

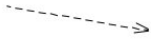



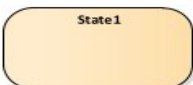
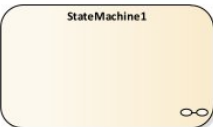

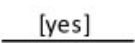
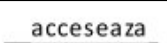





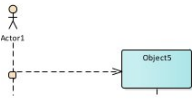
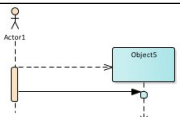
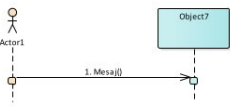
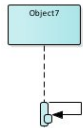



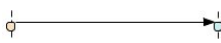
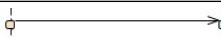
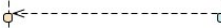
## Regulile sintaxei **Diagrama Componentelor**

Entități de bază	
	<b>Componentul</b> – reprezintă o entitate care specifică o parte a oricărui sistem (aplicație). Denumirea componentului întotdeauna va fi un substantiv.
	<b>Interfața</b> – reprezintă partea inițială a oricărui sistem (aplicație). Denumirea interfeței va începe întotdeauna cu litera „I” (ex. IBancomat, ISistemOperare).
Tipuri de Relații și Sintaxa diagramei componentelor	
	<b>Dependența</b> – arată corespondența dintre 2 sau mai multe componente.
	<b>Realizarea</b> – întotdeauna este utilizată pentru a conecta Componentul cu Interfața.



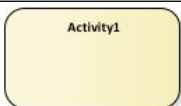

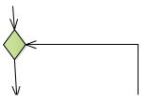
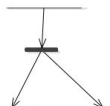
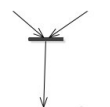
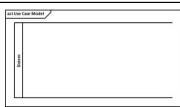

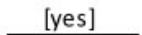

## Regulile sintaxei **Diagrama stărilor**

Entități de bază	
	<b>Punctul inițial</b> – reprezintă începutul unei diagrame. O diagramă poate avea doar un singur punct inițial
	<b>Punctul final</b> – reprezintă sfârșitul unei diagrame. O diagramă poate avea unul sau mai multe puncte finale, atunci când sunt mai multe puncte finale, acestea se numesc cu denumiri diferite.
	<b>Starea</b> – este o etapă (parte) din procesul de lucru al sistemului (aplicației). Denumirea stării va fi întotdeauna un verb la infinitiv (ex. Logare, Conectare, Deconectare, Salvare) sau un participiu (Conectat, Deconectat, Eliberat, Achitat, Actualizat).
	<b>Stare compusă</b> – reprezintă o stare compusă din mai multe substări. Denumirea stării compuse respectă aceeași regulă ca și denumirea stării simple. Starea compusă poate fi de 3 tipuri: a) stare compusă cu substări concurente (paralele) b) stare compusă cu substări depuse c) stare compusă cu substări disjuncte
Tipuri de Relații și Sintaxa diagramei componentelor	
	<b>Tranziția</b> – reprezintă relația dintre două stări.
	<b>Condiția de gardă</b> – este o informație scrisă între paranteze patrate. Atunci când condiția este îndeplinită, sistemul își continuă procesul de lucru, în caz contrar fie că se repetă procesul, fie că se întrerupe lucrul sistemului (aplicației).
	<b>Comentariu</b> – este un mesaj scris pe relația de tranziție. Are drept scop explicarea mai detaliată a procesului trecerii sistemului dintr-o stare în alta.




## Regulile sintaxei **Diagrama Secvențelor**

Entități de bază	
	<b>Actorul</b> – reprezintă elementul extern care inițiază acțiunile asupra sistemului. În diagrama secvențelor putem avea unul sau mai mulți actori. Denumirea actorului întotdeauna va fi un substantiv.
	<b>Obiectul</b> – este o instanță (un exemplar) a unei clase. Obiectul trebuie să reprezinte părți importante din sistem: meniuri, submeniuri (Ex. Interfața, Sistem, BD, Pagina Start, etc.). Denumirea obiectului întotdeauna va fi un substantiv.
	<b>Fragmentul</b> – permite parcurgerea diagramei atât în situații de succes cât și insucces, indicând care parte din diagramă se repetă. Denumirea fragmentului trebuie gândită sub forma unei condiții.
	<b>New Object</b> – este un obiect nou creat de un obiect cu putere de prelucrare, de obicei apare în procesul de lucru al sistemului (exemplu de New Object: Catalog, Formular, Chestionar, Bilet).
	<b>Delete Object</b> - reprezintă operația de ștergere a obiectului nou creat. Ștergerea va fi efectuată de către obiectul care l-a creat.
	<b>Mesajul</b> – reprezintă informația care se transmite între Actor-Obiect și Obiect-Obiect. Mesajul se numerează și poate conține un număr diferit de cuvinte.
	<b>Apel de întoarcere la obiect</b> – se utilizează pentru a face anumite verificări, prelucrări de date de către singur obiectul.
Stereotipuri	
	<b>Boundary</b> – poate fi utilizat pentru a reprezenta Interfața sistemului.
	<b>Control</b> – poate fi utilizat pentru a reprezenta Sistemul (aplicația).
	<b>Entity</b> - poate fi utilizat pentru a reprezenta BD a sistemului.
Tipuri de Relații și Sintaxa diagramei componentelor	
	<b>Relația sincronă</b> – este o relație care așteaptă răspuns imediat de la sistem, sau peste o anumită perioadă de timp.
	<b>Relația asincronă</b> – este o relație care nu așteaptă răspuns de la sistem.
	<b>Relația return</b> – este o relație care servește drept răspuns pentru o relație sincronă.











## Regulile sintaxei **Diagrama activităților**

Entități de bază	
	<b>Punctul inițial</b> – reprezintă începutul unei diagrame. O diagramă poate avea doar un singur punct inițial
	<b>Punctul final</b> – reprezintă sfârșitul unei diagrame. O diagramă poate avea unul sau mai multe puncte finale, atunci când sunt mai multe puncte finale, acestea se denumesc cu denumiri diferite.
	<b>Activitatea</b> – este rezultatul trecerii sistemului într-o anumită stare. Denumirea activității este un verb conjugat (ex. Se loghează, Conectează, Deconectează)
	<b>Bloc de decizie</b> – reprezintă anumite verificări în procesul de lucru al sistemului (aplicației). Blocul de decizie întotdeauna <b>VA AVEA</b> notate condițiile de gardă.
	<b>Ramificatorul</b> – se utilizează pentru a conecta mai multe tranziții împreună. Activity2 Ramificatorul niciodată <b>NU VA AVEA</b> notate condițiile de gardă.
	<b>Fork</b> - reprezintă bara de sincronizare care ramifică două sau mai multe activități ce au loc în paralel.
	<b>Join</b> - reprezintă bara de sincronizare care așteaptă îndeplinirea a două sau Activity3 mai multor activități și le conectează împreună.
	<b>Partiția</b> – reprezintă o parte responsabilă de anumite activități în procesul de lucru al sistemului (aplicației).
Tipuri de Relații și Sintaxa diagramei componentelor	
	<b>Tranziția</b> – reprezintă relația dintre două stări.
	<b>Condiția de gardă</b> – este o informație scrisă între paranteze patrate. Atunci când condiția este îndeplinită, sistemul își continuă procesul de lucru, în caz contrar fie că se repetă procesul, fie că se întrerupe lucrul sistemului (aplicației).
	<b>Comentariu</b> – este un mesaj scris pe relația de tranziție. Are drept scop explicarea mai detaliată a procesului trecerii sistemului dintr-o stare în alta.










## Regulile sintaxei **Diagrama desfășurării (de plasare)**

Entități de bază	
	<b>Nodul</b> – este entitatea care reprezintă un anumit echipament, dispozitiv care <b>PRELUCREAZĂ</b> date. Denumirea nodului întotdeauna va fi un substantiv. (ex.Server, Calculator, Telefon).
	<b>Device</b> – este entitatea care reprezintă un anumit echipament periferic care <b>NU PRELUCREAZĂ</b> date. Denumirea acestei entități întotdeauna va fi un substantiv și va avea stereotipul <<device>> (ex.Mouse, Imprimanta, Tastatura, Scanner).
Tipuri de Relații și Sintaxa diagramei componentelor	
	<b>Asocierea</b> - întotdeauna se folosește pentru a conecta doar Nodurile.




## Regulile sintaxei **Diagrama Cazurilor de utilizare**

<b>Entități de bază</b>	
	<b>Actorul</b> - poate reprezintă atât persoana care lucrează cu sistemul cât și sistemul la general, sau anumite părți din sistem (Baza de Date). Denumirea actorului întotdeauna va fi un substantiv.
	<b>Cazul de utilizare</b> - reprezintă o funcție îndeplinită de sistem. Denumirea cazului de utilizare va fi un verb la infinitiv sau un verb conjugat. Denumirea poate să includă maxim 3 cuvinte.
	<b>Boundary</b> - are scopul de a grupa anumite funcționalități ale sistemului într-un tot întreg. Denumirea întotdeauna va fi un substantiv.
	<b>Pachetul</b> - oferă posibilitatea de a împărți sistemul în părți componente. Denumirea pachetului întotdeauna va fi un substantiv.
	<b>Note</b> - permite comentarea oricărei entități de pe diagramă.
<b>Tipuri de Relații și Sintaxa diagramei componentelor</b>	
	<b>Asocierea bidirecțională</b> - întotdeauna se folosește între Actor și Cazul de utilizare.
	<b>Dependența</b> - poate fi utilizată atât între 2 Actori cât și între 2 Cazuri de utilizare.
	<b>Generalizarea</b> - poate fi utilizată între cel puțin 3 Actori sau între cel puțin 3 Cazuri de utilizare.
	<b>Dependența cu stereotip include</b> - se folosește doar între Cazuri de utilizare. Este direcționată întotdeauna de la Cazul de utilizare de bază către Cazul de utilizare care îi urmează. Este o funcție obligatorie pentru ca sistemul să funcționeze.
	<b>Dependența cu stereotip extend</b> - se folosește doar între Cazuri de utilizare. Este direcționată întotdeauna de la Cazul de utilizare copil către Cazul de utilizare de bază, cu alte cuvinte este inversă ca direcție relației de Dependența cu stereotip include .

## Regulile sintaxei **Diagrama Claselor**

<b>Entități de bază</b>	
	<b>Clasa</b> - reprezintă o totalitate de obiecte. În calitate de clasă poate fi atât persoana ca parte abstractă, cât și sistemul la general, sau anumite părți din sistem. Denumirea clasei întotdeauna va fi un substantiv.
	<b>Interfața</b> - reprezintă partea inițială a oricărui sistem (aplicație). Denumirea interfeței va începe întotdeauna cu litera „I” (ex. IBancomat, ISistemOperare).
<b>Tipuri de Relații și Sintaxa diagramei componentelor</b>	
	<b>Asocierea</b> - întotdeauna conectează 2 sau mai multe clase între ele.
	<b>Dependența</b> - arată corespondența dintre 2 sau mai multe clase.
	<b>Generalizarea (moștenirea)</b> - poate fi utilizată între cel puțin 3 Clase.
	<b>Compoziția</b> - se folosește pentru a arăta legătura dintre Clasa întreg și clasa parte componentă, fără de care întregul NU POATE exista.
	<b>Agregarea</b> - se folosește pentru a arăta legătura dintre Clasa întreg și clasa parte componentă, fără de care întregul POATE exista.
	<b>Realizarea</b> - mereu este utilizată pentru a conecta Clasa cu Interfața.
	<b>Multiplicitate</b> - se scrie doar pentru relațiile de asociere, compoziție și agregare. Atunci când multiplicitatea este 1 la 1 pentru toate relațiile, ea nu se scrie deoarece implicit se subînțelege că este 1 la 1.

## Regulile sintaxei **Diagrama Colaborărilor**

<b>Entități de bază</b>	
	<b>Actorul</b> - reprezintă persoana care folosește sistemul (aplicația) (ex. Utilizator, Administrator). Denumirea actorului întotdeauna va fi un substantiv.
	<b>Obiectul</b> - este o instanță (un exemplar) a unei clase. Obiectul trebuie să reprezinte părți importante din sistem: meniuri, submeniuri (Ex. Interfața, Sistem, BD, Pagina Start, etc.). Denumirea obiectului întotdeauna va fi un substantiv.
<b>Tipuri de Relații și Sintaxa diagramei componentelor</b>	
	<b>Asocierea</b> – întotdeauna conectează Actor-Obiect, Obiect-Obiect.
1: <call>() →	<b>Relația sincronă</b> – este o relație care așteaptă răspuns imediat, sau peste o anumită perioadă de timp.
1: <call>() →	<b>Relația asincronă</b> – este o relație care nu așteaptă răspuns.
← 1: <call>()	<b>Relația return</b> – este o relație care servește drept răspuns pentru o relație asincronă