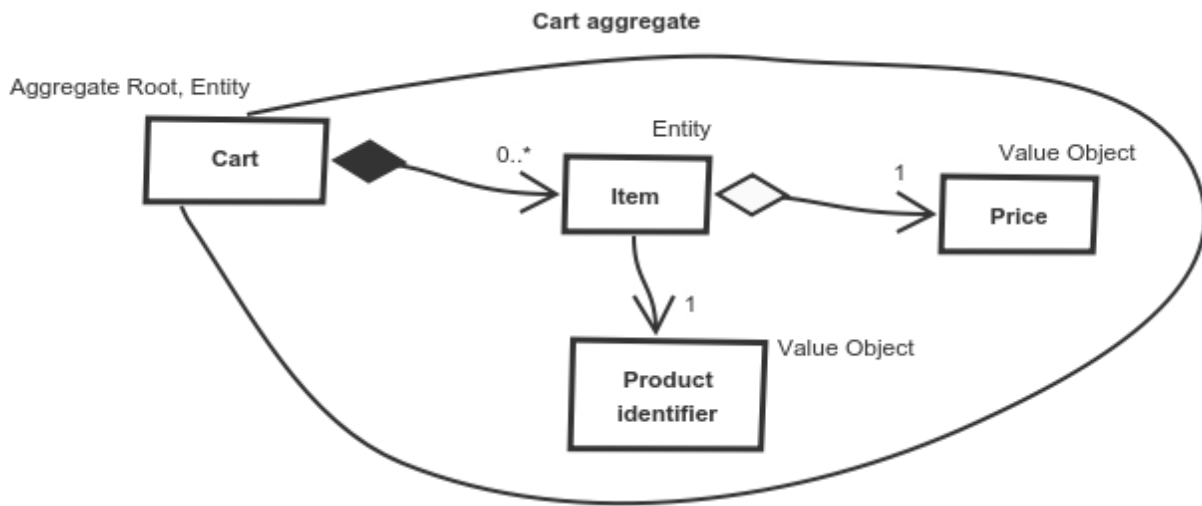


## TEMA 2 PSSC

### *Domain driven design – entity*

DDD reprezintă o metodă practică de a transforma o problemă complexă într-un model ce poate fi implementat ulterior de echipa de dezvoltare software. Există diferite abordări, astfel încât soluția finală să fie funcțională, bazată pe microservicii sau chiar *reactive programming*. Această abordare este în general utilă pentru aplicațiile software practice care încearcă să rezolve o problemă tehnică de business / enterprise.



Entitățile sunt concepte ce pot fi identificate în mod unic și se referă totuși la ceva mai mult de atât. Ele au un aspect ce nu se modifică, ce acționează ca un identificator, dar pe lângă acest aspect putem modifica tot ce are legătură cu identitatea, dar ea rămâne aceeași.

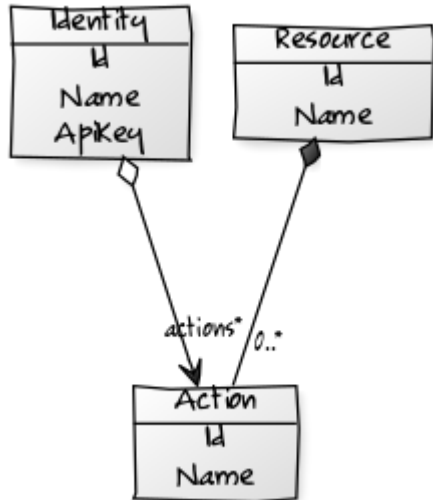
Argumente pro și contra pentru o clasă de entități standard și o clasă cu abordare DDD:

- Avantaje pentru o clasă de entități standard: Simplu. Cod minim
- Avantaje pentru o clasă cu abordare DDD : Foarte observabil, numele variabilelor cu sens. O bun control a acceselor datele.
- Dezavantaje pentru o clasă de entități standard: Un sistem mare devine foarte greu de înțeles. Posibilitatea de cod dublat.
- Dezavantaje pentru o clasă cu abordare DDD: Cod puțin de scris.

Există un tip de egalitate – egalitate de identificare. Se presupune că o clasă are un câmp pentru id. Două instanțe ale acestei clase sunt egale dacă au id-ul la fel. Prin aceasta înțelegem că

în cazul în care avem două instanțieri identice a unei entități, cu excepția proprietății *id*, nu le considerăm ca fiind echivalente.

Dacă am două obiecte Persoană, ce au același Nume, ele nu pot fi considerate ca fiind aceeași Persoană. Două obiecte Persoană cu același nume nu pot fi considerate ca fiind identice, ele, spre exemplu, pot avea Adresa diferită. Prin urmare un obiect Persoană nu poate fi identificat prin Nume, Adresa, NumarTelefon, etc. Un obiect Persoană are o identitate unică ce se manifestă diferit în sisteme diferite. Fiecare sistem are propriile atribute de care este interesat, dar Persoană e de tot timpul aceeași entitate.



Diferențe între o entitate și un value object:

Abilitatea de a fi modificat: Entitățile ar trebui să poată să fie modificate. Având în vedere că identitatea lor e unică, acestea ar trebui să poată să fie modificate, pentru că nu pot fi înlocuite de alte entități. În schimb, un value object ar trebui să nu poată fi modificat, pentru că în cazul în care este nevoie să modificăm un astfel de obiect e mai bine să creăm un nou obiect bazat pe cel existent, decât să-l modificăm pe cel existent.

Durata de viață: Entitățile acestea trăiesc în mod continuu, ele au o istorie cu modificările făcute asupra lor. În schimb, value objects au durată de viață zero. Ele sunt create și șterse cu ușurință. Acestea fiind interschimbabile unul cu altul rezultă că pot fi înlocuite unul cu altul, ceea ce înseamnă că un obiect poate să fie înlocuit de altul nou instanțiat.

## Value Object within Aggregate

