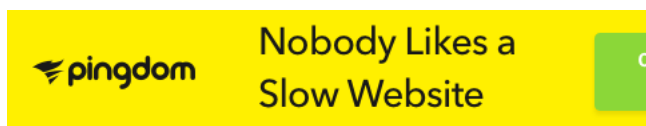


# SITE-UL MEU

ACASA   MATEMATICA   C#   SQL SERVER 2012   UML   APLICATII WINDOWS PH...   EXERCITII C++   SISTEME DE OPERARE   PROPRIUL SO   SO2



Bine ați venit, **Vizitator!**  
Înregistrare | Autentificare

AUTENTIFICARE



PRIETENII SITE-ULUI

- [Creați un website gratuit](#)
- [Forumul Comunității uCoz](#)
- [Manualul uCoz](#)
- [Tutoriale Video](#)
- [Magazinul oficial cu Șabloane](#)
- [Cele mai bune exemple de website-uri](#)

STATISTICI

Total online: **1**  
Vizitatori: **1**  
Utilizatori: **0**

## 1. Diagrama de stari

- Pentru modelarea comportamentului la nivel logic in UML pot fi utilizate la rind citeva diagrame canonice: de stari, de activitate, de secventa si de cooperare.
- Diagrama de stari descrie procesul de modificare a stariilor nuami pentru o clasa, pentru un exemplar al clasei.
- Este folosita pentru descrierea consecutivitatii de stari posibile trecerilor, care in ansamblu caracterizeaza comportamentul elementelor modelului in timpul ciclului de viata.
- Reprezinta comportamentul dinamic al entitatii in baza specificatiei reactiei lor la perceperea caror-va evenimente concrete.
- In realitate este un graf de infatisare speciala, care reprezinta un automat. Virfurile acestui graf sunt stările si alte elemente ale automatului.

## 2. Automatul (state machine)

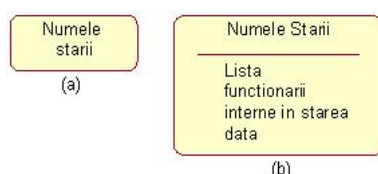
- In UML reprezinta o formalizare pentru modelarea comportamentului elementelor modelului si a sistemului intreg.
- In metamodelul UML automatul este un pachet, in care sunt definite o multime de definitii, necesare pentru reprezentarea comportamentului entitatii modelate in forma de spatiu discret cu un numar finit de stari si treceri
- Fiecare diagrama de stari reprezinta un automat



- Notiunile de baza care intra in formalizmul automatului sunt starea si trecerea.

## 3. Stare

- In UML starea este subinteleasa ca metaclassa abstracta, ce se utilizeaza pentru modelarea situatiei apartate, pe parcursul carei este prezenta executarea anumetei conditii.
- Starea poate fi in forma de valori concrete a atributului clasei sau obiectului, in acest caz modificarea anumitelor valori va respinge modificarea clasei modelate sau opbiectului
- Este reprezentata ca dreptunghi cu vifurile rotungite, care poate fi despartit in doua sectiuni



## 4. Numele starii

- Reprezinta aliniat de text, care dezvaluie sensul starii date. Este scris intotdeauna cu majuscula
- Este recomandat de folosit in calitate de nume verbele in timpul prezent (suna, asteapta) sau (ocupat, liber)
- Numele starii poate lipsi (stare anonima)

## 5. Starea initiala

- Reprezinta un caz particular de stare, care nu contine nici o actiune internă (pseudostare).
- Este utilizata pentru indicarea pe diagrama de stari a spatiului grafic, de la care incepe procesul de modificare a stariilor.
- Este reprezentata in forma de cerc hasurat de la care poate esi numai sageata corespunzatoare cu trecerea



- La cel mai înalt nivel de reprezentare a obj. trecerea de la starea initiala la starea finala poate fi marcata ca actiunea de creare (initializare) a obj. dat. In caz contrar tranzitia nu este marcata de loc (este prima trecere in starea urmatoare).

## 5. Starea finala

- Reprezinta un caz particular al starii, care nu contine nici o actiune internă
- In aceasta stare obiectul se va afla in starea initiala dupa finisarea lucrului automatului in ultimul moment de timp

## 6. Tranzitie

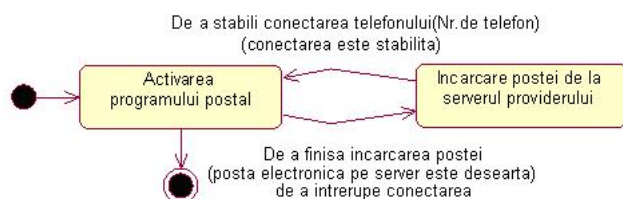
- O simpla tranzitie reprezinta o relatie intre doua stari consecutive indicind faptul schimbarii a unei stari cu alta.
- Incepe cind un anumit eveniment se petrece: terminarea executarii actiunii, trimiterea mesajului sau emiterea semnalului.
- In trecere se indica numele actiunii
- Este reprezentata ca o linie neintrerupta cu sageata, indreptata in starea de tinta.
- Se marcheaza cu aliniat de text: <signatura evenimentului>['<conditia de pază>'] <exprimarea actiunii>.

## 7. Eveniment (event)

- Este un element aparte al UML
- Reprezinta specificatia anumitui fapt, care are atasata o locatie in timp si in spatiu
- Joaca rol de stimulare, care initializeaza treceri de la o stare in alta
- In calitate de evenimente distingem semnale, apeluri, terminarea intervalurilor fixate de timp.
- Numele evenimentului identifica fiecare tranzitie aparte in diagrama de stari si pot contine aliniat de text, care incepe cu minuscula.
- Daca alaturi de sageata trecerii nu este indicat numele, atunci trecerea este netriggerata
- Dupa numele evenimentului pot urma paranteze rotunjite pentru parametrii evenimentului-trigger.

## 8. Conditie garda

- **daca exista, atunci intotdeauna este scrisa in paranteze dreptughiulare dupa evenimentul-trigger si reprezinta expresie booleana**

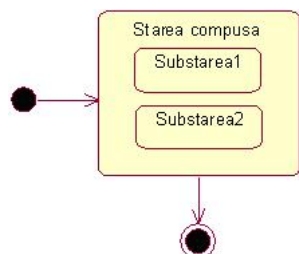


## 9. Expresia actiunii (action expression)

- se executa doar atunci cind se executa tranzitia
- Reprezinta operatia atomica, care se executa dupa efectuarea tranzitiei respective inainte de oricare actiune in stare obiectiva
- Expresia actiunii poate contine o lista intreaga de activitati,separate prin <<;>>

## 10. Stare si substare compusa

- Este o stare alcătuita din alte stari, care vor fi substari pentru primul element

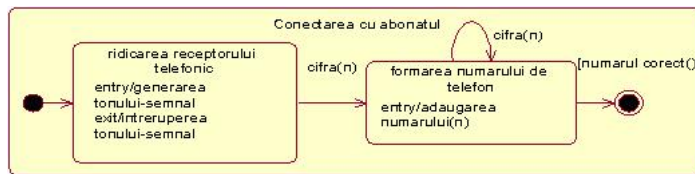


- Poate contine doua sau mai multe subautomate sau citeva sub-stari consecutive
- O substare pote fi stare compusa pentru alte substari

## 10. Substari disjuncte

- Sunt utilizate pentru modelarea comportamentelor obj. in timpul caruia in fiecare moment obiectul poate fi numai intr-o stare.

Comportamentul obj. in acest caz reprezinta scimbarea disjuncta a substarilor, incepind cu starea initiala si terminind cu cea finala.



## 11. Substari concurente

- pot specifica doua sau mai multe subautomate, care pot executa paralel in cadrul starii compuse.
- Fiecare subautomat ocupa o regiune in cadrul starii compuse, care este despartita de altele cu linia orizontala intrerupta

