UML

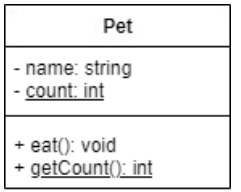
- Unified Modeling Language @ Rational ~ metà 1990
  - Grady Booch, James Rumbaugh e Ivar Jacobson
- Standard OMG dal 1997, ISO dal 2005
- Riferito alla programmazione Object Oriented
  - ma indipendente dal linguaggio di implementazione
- Non completamente formale, basato su diagrammi
- Usato per identificare in un sistema
  - le entità che lo compongono, loro caratteristiche e relazioni

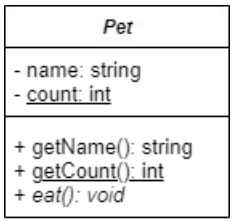
Diagrammi

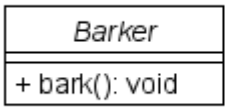
- Modellano il sistema secondo diversi punti di vista
  - Strutturali: caratteristiche statiche del sistema
  - Comportamentali: interazioni tra componenti nel sistema
  - Architetturali: descrizione della struttura del sistema
- Class, Interface, Object
- Use Case: comportamento del sistema in uno specifico caso
- State: cambiamento dello stato di un oggetto nel corso della sua vita
- Sequence: interazione tra gli oggetti
- Activity: sequenza di attività all'interno di un Use Case
- Tra gli altri
  - Communication: cooperazione tra elementi per obiettivo comune
  - Component: struttura di sistemi complessi
  - Deployment: configurazione del sistema a runtime

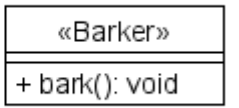
Class

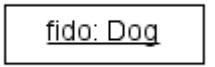
- Una classe è rappresentata da un rettangolo
  - Normalmente diviso in tre parti (+ 1)
    - **Nome**, può essere prefissato dal nome del package (separatore ::)
    - **Proprietà**, visibilità, nome: tipo = inizializzazione
    - **Metodi**, visibilità, nome(parametri): return type
    - Note aggiuntive
- Decoratori per i membri di classe
  - Visibilità: + pubblico, ~ package, # protetto, - privato
  - Staticità: sottolineatura, astrattezza: *corsivo*
- Hanno una rappresentazione simile a quella della classe
  - **Interfaccia**, nome in corsivo (oppure tra doppie parentesi angolari)
  - **Oggetto**, nomi sottolineati





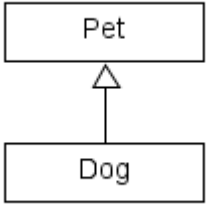


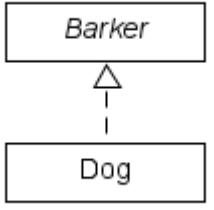




Gerarchie

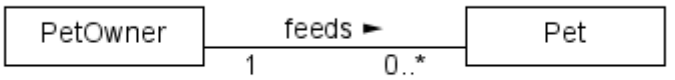
- Indicate con frecce da figlia a madre
- **Generalizzazione**
  - Ereditarietà, “is a”
  - Freccia linea intera con triangolo a punta vuota
- **Realizzazione**
  - Implementazione di interfaccia
  - Freccia tratteggiata con triangolo a punta vuota



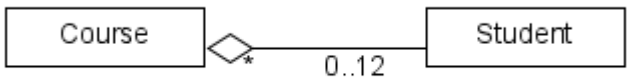


Associazioni

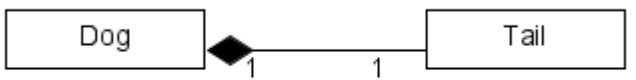
- **Associazione** generica tra classi
  - Una riga connette le classi
  - È possibile indicare come note
    - la molteplicità (1..*) e il comportamento di un oggetto nell'associazione



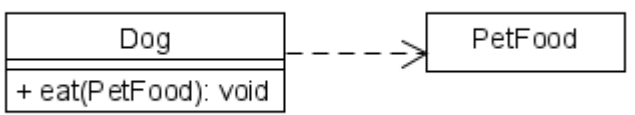
- Oggetti di una classe che posseggono oggetti di un altra classe
  - **Aggregazione**
    - Gli aggregati possono esistere indipendentemente
    - Freccia con diamante vuoto



  - **Composizione**
    - Dipendenza dall'esistenza del proprietario
    - Freccia con diamante pieno

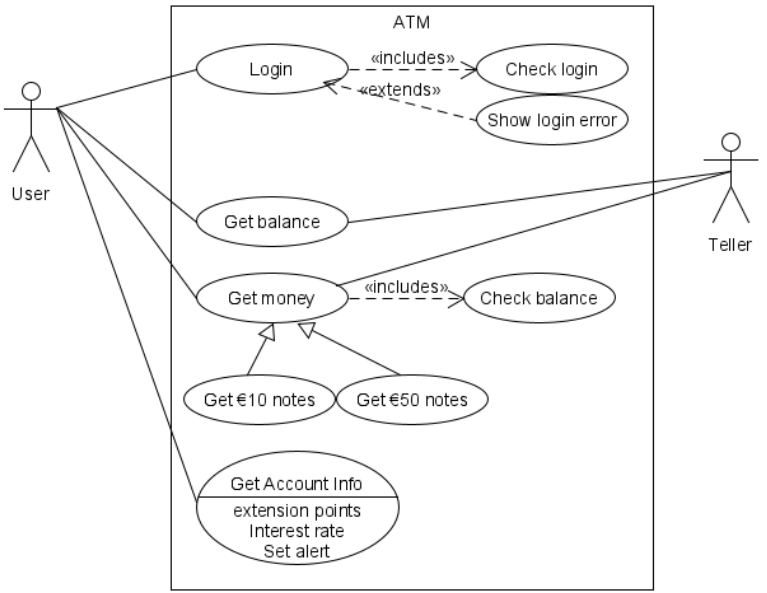


- Oggetti di una classe in relazione debole con oggetti di un altra classe
  - **Dipendenza**
    - Parametro o variabile locale di un metodo
    - Freccia tratteggiata



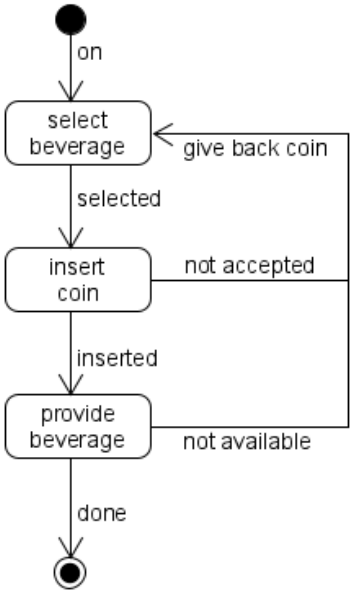
Use Case

- System: cosa si vuole ottenere (web app, componente software, …)
- Actor: omettino stilizzato, interagisce col sistema
  - Primary Actor: inizia l'azione (a sinistra)
  - Secondary Actor: reagisce alle azioni del Primary Actor (a destra)
- Use Case: azione all'interno di System
  - Possibili Extension Points per UC complessi
- Relationship: linea tra Actor coinvolti e UC
  - <<includes>> su freccia tratteggiata
    - Base UC richiede un included UC
  - <<extends>> su freccia tratteggiata
    - Base UC può essere esteso in casi specifici
  - Generalization
    - Gerarchia di UC (o Actor)



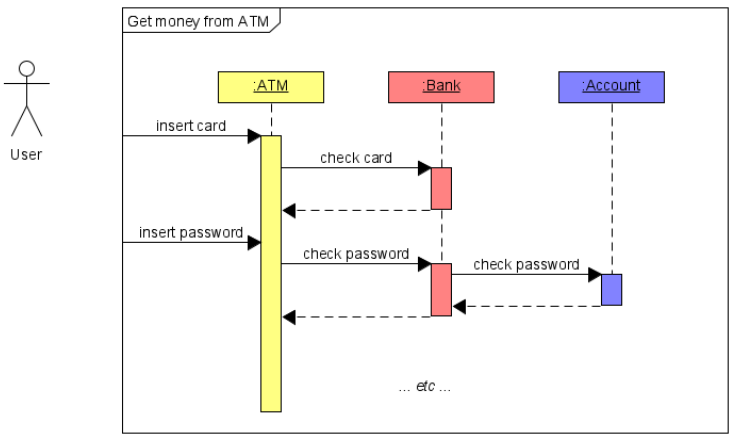
State (Machine)

- Comportamento di un oggetto in seguito a eventi
- Mostra le transizioni tra stati (freccia)
- Stati
  - Iniziale (pallino nero)
  - Intermedio (rettangolo)
  - Finale (pallino nero dentro un cerchio)



Sequence

- Interazione tra classi / oggetti nell'ordine in cui avvengono
  - Sequenza di eventi
- Actor
  - Interagisce col sistema
- Message e return message
- Activation box
  - tempo necessario per un task



Activity

- Flow chart più rigorizzato, state diagram più dettagliato
  - Fork-join per multithreading indicato con una barra
  - Box specifici per send/receive di messaggi asincroni