

# Progetto P2 - Model V 0.3

- Convezioni
  - Nomenclatura
- Gerarchia di classi: visione d'insieme
  - Classe `Item`
    - Campi dati
    - Definizione dei campi dati
  - Classe `Spell`
    - Campi dati
    - Definizione dei campi dati
  - Classe `Generic`
  - Classe `EquipItem`
    - Campi dati
    - Definizione dei campi dati
  - Classe `MagicItem`
    - Campi dati
    - Definizione dei campi dati
  - Classe `Weapon`
    - Campi dati
    - Definizione dei campi dati
  - Classe `Armor`
  - Classe `Potions`

## Convezioni

### Nomenclatura

Possibili convenzioni di nomenclatura:

- I nomi assegnati alle classi sono in lingua inglese con lettera maiuscola. (es: `Item`)
- I nomi assegnati alle variabili sono in lingua inglese composti unicamente da lettere minuscole. Se composti devono essere intervallati dal carattere `'_'` (es: `weapon_name`);
- I nomi dei metodi in lingua inglese. Se il metodo e' composto da più termini il primo ha lettera minuscola, tutte le iniziali dei successivi hanno lettera maiuscola (es. `setName`);
- Devono essere dichiarate le direttive d'uso che si vogliono usare per i metodi appartenenti a un namespace (es. se voglio non dover riscrivere `std::endl` ogni volta che lo utilizzo, devo notificare che viene dichiarata quella direttiva d'uso e in quale file);
- TUTTI i nomi che vengono utilizzati devono essere quanto più concisi possibile, prendendo come lunghezza indicativa tra i 5 e i 7 caratteri (non tassativo ma utile)

---

## Gerarchia di classi: visione d'insieme

La gerarchia si compone di:

- 2 classi base virtuali pure
  - Item
  - Spell
- 3 classi di prima derivazione
  - Generic (concreta)
  - Equipltem (virtuale)
  - MagicItem (virtuale)

Da Equipltem derivano due classi concrete

- Melee
- Ranged
- Armor

Da WizardItem deriva una classe concreta

- Potion



#### Sidenote

La classe **container** e' indicata con il nome temporaneo inventario

La classe **Personaggio** identifica l'oggetto a cui viene associato il container

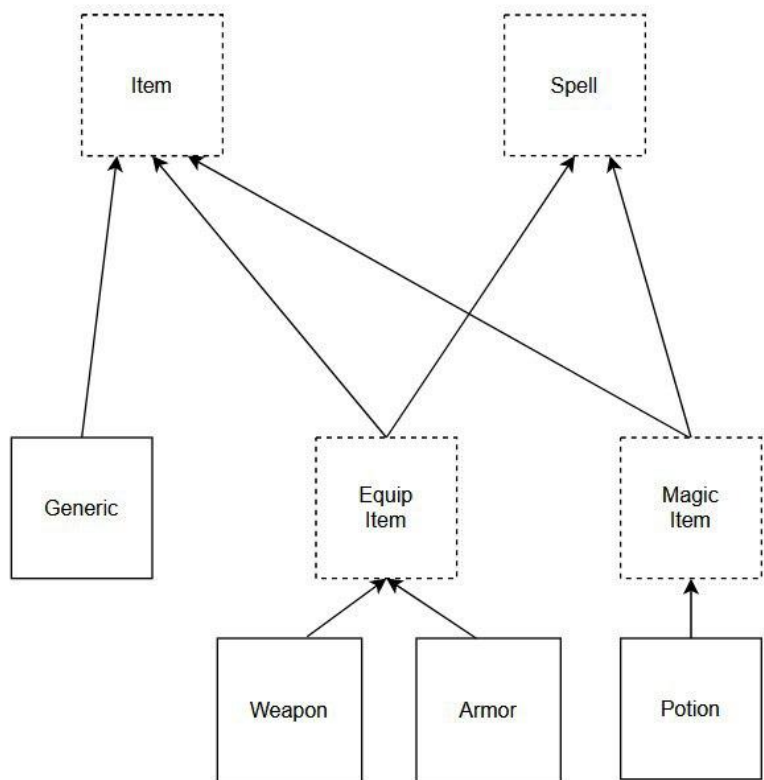
Non viene specificato nulla di aggiuntivo qui circa queste due classi

Le due classi base permettono la definizione di qualunque oggetto rappresentabile e utilizzabile in un gioco di ruolo moderno.

L'idea fondante consiste nel derivare da queste due classi qualora si volesse creare una nuova categoria di oggetti, oppure raffinare le categorie gia' presenti implementando classi che sono figlie di Equipltem, Generic o MagicItem.

Personaggio

Inventario



## Classe Item

Classe base virtuale pura che rappresenta le caratteristiche generiche di un oggetto.

### Campi dati

| Campo Dati   | Tipo   |
|--------------|--------|
| name         | string |
| value        | int    |
| weight       | int    |
| material     | string |
| volume_units | int    |

### Definizione dei campi dati

**name** Il nome dell'oggetto.

**val** Il valore dell'oggetto

**weight** Il peso dell'oggetto

**material** Il materiale di cui é composto l'oggetto

## Classe Spell

Classe base virtuale pure che rappresenta la magia come proprietà che può essere associata ad un item.



**N.B.**

Da osservare che la non presenza di proprietà magiche in un item viene rappresentata come una magia nulla e che quindi viene richiesto un costruttore nullo

### Campi dati

| Campo Dati  | Tipo   |
|-------------|--------|
| name        | string |
| type        | string |
| level       | int    |
| spell_power | int    |
| duration    | int    |

### Definizione dei campi dati

|                 |                              |
|-----------------|------------------------------|
| <b>name</b>     | Il nome della magia          |
| <b>type</b>     | Il tipo di magia (es. Fuoco) |
| <b>level</b>    | Il livello della magia       |
| <b>power</b>    | Il danno della magia         |
| <b>duration</b> | La durata della magia        |

## Classe Generic

Classe derivata unicamente da Item che rappresenta un item che non possiede proprietà magiche. Non implementa campi dati propri ma si limita a ridefinire i metodi ereditati da Item.

## Classe EquipItem

Classe derivata da Item e Magic che rappresenta il concetto generico di equipaggiamento da battaglia

## Campi dati

| Campo Dati  | Tipo   |
|-------------|--------|
| equip_power | int    |
| type        | string |
| rarity      | string |
| equipped    | bool   |

## Definizione dei campi dati

---

|                    |   |
|--------------------|---|
| <b>equip_power</b> | Il valore di potenza dell'equipaggiamento |
|--------------------|---|

---

|             |   |
|-------------|---|
| <b>type</b> | Il tipo di equipaggiamento, distinguibile in: <ul style="list-style-type: none"><li>• leggero</li><li>• pesante</li></ul> |
|-------------|---|

---

|               |  |
|---------------|--|
| <b>rarity</b> | La rarità dell'arma che ha come valori possibili: <ul style="list-style-type: none"><li>• common</li><li>• rare</li><li>• epic</li><li>• legendary</li></ul> |
|---------------|--|

### ! N.B.

Questa classe indica una generalizzazione di un oggetto, quindi non ha senso andare a definire un livello di rarità **Unique** per identificare un oggetto non categorizzabile (es. Excalibur e' un oggetto unico, non riproducibile). Nel caso in cui si volesse creare un oggetto unico si deve creare una nuova classe che deriva da `EquipItem` e che identifica quello specifico oggetto.

## Classe MagicItem

La classe MagicItem identifica un oggetto generico che pero' gode di proprieta' magiche e che quindi richiede una specifica rappresentazione.

## Campi dati

| Campo Dati | Tipo |
|------------|------|
|------------|------|

| Campo Dati  | Tipo   |
|-------------|--------|
| range       | int    |
| description | string |

## Definizione dei campi dati

|                    |   |
|--------------------|---|
| <b>range</b>       | L' area di effetto della magia          |
| <b>description</b> | La descrzione dell' effetto della magia |

## Classe Weapon

### Campi dati

| Campo Dati      | Tipo   |
|-----------------|--------|
| melee_type_dmg  | string |
| ranged_type_dmg | string |
| min_range       | int    |
| max_range       | int    |

## Definizione dei campi dati

|                        |   |
|------------------------|---|
| <b>melee_type_dmg</b>  | Il tipo di danno che l'arma e' in grado di infliggere corpo a corpo                           |
| <b>ranged_type_dmg</b> | Il tipo di danno che l'arma e' in grado di infliggere dalla distanza, se vi e' modo di farne. |
| <b>min_range</b>       | La distanza minima a cui e' possibile colpire   |
| <b>max_range</b>       | La distanza massima a cui e' possibile colpire  |

## Classe Armor

Classe derivata da Item e Magic che rappresenta un armatura nella sua interezza. Si occupa solamente di definire i metodi che deriva dalle classi da cui deriva, senza aggiungere informazioni particolari.

## Classe Potions

Classe derivata da MagicItem e che definisce gli oggetti pozione. Si limita ad implementare i metodi che vengono forniti dalla classe MagicItem.