

Progetto P2 - Model

Indice

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

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`

- Magic
- 3 classi di prima derivazione
 - Generic (virtuale)
 - Weapon (virtuale)
 - Armor (concreta)

Da Generic derivano due classi concrete

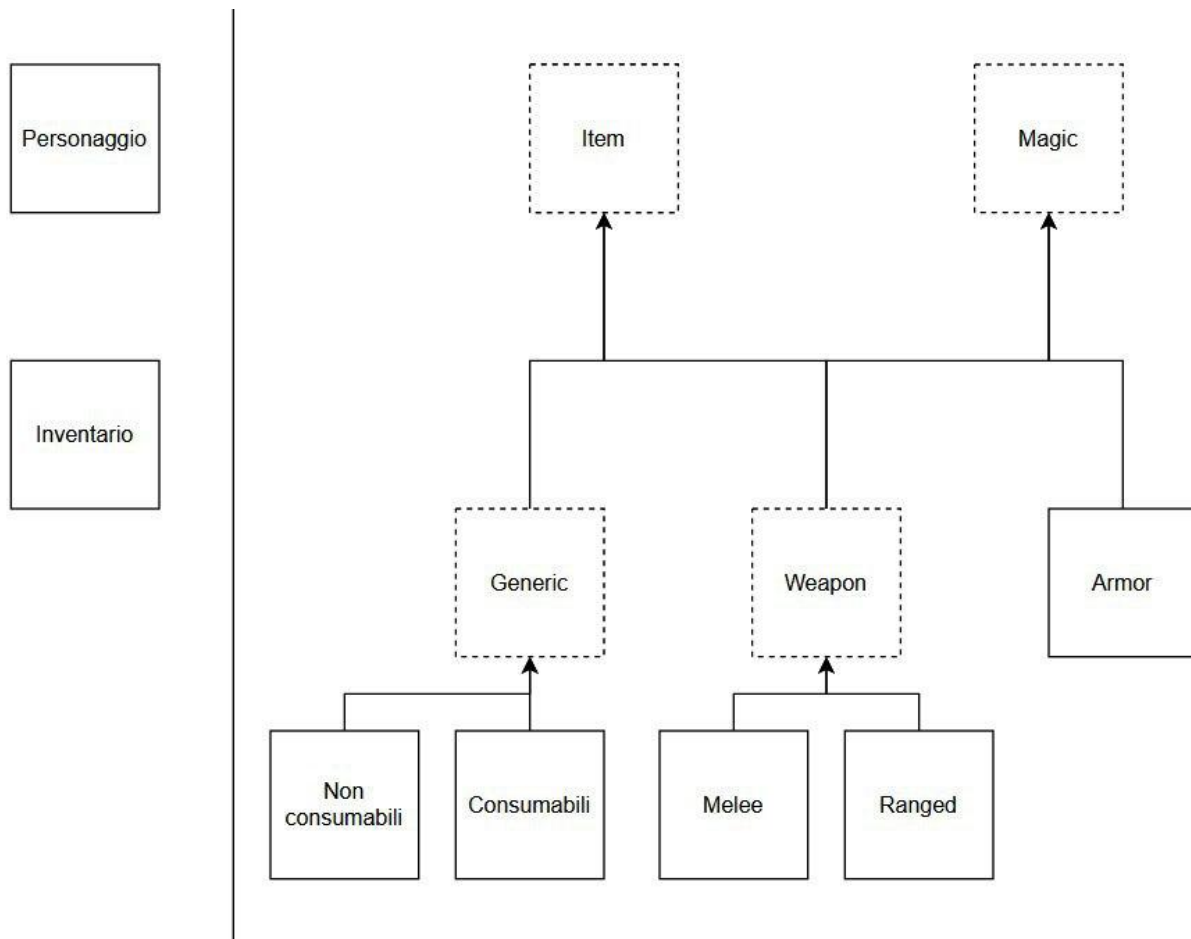
- Non consumabili
- Consumabili

Da Weapon derivano due classi concrete

- Melee
- Ranged

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

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



Classe Item

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

Campi dati

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

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

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 Magic

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 |
|------------|--------|
| type | string |
| level | int |
| damage | int |

Definizione dei campi dati

| | |
|---------------|------------------------------|
| type | Il tipo di magia (es. Fuoco) |
| level | Il livello della magia |
| damage | Il danno della magia |

Classe Generic



Punto cieco

Cosa incapsula il concetto di Generic???

Classe Weapon

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

Campi dati

| Campo Dati | Tipo |
|------------|--------|
| atk | int |
| type_dmg | string |
| rarity | string |

Definizione dei campi dati

atk Il valore di attacco dell'arma

typeDmg Il tipo di danno che l'arma procura , in questo caso vengono considerati:

- bludgeoning (impatto)
- piercing (perforazione)
- slashing (taglio)

rarity La rarità dell'arma che ha come valori possibili:

- common
- rare
- epic
- legendary
- unique (oggetto unico)

Classe Armor

Classe derivata da Item e Magic che rappresenta un armatura nella sua interezza

Campi dati

| Campo Dati | Tipo |
|------------|--------|
| def | int |
| type_armor | string |
| rarity | string |

Definizione dei campi dati

| | |
|------------------|---|
| def | Il valore di difesa dell'armatura |
| typeArmor | Il tipo dell' armatura, quindi <i>heavy</i> o <i>light</i> |
| rarity | La rarità dell'arma che ha come valori possibili: <ul style="list-style-type: none">• common• rare• epic• legendary• unique (oggetto unico) |



Problema

C'e' molta ripetizione di codice che potrebbe essere risparmiata, soprattutto tra le classi **Weapon** e **Armor**. Prima di continuare bisogna fare qualcosa.