

Diagrame de obiecte

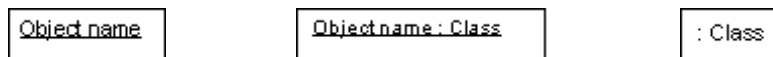
Cele ce urmează sunt o traducere din Instant UML, de Pierre-Alain Muller, 1997

Diagramele de obiecte (de instanțe) ilustrează obiecte și legături. Ca în cazul diagramelor de clase, diagramele de obiecte reprezintă structura statică. Notăția pentru diagramele de obiecte este derivată din aceea pentru diagramele de clase; elementele care sunt instanțe sunt subliniate.

Diagramele de obiecte sunt folosite în primul rând pentru a arăta un context – spre exemplu, înainte sau după o interacțiune. Totuși, ele sunt folosite în a ajuta la înțelegerea structurilor de date complexe, cum sunt structurile recursive.

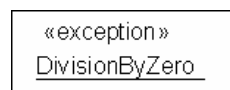
Reprezentarea obiectelor

Fiecare obiect este reprezentat printr-un dreptunghi care conține fie numele obiectului, numele și clasa obiectului (separate de simbolul două puncte) sau numai clasa obiectului (caz în care se zice că obiectul este anonim). Numele în sine corespunde unui model incomplet, în care clasa obiectului nu a fost încă specificată. Clasa singură evită introducerea de nume care nu sunt necesare în diagrame și permite în același timp exprimarea mecanismelor generale valabile pentru multe obiecte. Diagramele de mai jos ilustrează cele trei posibilități de reprezentare.



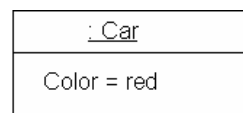
Numele clasei poate conține calea completă construită din diferitele pachete care o conțin, separate prin două puncte duble, ca în exemplul acesta:

ButtonOK : UI :: Controls :: PushButton



Stereotipul clasei poate reapărea în compartimentul de obiect, fie folosindu-se forma textuală (între ghilimele franțuzești (« ») deasupra numelui obiectului), forma sa grafică (în colțul din dreapta sus), sau folosind o anumită reprezentare grafică ce înlocuiește simbolul de obiect. Nu există stereotip de obiect;

întotdeauna stereotipul care apare într-un obiect este stereotipul clasei.



Dreptunghiurile care simbolizează obiecte pot să includă și un al doilea compartiment care conține valorile atributelor. Tipul de atribut este deja specificat în clasă, așa că nu este necesar să-l afișăm în reprezentările obiectelor. Diagrama din stânga reprezintă un obiect de clasă Car anonim, cu atributul Color cu valoarea red.

Reprezentarea legăturilor

Obiectele sunt conectate prin legături, care sunt instanțe de asocieri între clasele obiectelor considerate. Reprezentarea concretă a unei structuri prin obiecte este adesea mai relevantă decât reprezentarea abstractă a structurii folosind clasele, în special în cazul structurilor recursive. Diagrama de obiecte de dedesubt ilustrează o porțiune din structura generală a automobilelor. Fiecare automobil (**Car**) are un motor și patru roți – **wheel** (exclusiv roata de rezervă!).

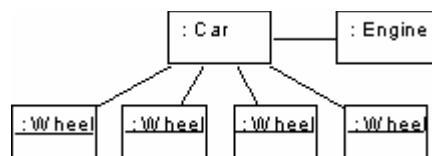
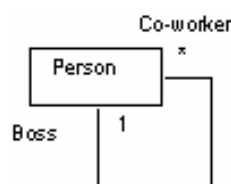
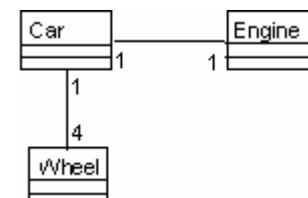
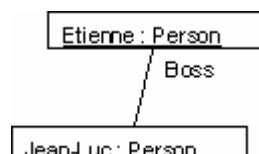


Diagrama din stânga este o instanță a diagramei de clase din dreapta.

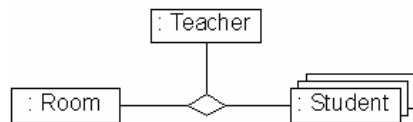


propriul său șef.

Legăturile care sunt instanțe ale cazului asocierilor reflexive sunt legături care pot lega un obiect de sine însuși. În acest caz, legătura este reprezentată de o buclă atașată unui singur obiect. Exemplul următor ilustrează două legături care sunt instanțe ale aceleiași asocieri. Prima legătură arată că Etienne este șeful lui Jean-Luc, iar cea de a doua arată că Denis este



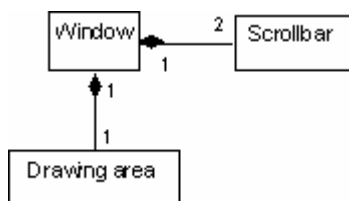
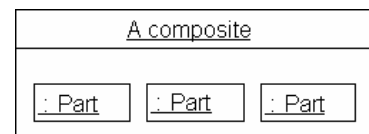
Majoritatea legăturilor sunt binare. Există totuși unele legături care pot conecta împreună mai mult de două obiecte – cele care corespund relațiilor ternare, de exemplu. Reprezentarea relațiilor ternare poate fi combinată cu celelalte elemente de notare: diagrama de mai jos reprezintă o familie de legături ternare care sunt instanțe ale unei asociații ternare de multiplicitate N pe partea clasei **Student**. Aceasta notație are avantajul înlăturării oricăror ambiguități inerente reprezentării multiplicității pentru asocieri non-binare.



În stânga este un exemplu de combinație de elemente de notare pentru a reprezenta relații ternare multiple într-un mod condensat.

Obiecte compozite

Obiectele compuse din sub-obiecte se pot reprezenta folosind un obiect compozit pentru a reduce complexitatea diagramei. Obiectele compozite sunt reprezentate ca obiectele normale cu excepția faptului că atributele sunt înlocuite de obiecte, fie folosind text subliniat, fie o reprezentare grafică. Diagrama din dreapta următoare ilustrează reprezentarea grafică a obiectelor compozite.



Obiectele compozite sunt instanțe de clase compozite – clase construite din alte clase folosind cea mai tare formă de agregare. Diagrama din stânga reprezintă o clasă **Window** compozită.

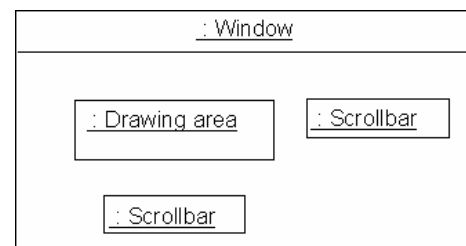
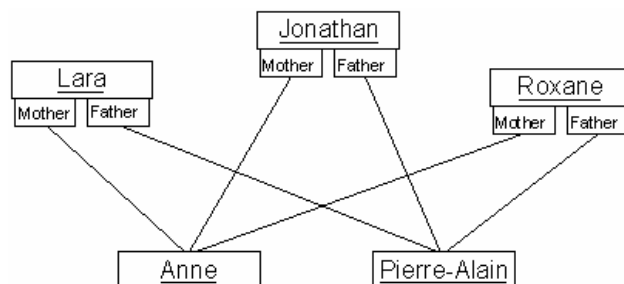
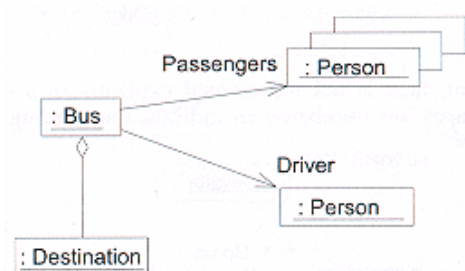


Diagrama de obiecte din dreapta este o instanță a clasei compozite de deasupra. Ea reprezintă cea mai generală formă de obiecte compozite **Window**, din punctul de vedere al obiectului.

Asemănările cu diagramele de clasă

Etichetele care figurează în diagrama de clase pot fi în majoritate copiate în diagramele de obiecte pentru a facilita înțelegerea interacțiunii. Aceasta se aplică tuturor caracteristicilor asocierii (nume, nume de rol, agregare, compoziție și navigare), cu excepția multiplicității, care este reprezentată explicit prin legături. Diagrama de obiecte din dreapta poate fi deosebită grafic de o diagramă de clase deoarece numele de obiecte sunt subliniate.



Valorile calificatorilor asocierii pot fi și ele adăugate în diagramele de obiecte. Diagrama din stânga reprezintă legăturile parentale dintre Lara, Jonathan, Roxane, Anne și Pierre-Alain.

Exerciții

Desenați diagramele de obiecte pentru următoarele clase, declarații și atribuiri:

Exemplul 1:

```
public class ClassA
{
    char letterA;
    ClassB letters;
}
public class ClassB
{
    char one;
    char two;
}
ClassA object = new ClassA();
object.letterA = 'x';
object.letters = new ClassB();
object.letters.one = 'y';
object.letters.two = 'z';
```

Exemplul 2:

```
public class ThreeInts
{
    int firstInt;
    OneInt secondInt;
    int thirdInt;
}
public class OneInt
{
    int theInt;
}
ThreeInts object = new ThreeInts();
object.firstInt = 20;
object.secondInt = new OneInt();
object.thirdInt = 40;
object.secondInt.theInt = 30;
```