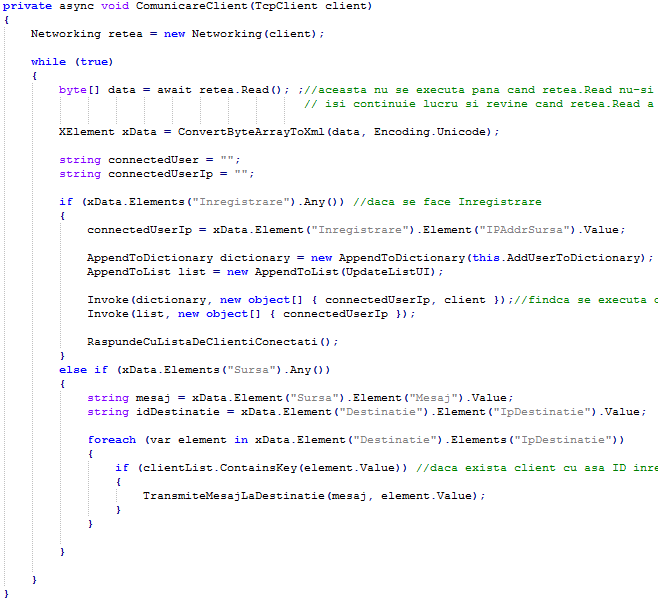
Integrarea bazată pe agenți de mesaje care ar permite o comunicare asincronă dintre componentele distribuite ale unui sistem

**Realizarea lucrarii:**

1. Crearea metodei AscultaClienti().

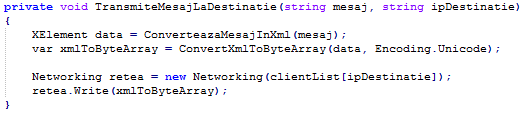
Metoda *AscultaClienti()* folosește un ciclu infinit care asculta clienți la adresa și portul indicat de obiectul *serverSocket*, ascultarea clientilor este asincrona, ceia ce înseamnă ca firul de execuție nu se blochează cât timp se fectueaza ascultarea, ci continue mai departe și revine când sa conectat un client. Atunci când se conecteaza un client se populeaza cu date un obiect (clientConectat) de tip *TcpClient* și se transmite metodei *ComunicareClient() care la fel este asincrona.*

1. Crearea metodei ComunicareClient :

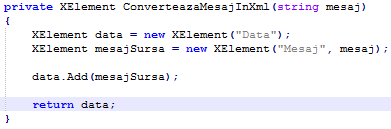
Metoda *ComunicareClient* este folosită pentru comunicarea cu un client conectat la *Broker,* metoda folosește un obiect de tip *Networking* și un ciclu infinit pentru a citi data din canalul de comunicare, dacă acestea exista, care la fel se efectueaza asincron. Datele transmise prin canalul de comunicare sunt în format XML, create cu ajutorul clasei *XElement,* deși sirul de date din canal sunt niște bytes, aceștia sunt convertiti în date XML cu ajutorul metodei *ConvertByteArrayToXml().*

Dacă Broker primește un mesaj de la utilizator cu un nod cu numele *Inregistrare* atunci Broker va considera ca se inregistreaza un utilizator nou, acesta va prelua adresa IP al utilizatorului din nodul cu numele *IPAddrSursa,* datele despre acest nou client conectat sunt pastrate într-un dicționar (Cheie – Valoare), unde cheia este adresa IP al clientului conectat iar valoarea este obiectul de tip *TcpClient* primit ca parametru la începutul metodei.

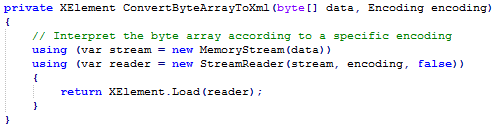
În caz dacă nu exista un nod cu numele *Inregistrare,* atunci Broker va considera ca un client încearcă sa transmite un mesaj altui client, aceasta se verifica prin prezenta nodului cu numele *Sursa* în mesajul obținut din canal de comunicare, după care Broker „scoate” mesajul clientului din mesajul XML care se afla la nodul cu numele *Mesaj*, după care scoate adresa IP din mesajul XML prin nodul cu numele *IpDestinatie* și dacă exista în dicționar cu cheia egala cu adresa IP obtinuta, atunci Broker va transmite mesajul la clientul destinație.



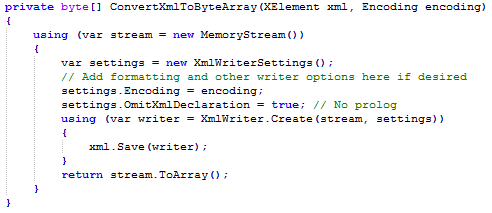
Metoda *TransmiteMesajLaDestinatie()*, accepta doi parametri, mesajul care trebuie transmis, și adresa IP al clientului destinație care este folosit ca cheie pentru a obtine valoarea din dicționar. În primul rând se adauga mesajul de tip *string* intr-un mesaj de tip *XElement* (care va conține și alte date) care prin urmare va fi transformat într-un sir de bytes prin metoda *ConvertXmlToByteArray* pentru a fi transmis prin canalul de comunicare la destinație. Cu ajutorul metodei *write* din clasa *Networking* sirul de bytes este transmis în canalul de comunicare.



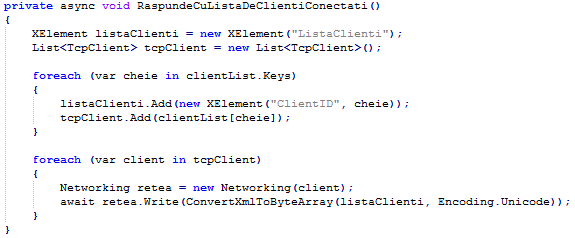
Metoda ConverteazaMesajInXml adauga mesajul clar într-un mesaj de tip XML, aici se creaza un nod cu numele *Data* după care este creat încă un nod cu numele *Mesaj* și care conține mesajul, după care nodul *Mesaj* este adaugat la nodul *Data* care este returnat.



Metoda *ConvertByteArrayToXml* converteste un sir de bytes într-un mesaj XML, aici se creaza o reprezentare în memorie a mesajului cu ajutorul clasei *MemoryStream,* apoi cu ajutorul unui cititor(reader) de tip *StreamReader* pentru a citi aceste date și sa le converteasca în XML, dacă este posibil.



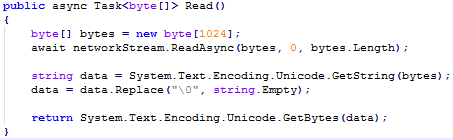
Metoda *ConvertXmlToByteArray* converteste un mesaj în format XML într-un sir de bytes. Aici se folosește clasa *MemoryStream* pentru a obtine o reprezentare în memorie ai acestui sir de bytes. Cu ajutorul clasei *XmlWriter* se „scrie” în memorie a acestor date de tip XML, care ulterior vor fi returnate ca un sir de bytes.



Aceasta metoda este apelata de fiecare data când un client se conecteaza la Broker. Aici se creaza un mesaj în format XML, cu un nod cu numele *ListaClienti*, după care, pentru fiecare utilizator înregistrat se creaza un nod cu numele *ClientID* care va conține adresa IP (cheia din dicționar) și adaugat la nodul *ListaClienti*, după toate acestea, pentru fiecare utilizator se transmite acest mesaj XML cu ajutorul metodie *write* din clasa *Networking*.



Metoda *write* din clasa *Networking* scrie un sir de bytes într-un canal de comunicare cu ajutorul metodei *WriteAsync*, aceasca operație se efectueaza asincron.



Metoda *Read* citește un sir de bytes din canalul de comunicare, acest sit poate avea o lungime pana la 1024 bytes. Citirea din canala se face asincron cu ajutorul metodei *ReadAsync* după care acest sir de bytes este converti într-un sir de caractere (string), în caz dacă sirul de bytes avea o lungime mai mica de 1024 atunci spațiu liber este populat cu caracterul „\0”, acest caracter este elimenat din sirul de caractere cu ajutorul metodei *Replace,* după care acest sir de caractere este convertit în sir de bytes de lungimea reala a mesajului și returnat.

1. Crearea diagramei use-case a aplicatiei

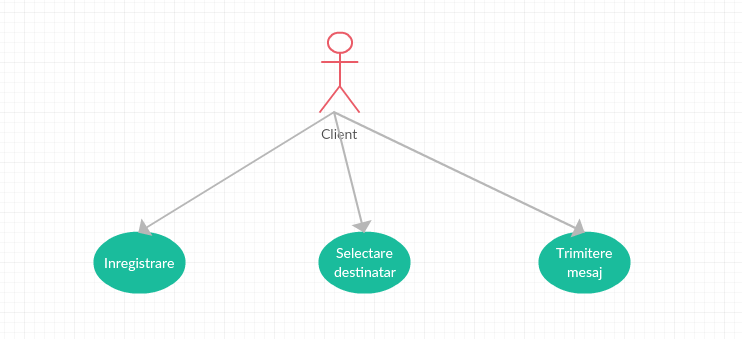


Figura1. Diagrama use-case a clientului

**Concluzie:**

În urma efectuării acestei lucrari de laborator am insușit că într-o aplicatie client –server , exista un broker care are rolul de intermediar. Acesta colectionează datele despre client și server și efectuază o legatură intre acestia doi cu ajutorul unor interfete bine definite .Se realizează conversia datelor dintr-un format in altu, acceptat de ambele părși , care comunică .