

Generic types



Generic types

- Una classe può essere parametrizzata su più tipi
- Grande riutilizzo di codice
- Il tipo generico viene inserito in tutti i metodi e campi
- Possono essere passati ai supertipi
- Utilizzati soprattutto per le strutture dati (Dove importano le operazioni e non il contenuto)



Generic types

- Sono **invarianti**: se i generic di due espressioni sono differenti non si può assegnare una all'altra anche se hanno una relazione di sottotipo
- Gli array invece sono **covarianti**, ma questo può portare ad errori a runtime (generics prevengono questo problema)

Generic methods

```
public static <T, S> Type<S> method(T param, ...)
```

- I metodi possono essere parametrizzati indipendentemente dalla classe, dichiarando il generic prima del tipo di ritorno

Inferenza dei tipi

- Quando si istanzia una classe o si chiama un metodo che utilizza generic, non è necessario specificarli
- Java riesce a inferire il tipo, in quanto il tipo statico è conosciuto a compilazione
- Per istanziare, bisogna comunque specificare che la classe è parametrica, aggiungendo `<>` prima delle parentesi

`new ClasseParametrica(...)` \neq `new ClasseParametrica<>(...)`

Bounded generics

<T extends Classe>

- Si può dare un limite superiore ad un generic utilizzando la keyword `extends`
- In questo modo i tipi accettati saranno solo i sottotipi di *Classe*

Wildcard

<?>

- Una classe parametrizzata con un wildcard è superclasse di ogni altra classe parametrizzata con T
- Possono essere bounded come i generic
- Sono **covarianti**: se B estende A , $\text{Classe}<? \text{ extends } B>$ è sottotipo di $\text{Classe}<? \text{ extends } A>$



Wildcards, super e extends

- Se la wildcard di una lista è bounded superiormente (? extends C), è solamente possibile leggere i suoi elementi, che saranno di tipo C (non si conosce il tipo degli elementi che si può inserire)
- Se la wildcard di una lista è bounded inferiormente (? super C) è solamente possibile inserire nuovi elementi, che saranno di tipo C (un sottotipo di C può essere messo al posto di C, ma non si sa che tipo ritornare)