

 Colecțiile techIT.ro

 Analiza cerințelor software

 Colecția Tutorial SQL


 Tutorial UML

 Despre informație

 Dicționar IT

 Exemple de cod

 Economie

 Căutare după tag:

.NET  
Actualitate  
ADO  
Analiza cerințelor  
Apple  
Baze de date  
Business  
C#  
Capability Maturity Model (CMM)  
Cascading Style Sheets (CSS)  
Cloud  
Criza economică  
Cum să...  
Despre informație  
Dezvoltare software  
Dicționar  
Economia informației  
Economie  
EPOS  
Exemple  
Exemple de cod  
HTML  
Informație  
Inginerie software  
Internet  
JavaScript  
Microsoft SQL Server  
Modelare  
MySQL  
Opinii  
PHP  
Piață  
Pregătire profesională  
Programare  
SaaS  
Seria Analiza cerințelor software  
Specificații software  
SQL  
Știri  
Tehnologia informației  
Tooluri  
Tutorial SQL  
Tutoriale  
Unified Modelling Language (UML)  
Vânzare  
Visual Studio .NET

  
☐ Web ☒ techit.ro  

[Home](#) | [Dicționar IT](#) | [Download](#) | [Forum](#) | [Despre noi](#) | [Contact](#)

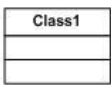

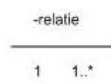
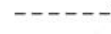
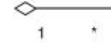

Tag-uri: Tutoriale, Unified Modelling Language (UML), Modelare, Cum să...

## Tutorial UML (partea a V-a). Class Diagram

*Class diagram* este un tip de diagramă utilizată pentru descrierea structurii statice, adică a *entităților* sau *claseelor* existente într-un sistem. Acest tip de diagramă este utilizat cel mai adesea de către dezvoltatori pentru specificarea claselor dar poate fi foarte util și pentru specificarea structurii unor sisteme sau subsisteme dintr-un business real.



Elementele utilizate și notațiile lor sunt următoarele:

Element	Descriere	Notație
Clasă	O clasă este reprezentată printr-un dreptunghi cu trei compartimente: în cel de sus se trece numele clasei, în mijloc se trec <i>atributele</i> clasei iar jos se trec <i>operațiile</i> specifice clasei.	
Moștenire	Moștenirea este o relație care indică faptul că o clasă moștenește caracteristicile unei clase părinte. Sensul săgeții indică sensul în care se poate spune despre clasa copil că <i>este o</i> <->, <i>sau este de tipul</i> <-> clasă părinte.	
Asociere	Asocierea este o relație generică între două clase. Aceste relații pot fi de tipurile pot defini și regulile numerice de asociere (unu la unu, unu la mai mulți, mai mulți la mai mulți).	
Dependență	Atunci când o clasă depinde de o altă clasă, în sensul că utilizează acea clasă ca și atribut al său, se folosește relația de dependență.	
Agregare	Agregarea indică o relație de tip întreg-parte (se poate spune despre clasa părinte că are clase de tip copil). În această relație, clasa copil poate exista și fără clasa părinte.	
Compoziție	Această relație derivă din agregare dar se utilizează atunci când o clasă copil nu poate exista decât în cazul existenței clasei părinte.	

În reprezentarea clasei atributele și operațiile sunt declarate în compartimentele speciale din dreptunghi, astfel:

- atributele:

numele atributului : tipul atributului = valoare implicită

- operațiile:


numele operației (parametri) : tipul valorii returnate

Atunci când diagrama este folosită pentru a modela structuri de business se pot folosi tipurile de date specifice business-ului, nu programării, de exemplu: minut, dată calendaristică, minut etc.

Autovehicul	
+numar_inmatriculare :	string
+capacitate_cilindrica :	Decimal
+ultima_revizie :	Date
+modifica_ultima_revizie(in data :	Date) :
	Boolean

Moștenirea este o relație prin care se indică faptul că o clasă moștenește caracteristicile clasei părinte. În plus, clasa copil poate avea propriile caracteristici.

*Daca ai impresia ca educatia e scumpa, atunci incearca sa vezi cum e ignoranta.*  
Andy McIntyre

 Topul celor mai citite articole

1. Tutorial SQL. Baze de date (I). Crearea unui exemplu
2. Tutorial UML (partea a V-a). Class Diagram
3. Tutorial SQL. Baze de date (partea II-a). Crearea unui nou exemplu
4. Bază de date relațională
5. Exemplu de utilizare PHP cu MySQL: formular pentru introducerea date
6. Tutorial UML (partea a III-a). Activity Diagram
7. Tutorial UML (partea a II-a). Use Case Diagram
8. Tutorial SQL. Baze de date (partea III-a). Normalizarea
9. Despre UML. Introducere, generalități
10. Tutorial UML (partea a IV-a). Statechart Diagram

BURSA DE MARFA

# Produs

V [Arpagic](#)

V [DOWEX SBR-P rasina anioni ...](#)

V [DOWEX SBR-P rasina anioni ...](#)

V [MĂ/lai grisat](#)

V [Yellow Union](#)

V [Red Union](#)

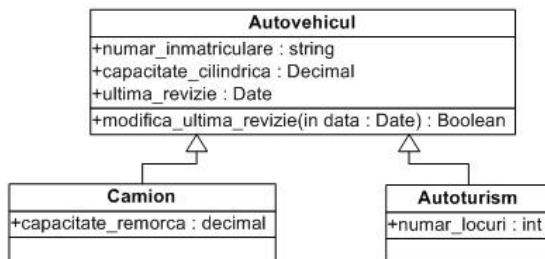
V [Miere](#)

V [ulei catina natural](#)

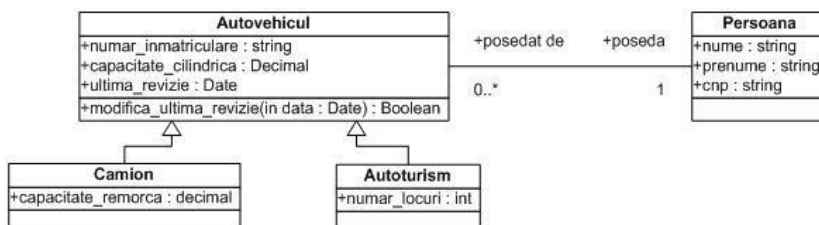
V [Stoc incaltaminte piele + ...](#)

[www.bursademarfa.ro](http://www.bursademarfa.ro)



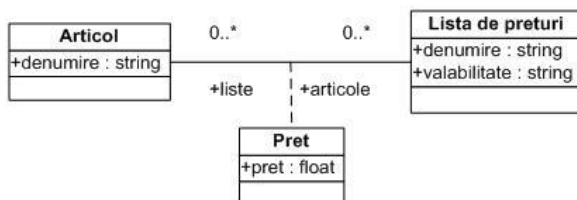


Asocierea arată existența unei relații între clase. În exemplul de mai jos, între Persoană și Autovehicul următoarea relație: o Persoană poate avea zero, unul sau mai multe Autovehicule.

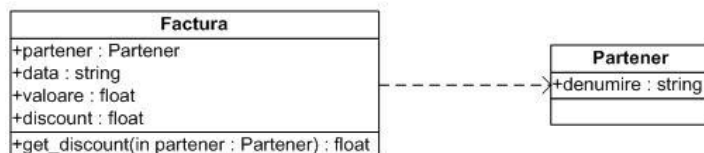


Un tip special de asociere este indicat printr-o *clasă de asociere*. Altfel spus, relația în sine este o clasă.

În exemplul de mai jos, relația dintre Articol și Lista de prețuri este de tip mai mulți la mai mulți: un Articol poate să apară pe mai multe Liste și o Listă poate avea mai multe Articole. Pe Liste diferite Articolele pot avea prețuri diferite.



Dependența indică faptul că o clasă depinde de altă clasă, în sensul în care o funcție oarecare depinde de un parametru al său.



Agregarea indică faptul că o clasă părinte are elemente de tipul clasei copil. În exemplul de mai jos Țara poate avea mai multe Județe dar, în același timp, un Județ poate exista chiar și în cazul în care clasa Țara nu există.



Într-o relație de tip compoziție clasa copil nu poate exista decât dacă există o instanță a clasei părinte. În exemplul de mai jos instanța clasei Comisie există atâta timp cât există instanța clasei Examen.



techit.ro

Noi rute din C  
Zburati cu  
Wizz Air

▶ Colecția: Tutorial UML

▶ Articolul precedent: Tutorial UML (partea a IV-a). Statechart Diagram



✚ Spune-ti parerea despre acest articol!

Copyright 2008 techIT.ro

[Home](#) | [Termeni de utilizare](#) | [Politica de confidentialitate](#) | [Contact](#)

