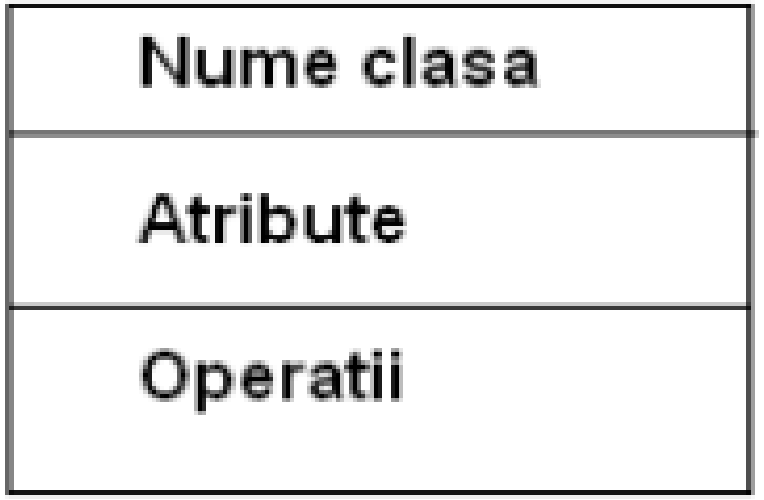
Diagrame de clase
Diagrame de componente
Diagrame de distribuție

Documentul de proiectare arhitecturală

Diagrame de clase

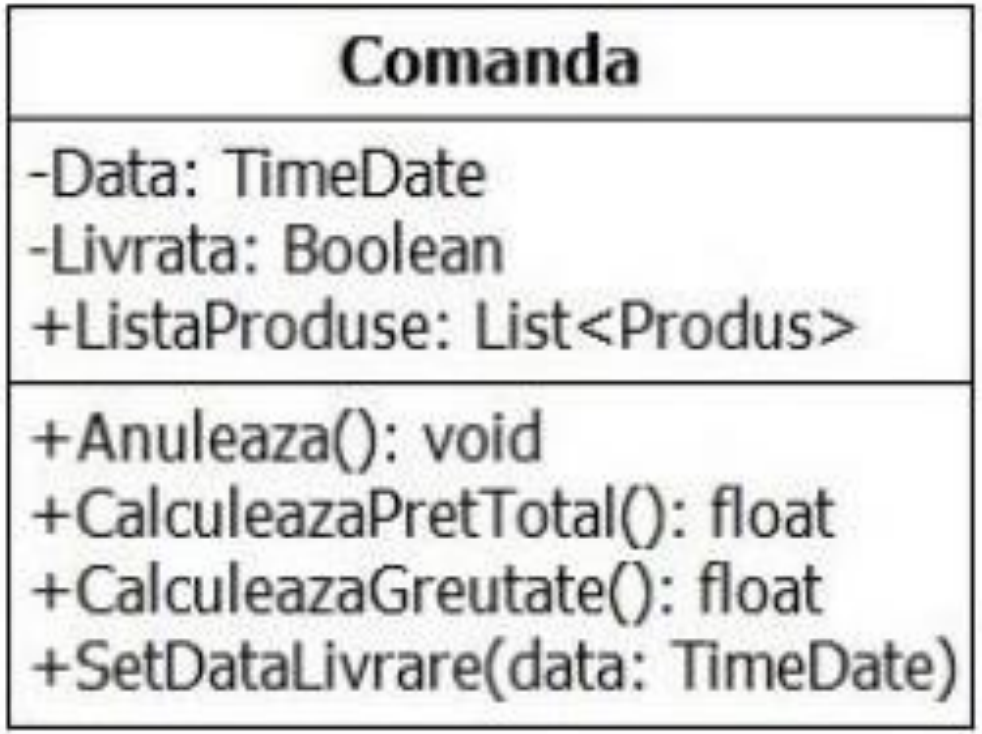
- Un grup de obiecte cu proprietăți similare și comportament comun
- Instanțiere => obiecte
- Diagrama de clase
  - Clase
  - Interfețe
  - Relații

Clase



Clase – proiectarea de detaliu

- Tipul atributelor
- Tipul valorilor intoarse de metode
- Lista de parametri a fiecărei metod
- Vizibilitatea
  - + : public
  - - : private
  - # : protected
  - ~ : package (element vizibil numai în propriul său pachet)



Clase

Atribute

- [Vizibilitate]Nume [: Tip][= Valoare inițială]
- **Vizibilitate**:
  - public, private, protected, package
  - dacă nu se specifică nicio valoare pentru vizibilitate, atributul este private
- **Nume**: denumirea atributului
- **Tip**: tipul (de date al) atributului respectiv
- **Valoare inițială**: valoarea inițială aatributului

Clase

Operații

- [Vizibilitate] Nume [(Parametri)][: Tip_întors]
- **Vizibilitate**: +, #, - sau ~
- **Nume**: denumirea operației
- **Parametri**: lista de parametri ai operației
- **Tip_întors**: poate fi un tip primitiv sau un tip definit de utilizator

Clase

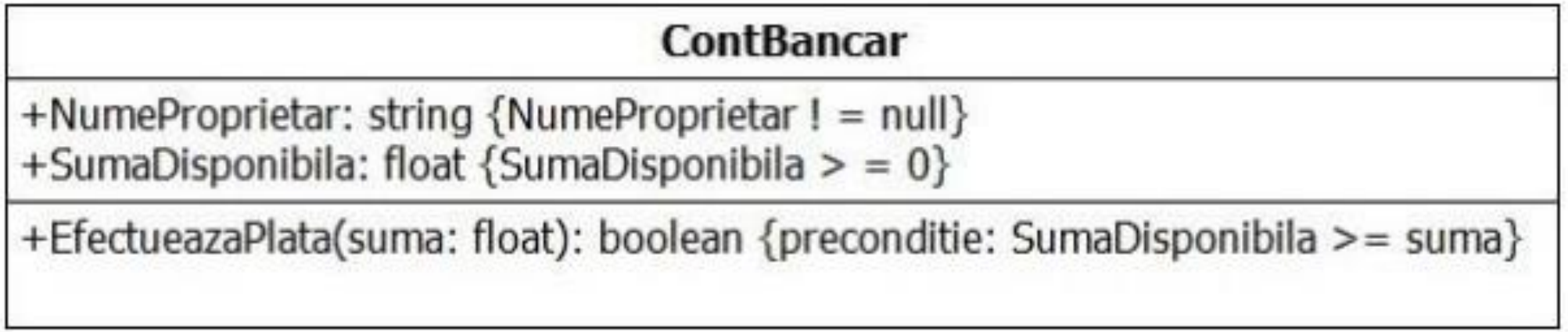
Parametri

- [Direcție]Nume : Tip [= Valoare inițială]
- **Direcție**
  - in – parametru de intrare, valoarea sa nu poate fi modificată
  - out – parametru de ieșire, valoarea sa poate fi modificată pentru a transmite informație spre apelant
  - inout – parametru de intrare a cărui valoare poate fi modificată

Clase

Constrângeri

- Se aplică unui atribut/unei operații/unei asocieri

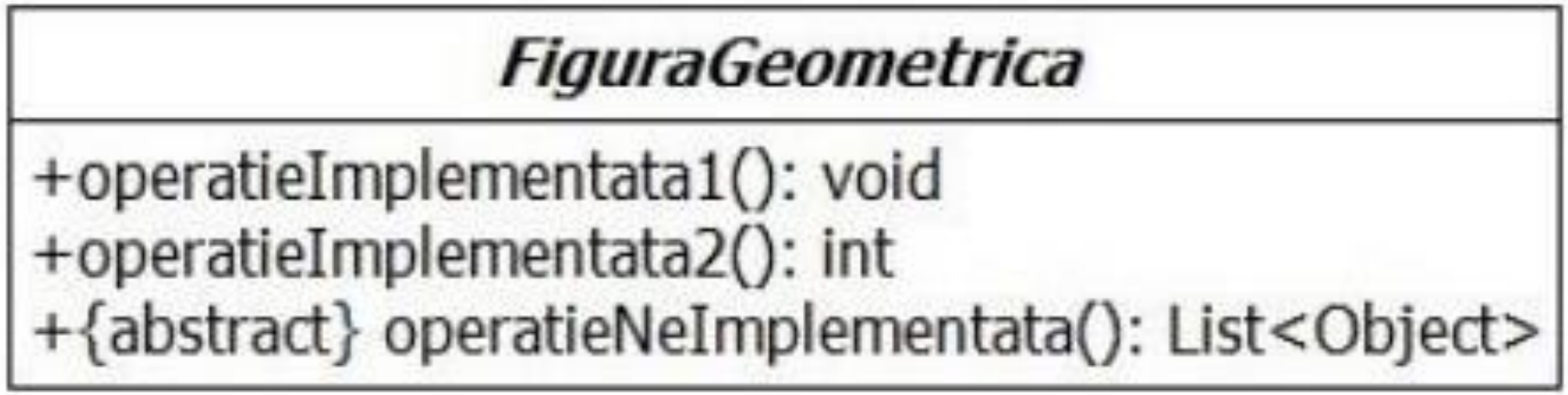


Clase abstracte

- O clasă care nu poate fi instanțiată
- Implementează o parte invariabilă de funcționalitate
- Restul implementării – clasele derivate din ea

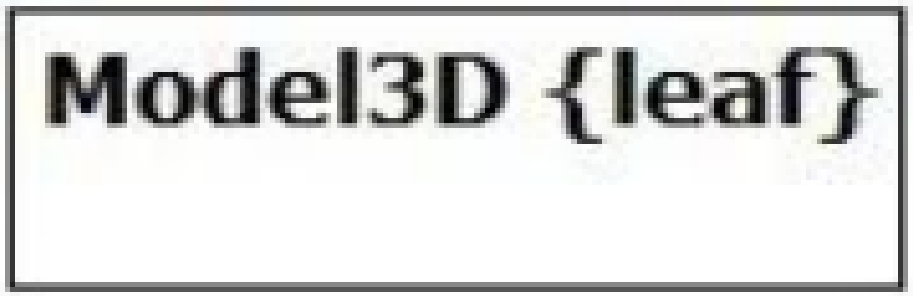
Clase abstracte

- O operație face parte dintr-o clasă abstractă și nu e implementată => font italic sau {abstract}



Clase finale

- Nu pot fi extinse/moștenite

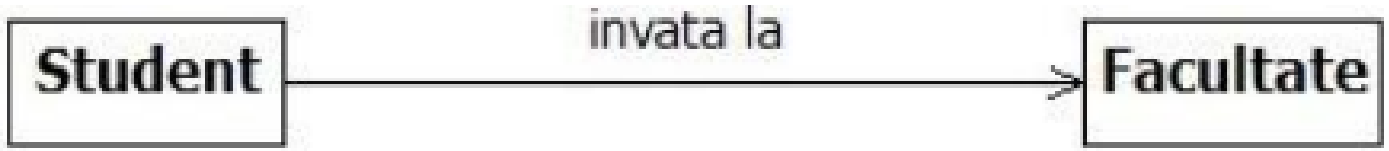


Relații între clase

- Asociere
- Agregare
- Compunere
- Dependență
- Generalizare

Asocierea

- Abstracție a unui set de legături între obiectele claselor
- **Nume**



- **Roluri**

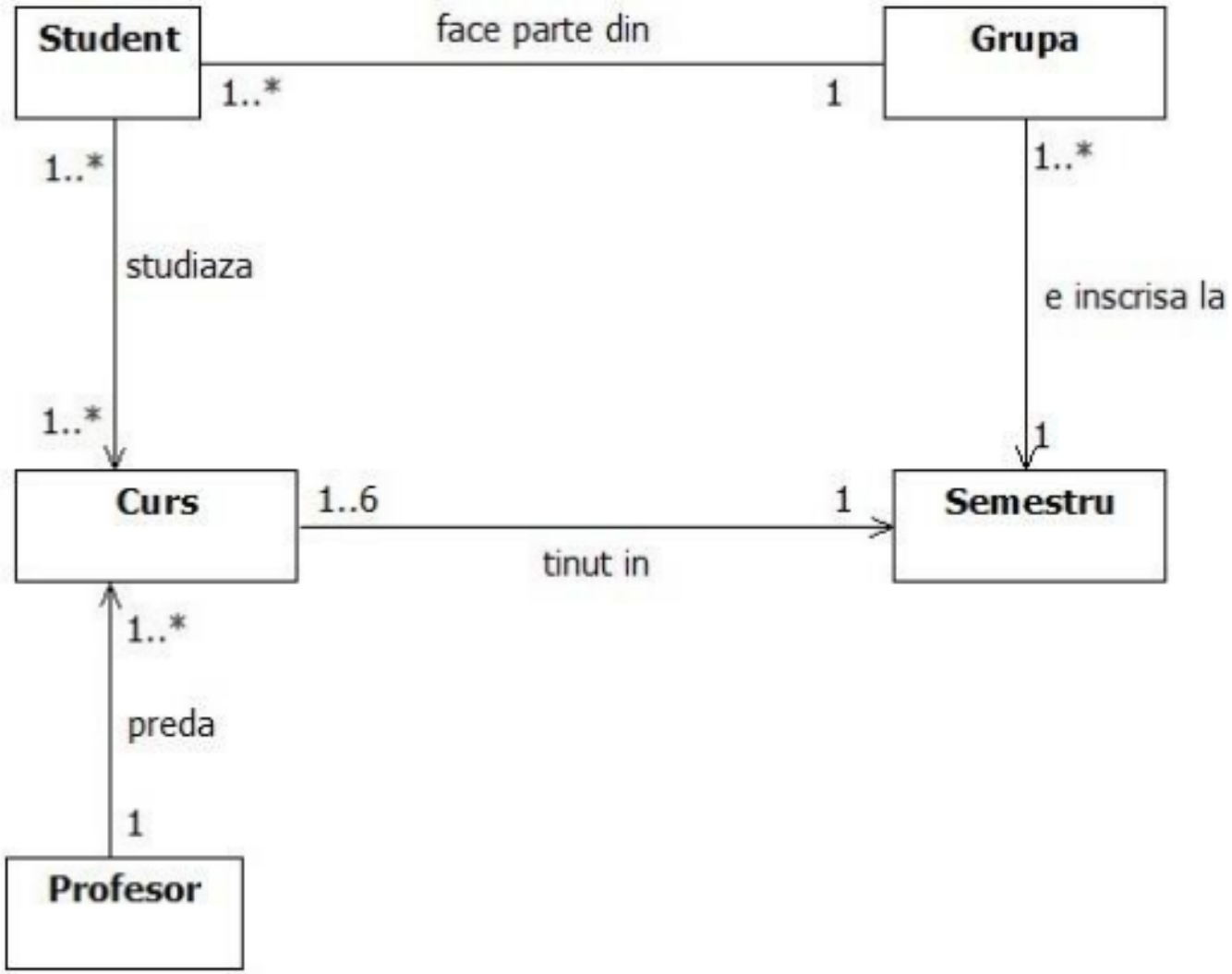
Asocierea

- **Multiplicitatea asocierilor**
  - Numărul de obiecte ce pot fi legate unui obiect al celeilalte clase, la un moment dat
  - Dacă nu e specificată – valoarea 1

  - Un singur număr (Ex.: 4, 10)
  - Un asterisc (*):cu semnificația „zero sau mai multe” instanțe
  - Un interval precizat sub formamin..max

Asocierea

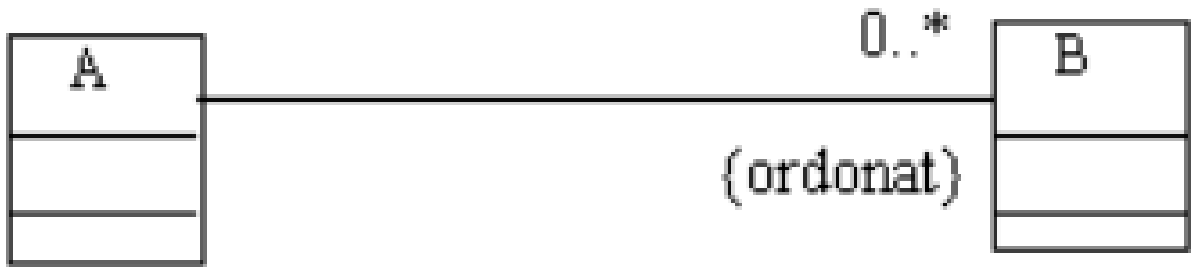
- **Multiplicitatea asocierilor**



Asocierea

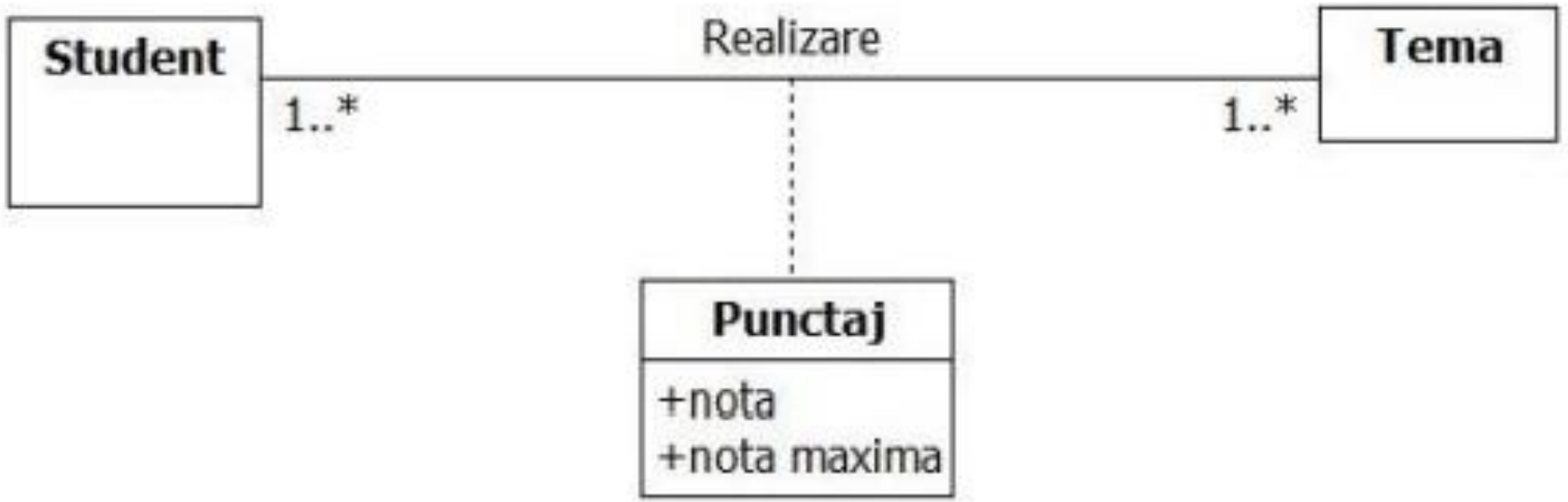
- Asocieri constrânse
  - Asocierea e valabilă numai dacă e îndeplinită condiția

  - Dacă există mai multe instanțe ale clasei B legate la o instanță a clasei A => trebuie ordonate dupa un criteriu



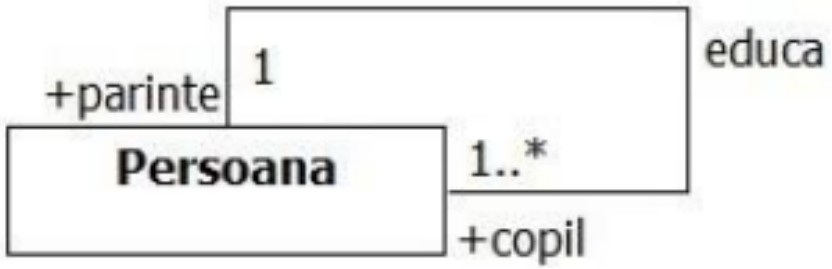
Clasa asociere

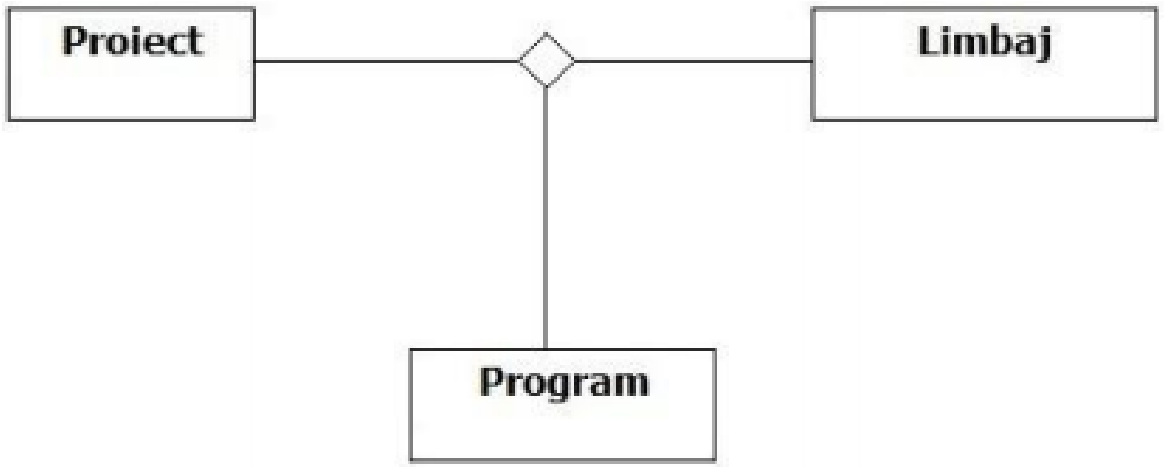
- Are și proprietăți de clasă, și de asociere



Aritate

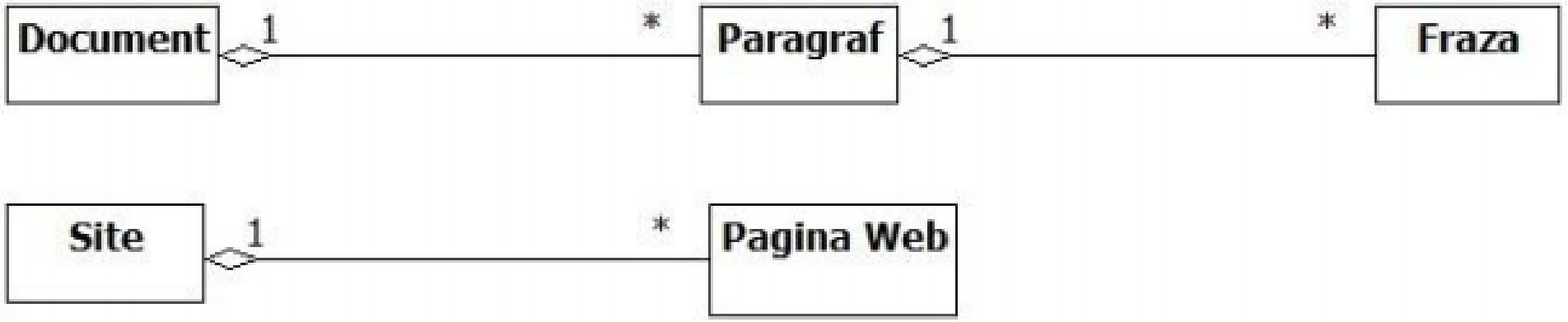
- Numărul de clase asociate prin acea asociere
- Unară
- Binară
- Ternară





Agregarea

- Semnifică cuplaj de tip compus – componenți
- O clasă e întregul, iar cealaltă componenta
- Aceleași componente pot fi incluse în mai mulți compuși
- Dacă agregatul e distrus, componentele pot exista în continuare



Compunerea

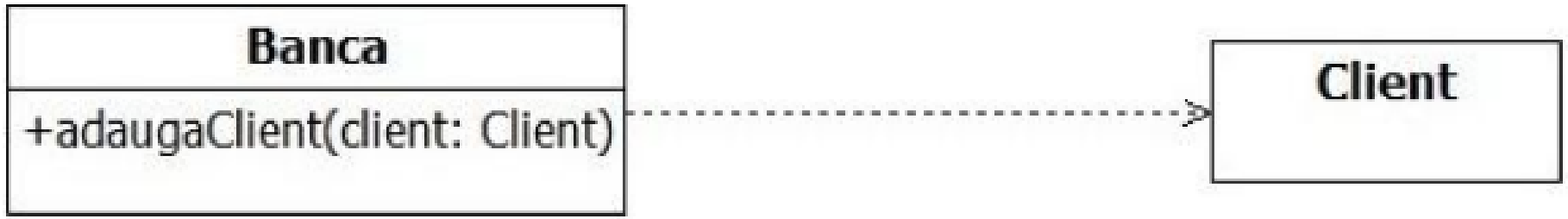
- Tip puternic de agregare
- O instanță a unei componente aparține unei singure instanțe a compusului la un moment dat
- Orice instanță a componentei nu poate exista de sine stătător
- La distrugerea agregatului se distrug și componentele sale

Dependența

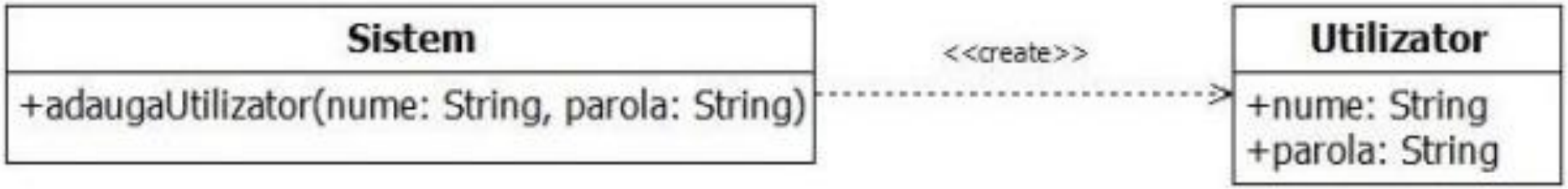
- Se utilizează când între cele două elemente nu există celelalte relații, dar ele își cunosc existența
- Când un obiect al unei clase reprezintă pentru un obiect din cealaltă:
  - O variabilă globală/locală
  - Un parametru
- În cazul apelului unei metode statice din clasa țintă
- Atunci când între clase se trimit mesaje

Dependența

- Metoda adaugăClient primește ca param o instanță a clasei Client

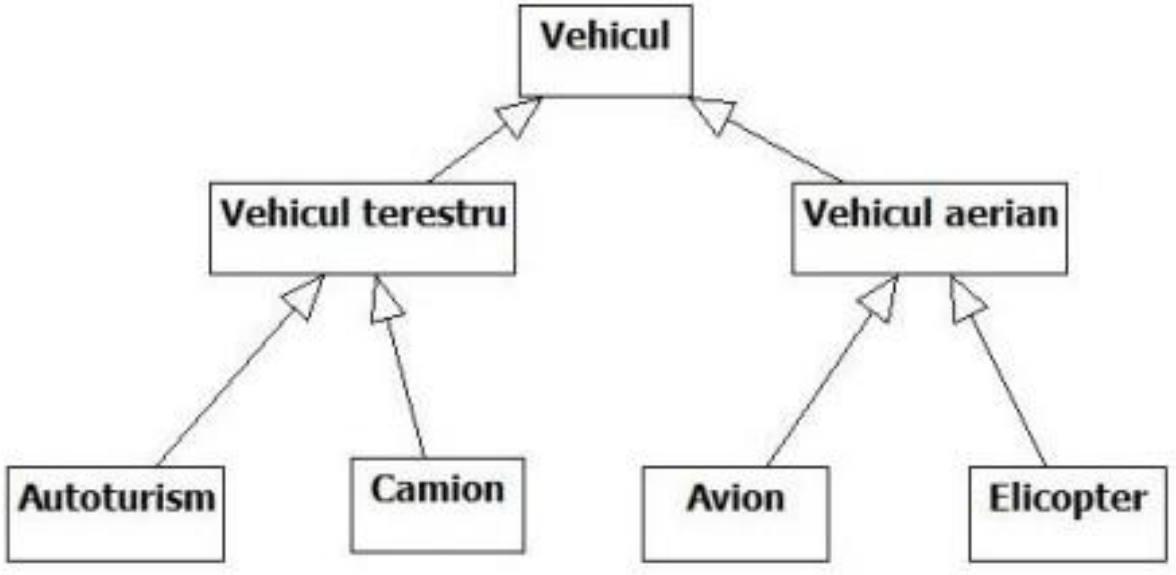


- Obiectele din clasa Sistem pot crea, la apelul metodei adaugăUtilizator, instanțe ale clasei Utilizator



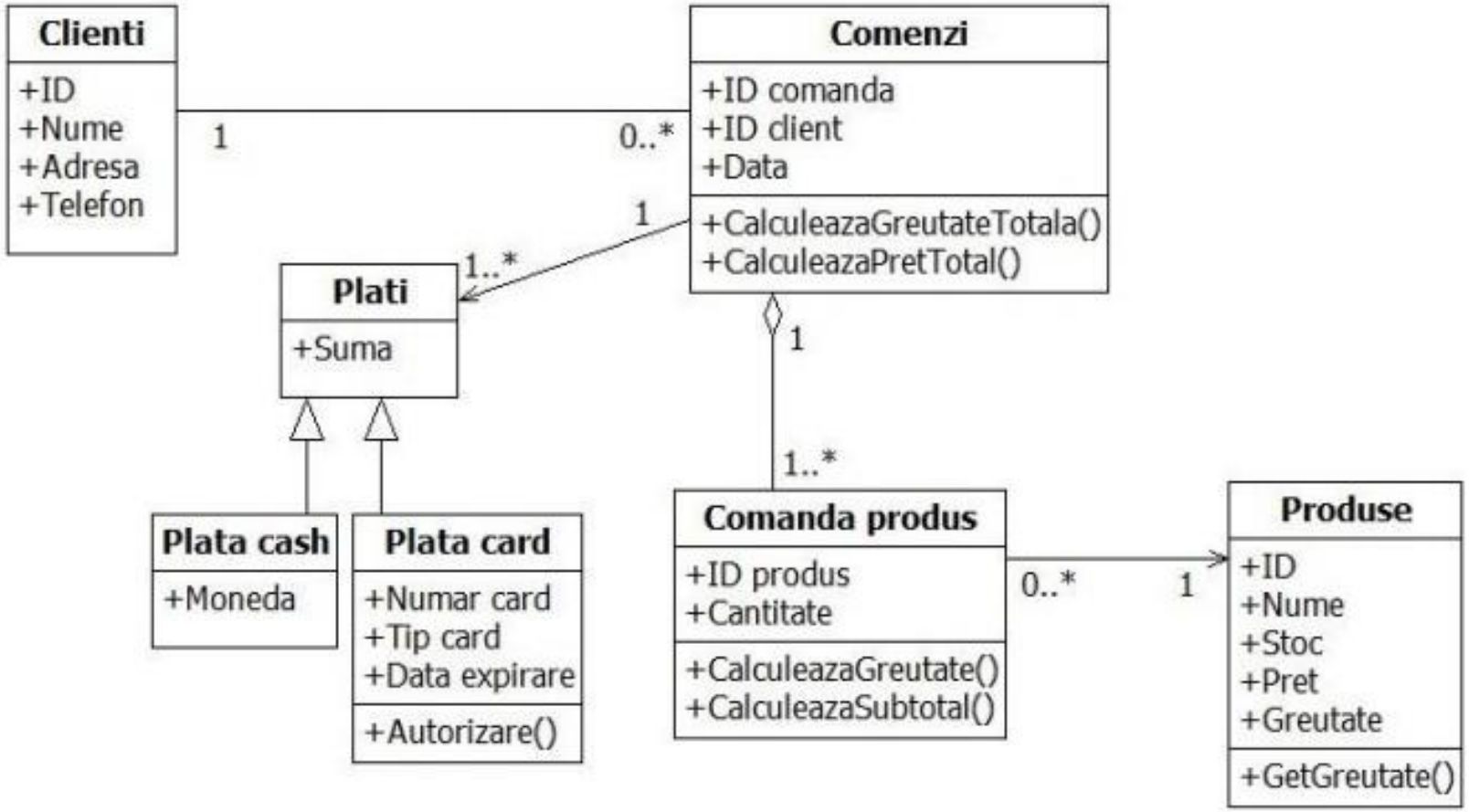
Generalizarea

- Factorizarea elementelor comune ale unui ansamblu de clase specializate (clase copii) într-o clasă mai generală (superclasă, clasă părinte)



Diagrame de clase - exemple

- Magazin online

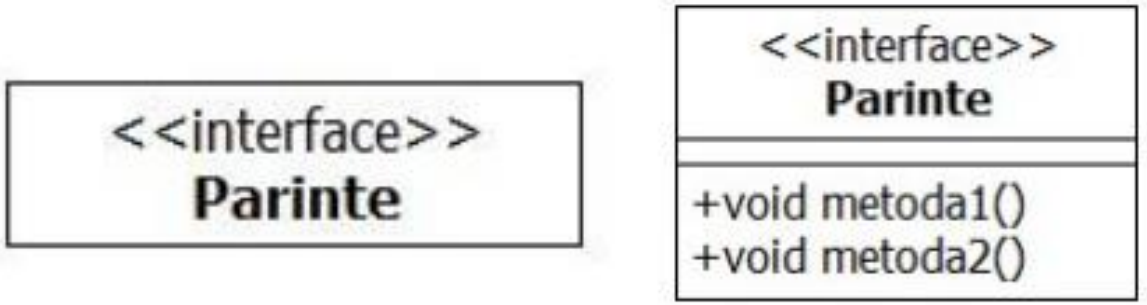


Interfețe

- Interfața: metodele publice ale clasei (fără constructor și destructor)
- Implementarea: implementările metodelor din interfață, constructorul, destructorul și câmourile clasei

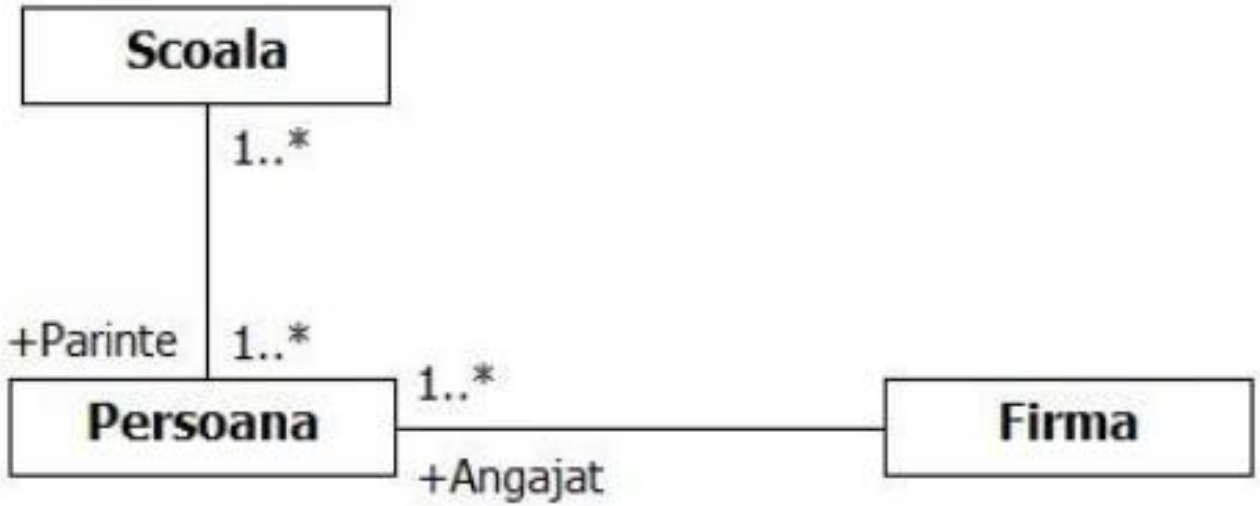
- Interfețe:
  - Un set de metode corelate care definesc o anumită comportare
  - Toate metodele publice, fără implementare
  - Nu conține variabile

Interfețe


<<interface>>
Parinte
<<interface>>
Parinte
+void metoda1()
+void metoda2()


Model
IModel

Interfețe

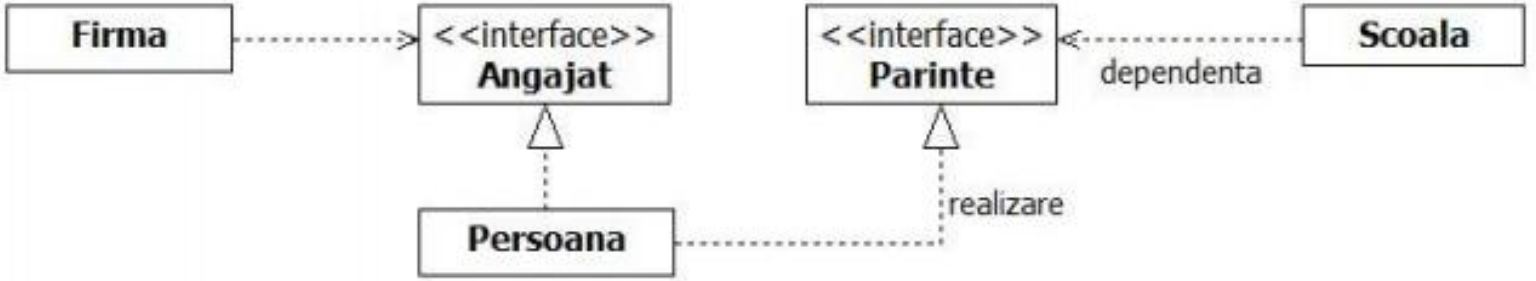

Scoala
1..*
+Parinte
1..*
1..*
Persoana
Firma
+Angajat

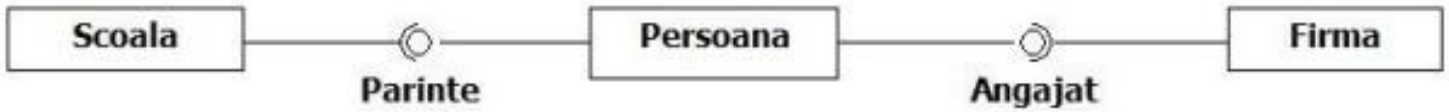

Persoana
Parinte
Angajat

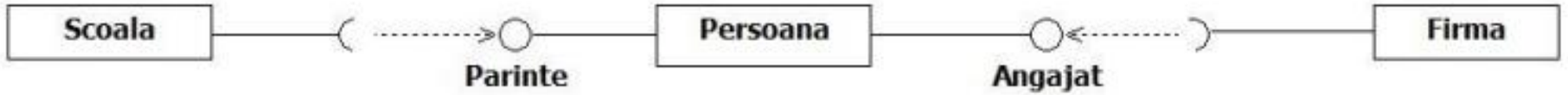

Scoala
Firma

Relații între clase și interfețe

- Relația de realizare – între o interfață și clasa care o realizează
- Relația de dependență – între o clasă și interfața pe care o utilizează



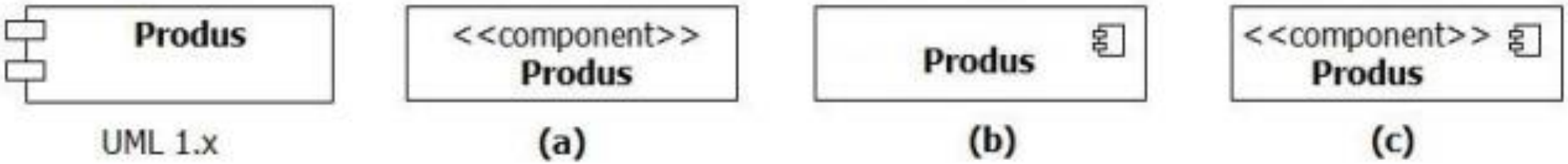




Diagrame de componente

- Ilustrează relațiile structurale dintre componentele software ale unui sistem
- Componente
  - Element fizic: document, fișier, tabelă a unei BD, componentă binară (UML 1.x)
  - O unitate de proiectare de nivel înalt care poate fi implementată prin entități fizice care pot fi înlocuite (UML 2.x)

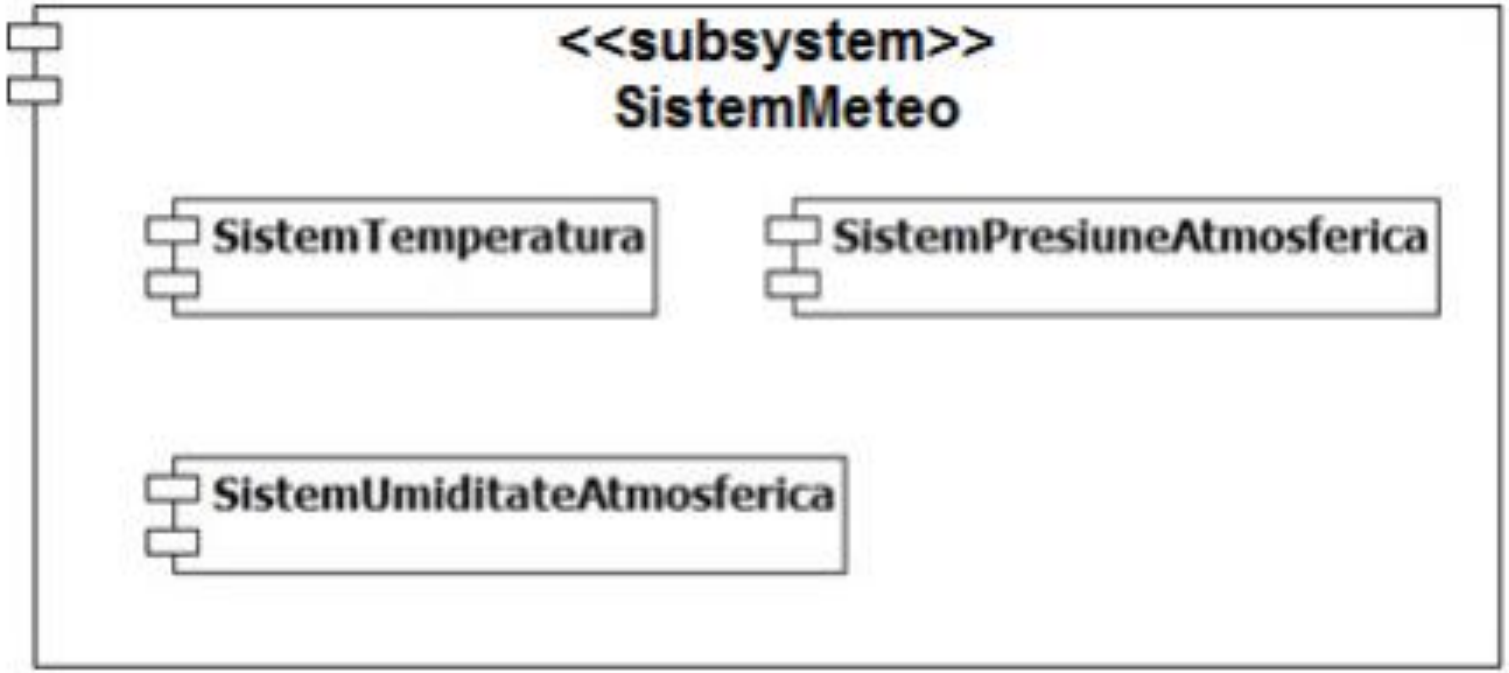
Diagrame de componente



UML 1.x (a) (b) (c)

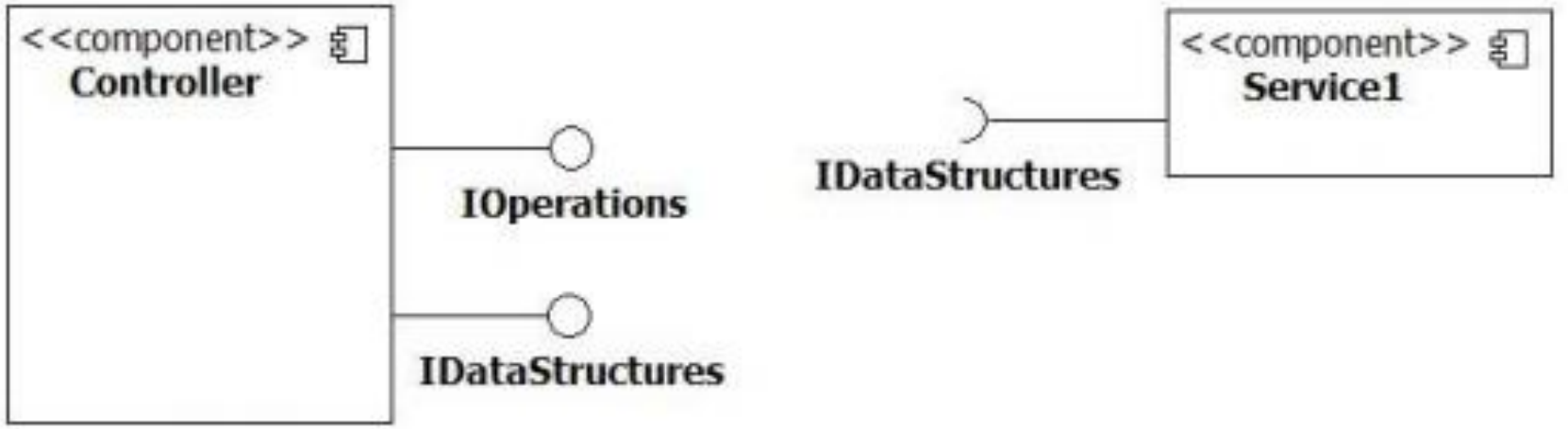
Diagrame de componente

- Subsistem – grupează mai multe componente



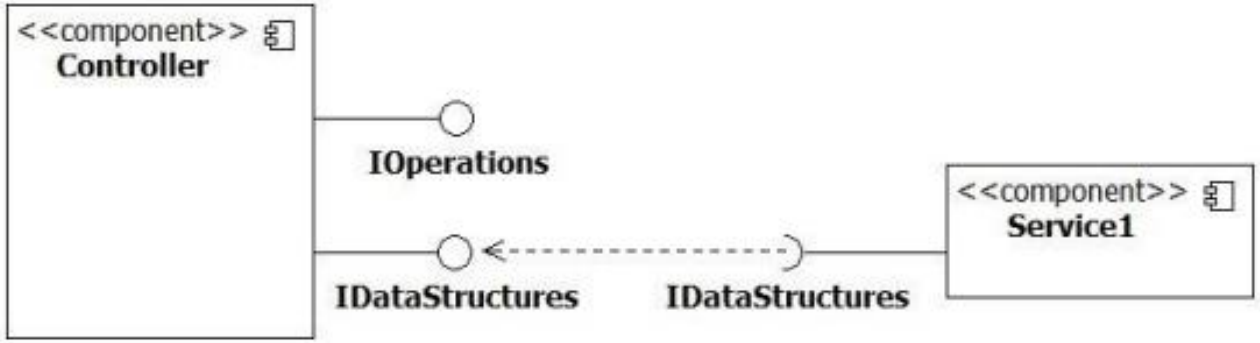
Diagrame de componente

- Interfețe
  - Furnizată de o componentă poate fi
    - Realizată de componenta însăși
    - Realizată de alte componente

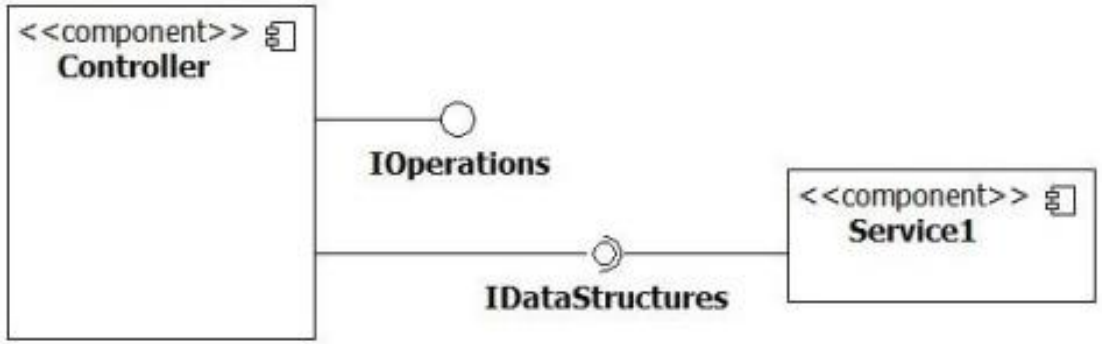


Diagrame de componente

- Relații între componente
  - Dependența

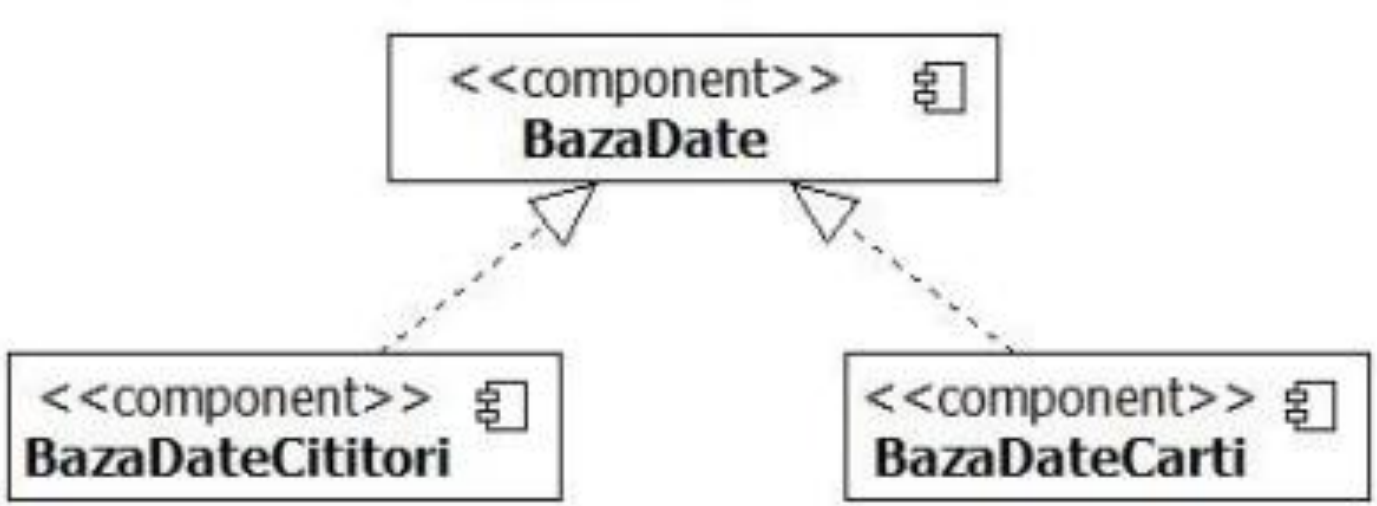


(a)



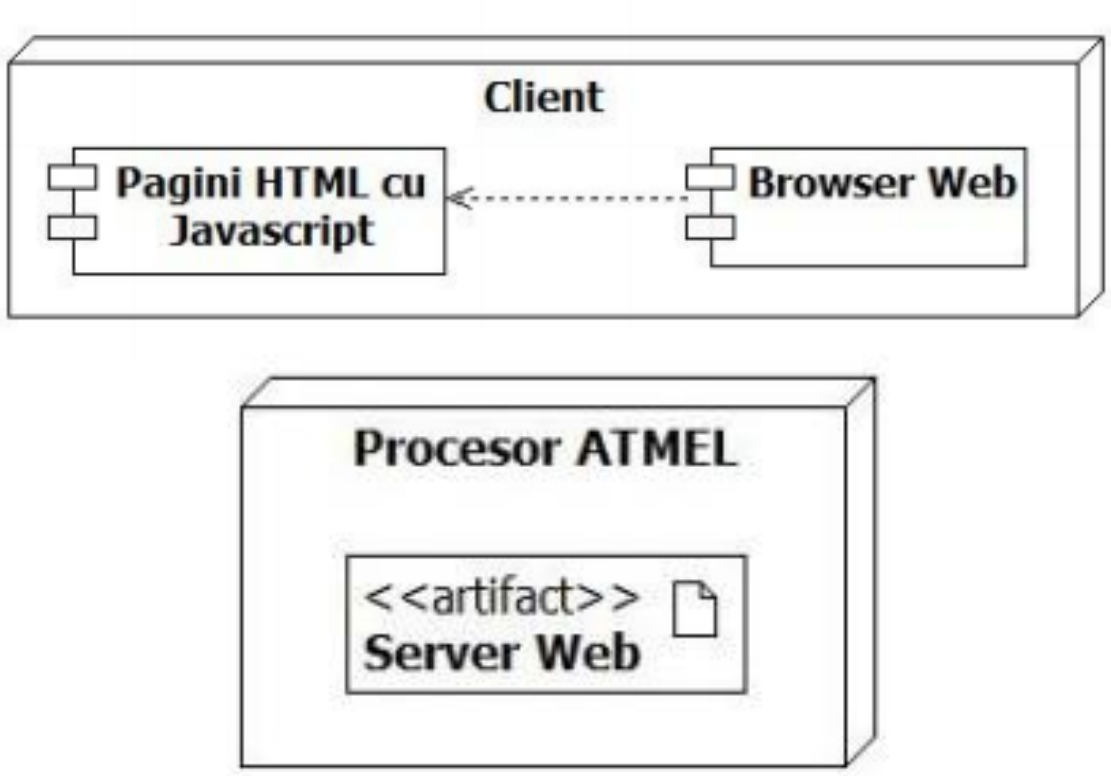
Diagrame de componente

- Relații între componente
  - Realizarea (comportamentul e realizat de alte componente)



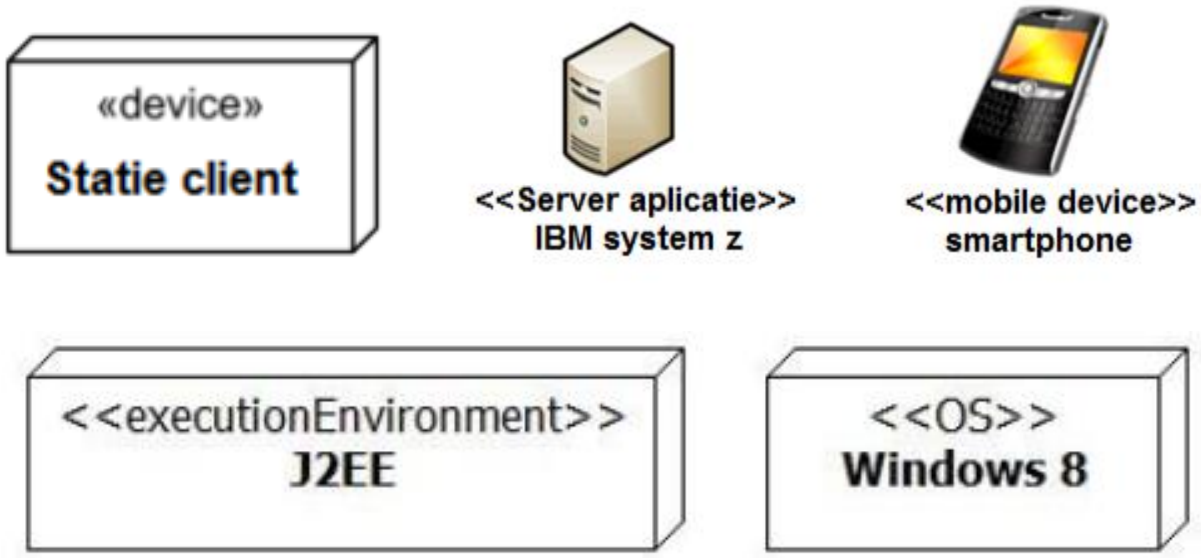
Diagrame de distribuție

- Reprezintă arhitectura unui sistem prin distribuția artefactelor software pe echipamentele mediului de implementare (noduri)



Diagrame de distribuție

- Noduri
  - De tip dispozitiv (echipament hardware)
  - De tip mediu de execuție (resursă software ce rulează pe un echipament hardware)



Diagrame de distribuție

- Artefacte
  - Entități fizice obținute printr-un proces de dezvoltare software
  - Ex: fișiere executabile, biblioteci, fișiere arhivă, BD, documente