

Quarta Lezione: Presentazione Progetto e Specifica con JMI

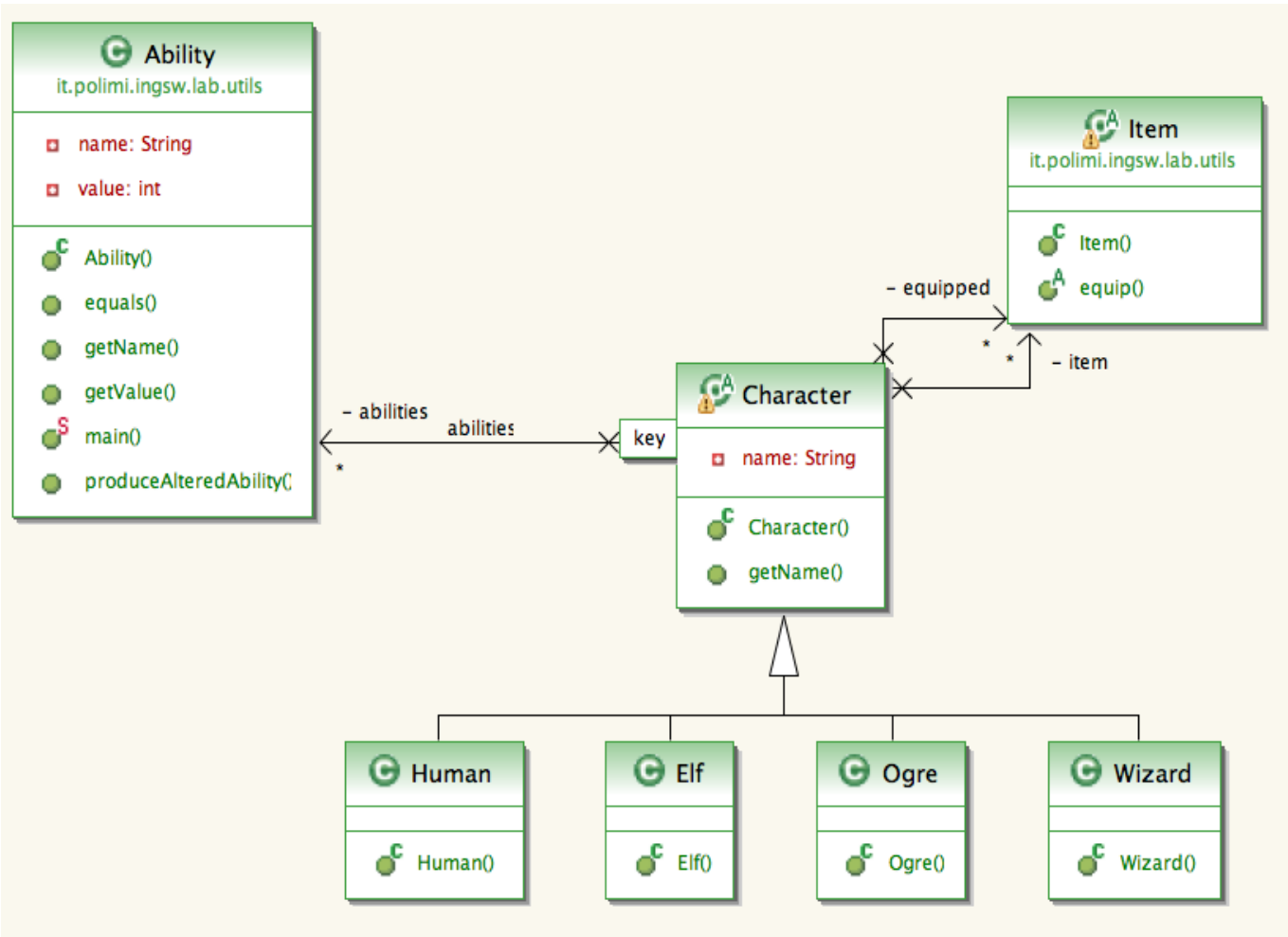


Laboratorio di Ing SW
Sez. Ghezzi

Agenda

- Abstraction Function
 - La classe Character
- Eccezioni
- Representation Invariant
 - RepOk
- Logica di gioco

Rapido riepilogo...



Abstraction Function

- Definisce il significato della rappresentazione
- E' importante che per oggetti uguali restituisca uguale valore
- ES: Ability: quando due Ability si dicono uguali?
 - Una toString potrebbe andare bene?
 - Cosa deve restituire?

La Classe Character

- Complichiamo le cose: la classe Character
- Come è fatta la rappresentazione di Character?
- E delle sue sottoclassi?
- Proviamo a pensarci insieme...

Eccezioni

- Servono a gestire casi di funzionamento eccezionale
- Ampliano il dominio di input

Representation Invariant

- Definisce le ipotesi sottostanti all'implementazione del tipo
- E' importante rispettare controllare l'invariante quando si progetta un tipo
- Possibili soluzioni
 - JML
 - repOk

Esempi

- Ability: la scorsa volta abbiamo progettato la rep di Ability
 - JML : `//@private invariant this.name != null && !this.name.equals("") && this.value >= 0 && this.value <= 100;`
 - RepOk:
 - ```
public boolean repOk(){
 if(this.name != null && !this.name.equals("")
 && this.value >= 0 && this.value <= 100)
 return true;
 }
 return false;
```



# Esempi

- E per la classe Character?
- Come e dove riusciamo a definirla?
- Vediamolo insieme

# La logica del gioco

- Fino ad ora abbiamo usato un progetto con Java 1.4
- Per questa parte useremo Java 1.5
- Bisogna creare un nuovo progetto
- Bisogna creare una dipendenza con il vecchio progetto
- Vediamo come

# Consigli

- Vi interessa veramente che forma ha il tabellone?
- Non pensatelo come una matrice!!!
- Interfaccia utente: Vi interessa veramente sapere se l'interfaccia è grafica o testuale?
- E' bene disaccoppiare l'interfaccia dal resto del progetto
- Come si può fare?

# MVC

- Model-View-Controller
- Model = quello che rappresentate
- View = quello che vedete
- Controller = intermediario che fa comunicare quello che vedete e quello che rappresentate
- Esempio: Character = model, rappresentazione di character a schermo = view... e il controller?