Ministerul Educaţiei, Culturii și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Departamentul Ingineria Software și Automatică

**RAPORT**

Lucrare de laborator Nr.3

Disciplina: Analiza și Modelarea Orientată pe Obiect

Tema: Realizarea diagramelor de secvenţă.

A efectuat: st.gr.TI-194 , Zavorot Daniel

A verificat : asist.univ. Rusu Cristian

Chișinău 2021

**Scopul:** Studierea noțiunii de obiect, focus control, relație sincronă și asincronă, tipuri de steriotipuri

**Sarcina:** De realizat 3-4 diagrame de secvență pentru sistemul informațional ales

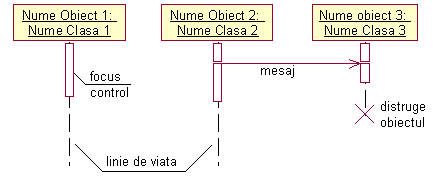
**Considerații teoretice**

În limbajul UML colaborarea între entităţi se cercetează în aspectul informativ al comunicaţiilor lor, adică obiectele care interacţionează între ele fac un oarecare schimb de informaţii. Pentru modelarea colaborării între obiecte în limbajul UML se utilizează diagramele de interacţiune. Vorbind despre aceste diagrame se iau în consideraţie două aspecte.

Mai întîi, colaborarea între obiecte poate fi cercetată în timp şi atunci pentru reprezentarea particularităţilor temporale şi modului de acceptare a mesajelor se utilizează diagrama de secvenţă.

În al doilea rînd pot fi cercetate particularităţile structurale ale colaborării între obiecte. Pentru reprezentarea particularităţilor de transmitere şi acceptare a mesajelor între obiecte se utilizează diagrama de colaborare.

În diagrama de secvenţă se descriu numai obiectele care sunt direct implicate în interacţiune şi nu reflectă asocierile statice cu alte obiecte. Pentru diagrama de secvenţă momentul principal este dinamica colaborării între obiecte în timp. Grafic fiecare obiect se reprezintă printr-un dreptunghi şi se plasează în partea de sus a ciclului său de viaţă (fig. 54). În înteriorul dreptunghiului se indică numele obiectului şi numele clasei despărţite prin două puncte. Totodată toată înregistrare se subliniază, ce indică că obiectul este exemplarul unei clase. În caz dacă numele obiectului lipseşte, atunci se indică numai numele clasei şi obiectul se consideră anonim.



**Fig. 55.** Primitivele grafice ale diagramei de secvenţă.

Obiectul din stînga diagramei este cel care iniţiază colaborarea (obiectul 1 în fig. 55). Mai la dreaptă se reprezintă obiectul care interacţionează cu primul. Totate obiecte în diagrama de secvenţă formează o anumită ordine determinată de activitatea colaborării lor.

A două măsură a diagramei de secvenţă este axa verticală (de sus în jos). Momentului iniţial de timp îi corespunde partea de sus al diagramei. Totodată colaborarea obiectelor este realizată prin mesajele transferate. Mesajele se reprezintă sub forma de săgeţi drepte cu numele mesajelor, ele de asemenea sunt într-o ordine anumită în timp. Cu alte cuvinte, mesajele plasate în diagrama de secvenţă mai sus sunt iniţiate mai devreme decît cele din jos. Totodată proporţiile pe axa temporală nu se indică fiindcă diagrama de secvenţă modelează doar ordonarea în timp a legăturilor de tip «mai devreme–mai tîrziu».

### Linia de viaţă al obiectului

*Linia de viaţă a obiectului* (object lifeline) se reprezintă printr-o linie verticală punctată asociată cu un singur obiect în diagrama de secvenţă. Linia de viaţă defineşte intervalul de timp în care obiectul există şi interacţionează cu sistemul dat.

Obiecte aparte, după finisarea activităţiilor sale în sistem, pot fi distruse pentru eliberarea resurselor allocate lor în sistemul priectat. Pentru aceste obiecte linia lor de viata «se rupe» în momentul de distrugere. Pentru reprezentarea momentului de distrugere al unui obiect în limbajul UML se utilizează un simbol special sub forma de litera latină «X». Mai jos de acest simbol, linia punctată nu poate fi desenată fiindcă obiectul în sistemul deja nu este şi acest obiect este exclus din toate interacţiunile ulterioare.

Nu este obligatoriu a crea obiecte la momentul iniţial de timp. Obiecte pot fi create la necesitate, economisind resursele acestui sistem şi majorînd randamentul lui. În acest caz dreptunghiul obiectului respectiv se desenază în partea diagramei care corespunde momentului de creare a obiectului.

## Focus control

În procesul de funcţionare a sistemelor OO unele obiecte pot fi în stare activă executînd acţiuni anumite sau pot fi pasive aşteptînd mesaje de la alte obiecte. Pentru a evidenţia obiectele active în limbajul UML se utilizează notaţia specială – *focus control* (focus of control). Focus control se reprezintă în formă de dreptunghi subţire, latura de sus a căruia indică (reflectă) începutul activităţii, latura de jos – finisarea activităţii (focusului de control).

În unele cazuri ca iniţiator al activităţii în sistem poate fi un actor sau utilizatorul extern. În acest caz actorul se desenează primul din stînga cu focus control respectiv. Totodată actorul poate să aibă un nume propriu sau să rămînă anonim.

## Mesaje

Scopul colaborării în contextul limbajului UML constă în specificarea comunicaţiei între o mulţime de obiecte. Fiecare interacţiune este descris de un set de mesaje cu care obiectele-participante se schimbă intre. În acest sens *mesajul* (messaje) reprezintă un fragment de informaţie care este transferat de către un obiect altuia. Totodată acceptarea mesajului iniţializează anumite acţiuni îndreptate spre rezolvarea problemei de către obiectul căruia mesajul îi este transferat.

Mesajele nu numai transmit informaţia, ci şi presupun executarea acţiunilor aşteptate de către obiectul acceptabil. Mesajele pot iniţia executarea operaţiilor de către obiectul unei clase, dar parametrii acestor operaţii sunt transferaţi împreună cu mesajul. În diagrama de secvenţă toate mesajele sunt coordonate după timpul de apariţie în sistemul modelat.

În acest context, fiecare mesaj are o direcţie de la obiect, care iniţiază şi trimite un mesaj la obiectul care le primeşte. Uneori, expeditorul mesajului se numeşte client şi beneficiar - server. Totodată mesajul de la client este sub formă de iniţializare a unui anumit serviciu, iar reacţia aşa numitului server după primirea mesajului presupune executarea anumitor acţiuni şi transmiterea informaţiei sub forma a unui mesaj către client.

În limbajul UML pot fi folosite mai multe tipuri de mesaje, fiecare dintre ele are propria reprezentare grafică.

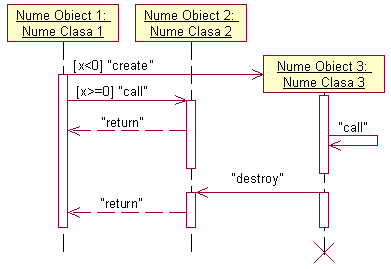
În unele cazuri, obiectele pot transmite mesaje sie însuşi, iniţiind aşa-numita comunicare reflexivă.

### Stereotipuri de mesaje

În limbajul UML sunt presupuse numai acţiuni standarde, care se execută la primirea mesajului respectiv. Ele pot fi indicate în diagrama de secvenţă sub forma de stereotipuri ataşate mesajelor respective şi se scriu în ghilimele. Pentru diagrama de secvenţă sunt următoarele stereotipuri de mesaje:

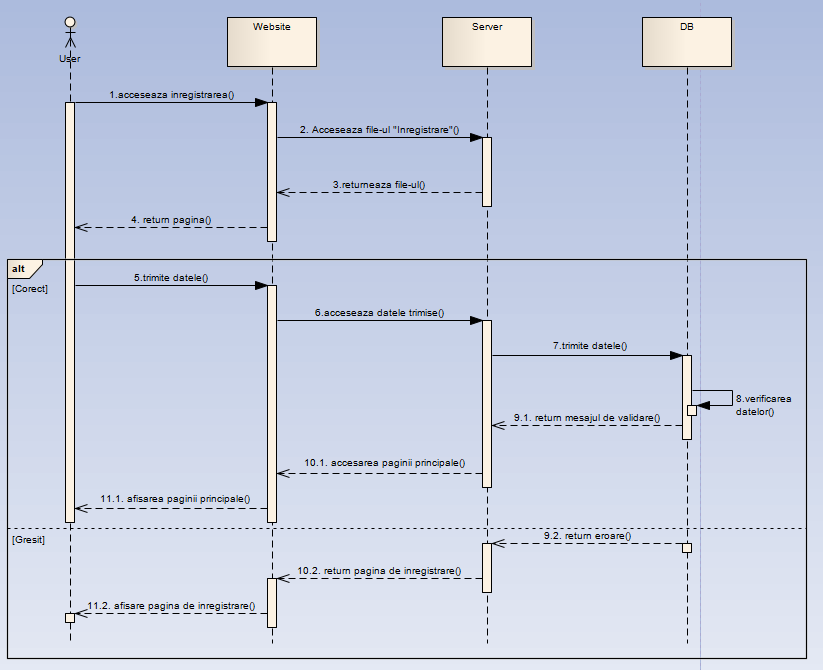
* "call" – invocă chemarea operaţiei sau procedurei obiectului-destinatar;
* "return" – returnează valoarea operaţiei executate sau procedurei obiectului apelant;
* "create" – crează un alt obiect pentru executarea anumitor acţiuni;
* "destroy" – mesaj cu o cerinţă clară de a distruge obiectul corespunzător.Se transmite în cazul când este necesar pentru a opri acţiunile nedorite a obiectului din sistemul existent, sau atunci când obiectul nu mai este necesar şi ar trebui să elibereze resursele alocate lui;
* "send" – trimiterea unui semnal care este iniţializat asincron de către un obiect şi este acceptat de altul. Diferenţa între un semnal şi un mesaj constă în fapt că semnalul trebuie să fie descris în clasa obiectul căruia iniţializează transmiterea lui.

În afară de stereotipuri, mesajul poate avea denumirea sa proprie, a operaţiunii. În acest caz lîngă săgeată se scrie numele operaţiei cu paranteze rotunde în care se indică parametrii sau argumentele operaţiei respective. Dacă parametrii lipsesc atunci parantezele rămîn neapărat după numele operaţiei.



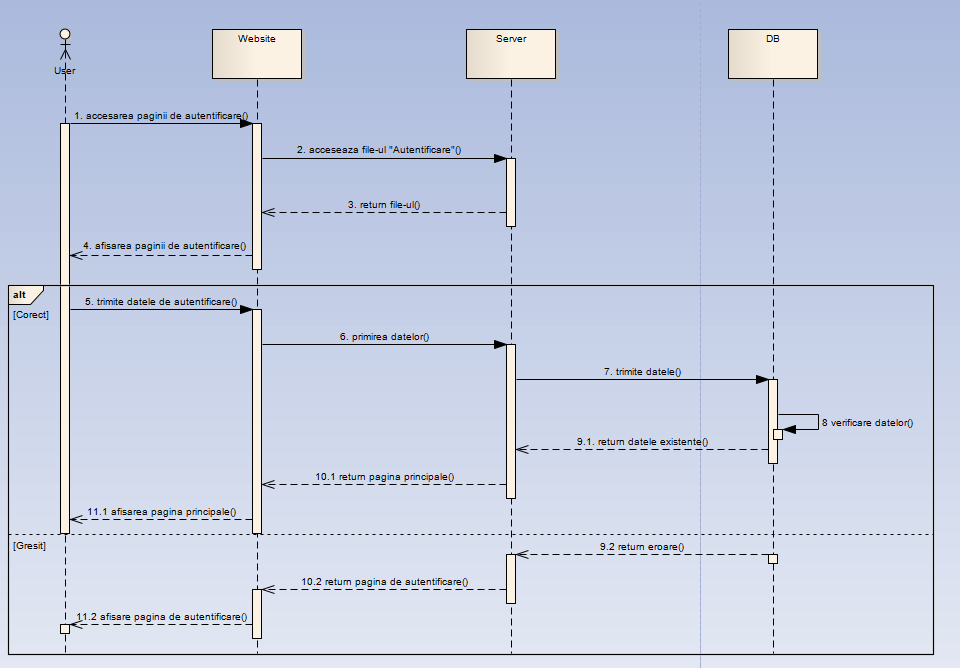
**Fig. 56.** Notaţiile mesajelor in diagrama de secvenţă.

**Implementare, rezultate practice:**



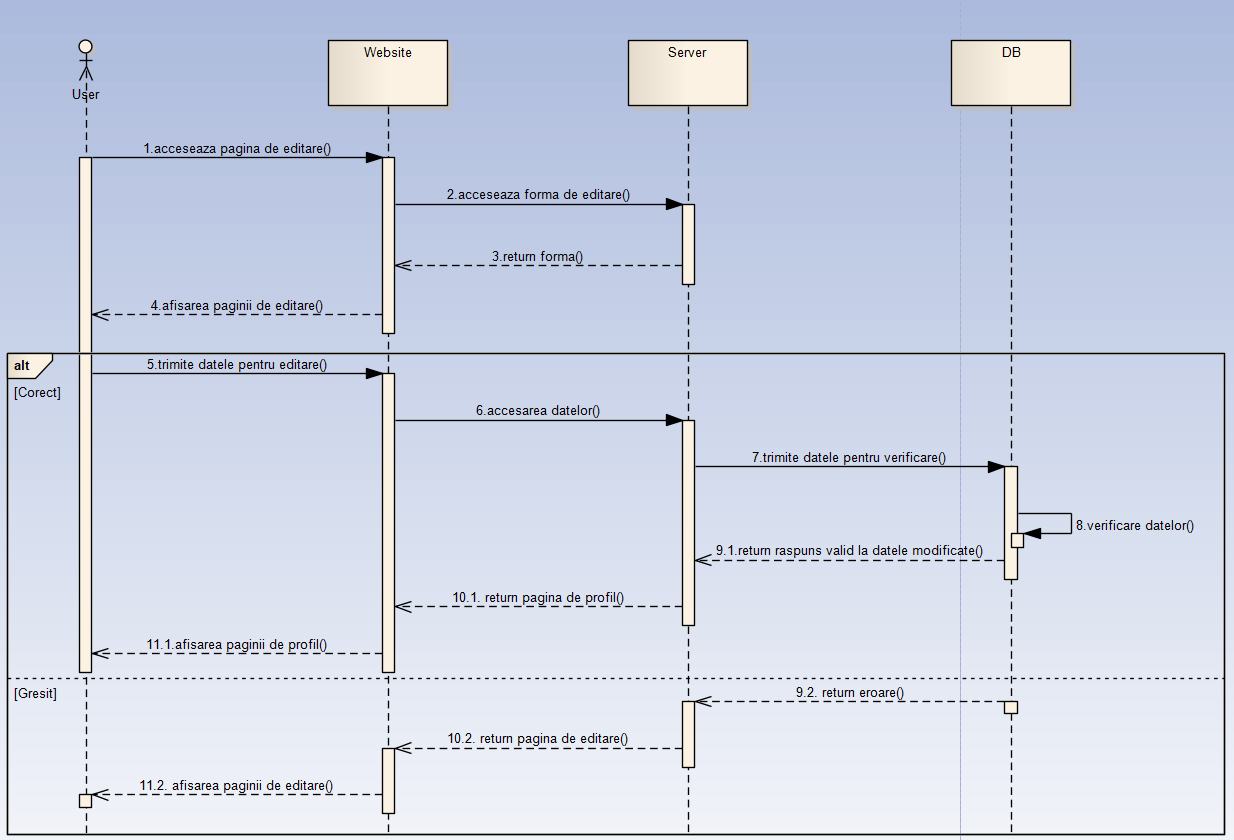
***Figura 1.*** Diagrama de secventa a procesului de inregistrare

In figura data este reprezentata diagrama de secventa a procesului de inregistrare. In sistema data putem observa ca mesajele sunt de tip *call* si *return*. Mesajul de tip *call* este utilizat pentru a invoca chemarea unei operatiuni(transmitere date, accesare forma inregistrare). Mesajul de tip *return* returnează valoarea operaţiei executate, in cazul dat daca inregistrarea a fost cu succes. In aceasta diagrama se mai utilizeaza *Fragment* de tip *alt*(2 variante). In acest fragment procesul are sa decurga in 2 moduri diferite in dependenta de rezultatul obtinut dupa verificarea datelor introduse. In cazul in care datele sunt corecte utilizatorul urmeaza sa aleaga o data si sa procedeze mai departe iar in cazul datelor gresite utilizatorul urmeaza sa reintroduca datele.



***Figura 2.*** Diagrama de secventa a procesului de logare

In figura 2 este reprezentata diagrama de secventa a procesului de logare. Ca si in *Figura 1* mesajele utilizate sunt *call* si *return* care sunt utilizate pentru a afisare ca este executata o actiune de invocare a ceva si este primit un raspuns. In aceasta diagrama se utilizeaza *Fragment* de tip *alt*(2 variante). In acest fragment procesul are sa decurga in 2 moduri diferite in dependenta de rezultatul obtinut dupa verificarea datelor introduse. In cazul in care datele sunt corecte utilizatorul urmeaza sa aleaga o data si sa procedeze mai departe iar in cazul datelor gresite utilizatorul urmeaza sa reintroduca datele.



***Figura 3.*** Diagrama de secventa a procesului de editare a profilului

In figura 3 este reprezentata diagrama de secventa a procesului de editare a profilului. Utilizatorul urmeaza sa introduca datele care doreste sa le modifice, iar daca datele sunt importnate o sa fie obligat sa introduca parola curenta. Ca si in figurile precedente se utilizeaza mesaje de tip *call*, *return* si *Fragment* de tip *alt*.

**Concluzie:**

În această lucrare de laborator am studiat noțiunile de obiect, focus control, relație sincronă și asincronă, tipuri de stereotipuri. În urma cunoștințelor obținute am realizat sarcina acestui laborator, și anume să creăm 4 diagrame de secvență pentru sistemul informațional ales: “Analiza și modelarea aplicației de monitorizare a activității unui jucător”. În crearea diagramelor am folosit limbajul UML și aplicația Enterprise Architect.

**Bibliografie:**

1. Diagrama de secvență [Resursă electronică] https://sites.google.com/site/uml4students/diagramele-de-secventa
2. Descrierea sistemului de operare Windows [ Resursă electronică] <http://www.ctalicuza.ro/pluginfile.php/826/mod_resource/content/1/SO_Windows%20.pdf>
3. Tipuri de fișiere [ Resursă electronică]

https://totuldesprecalculatoare.weebly.com/tipuri-de-fisiere.html