

JAVA GENERICS E VETTORI

Angelo Di Iorio

Università di Bologna

Java Generics

- Java permette di dichiarare classi e metodi *generici* che possono *operare su tipi di dato diversi*
 - *Non necessarie operazioni esplicite di casting*
 - *Possibile scrivere algoritmi generici e applicabili a diversi tipi di dato*
- E' possibile definire classi parametriche rispetto a *formal type parameters* che possono essere usati come tipi dichiarati per variabili, parametri e valori di ritorno
- La definizione di una classe generica si esprime con la sintassi: `ClassName<T>`
- `T` è un tipo utilizzabile nella classe e nei suoi metodi

Esercizio 1

- Implementare in Java una classe generica per gestire una coppia di oggetti dello stesso tipo
- La classe espone i seguenti metodi:
 - Costruttore: prende in input i due oggetti
 - `getFirst()` e `getSecond()`: ritornano rispettivamente il primo e il secondo oggetto
 - `inverti()`: scambia la posizione dei due oggetti

Esercizio 2

- Implementare in Java una classe generica per gestire una coppia di oggetti (anche di tipo diverso)
- La classe espone i seguenti metodi:
 - Costruttore: prende in input i due oggetti
 - `checkOmogenei()`: ritorna `true` se i due oggetti sono dello stesso tipo (istanze della stessa classe), `false` altrimenti

Esercizio 3

- Implementare in Java una classe generica per gestire una coppia di oggetti (anche di tipo diverso) **colorati**
- La classe espone i seguenti metodi:
 - Costruttore: prende in input i due oggetti
 - `coloraTutti(Color c)`: setta il colore di tutti gli oggetti a `c`
- Come imporre che gli oggetti siano colorabili?

Vettori in Java

- Un array (vettore) è un contenitore:
 - di dimensione fissa
 - di elementi dello stesso tipo

- Per inizializzare un vettore in Java:

```
Integer[] integers = new Integer[10];
```

```
Integer[] integers = {10, 2, 4, 5, 6};
```

```
String[] words = {"cane", "gatto", "tacchino"};
```

- La classe *java.util.Arrays* (statica) espone metodi per accedere e manipolare vettori

Esercizio 4

- Scrivere una classe Java per modellare e gestire una mensola di libri
- Per questo esercizio:
 - ogni libro ha un titolo
 - la capacità della mensola è fissata in n libri
 - è possibile mettere/prendere un libro in/da una data posizione

Esercizio 4

- La classe `Mensola` espone i metodi per eseguire le seguenti operazioni:
 - inizializzare una mensola di capacità c
 - verificare se la posizione p è libera
 - mettere un libro nella posizione p (se libera)
 - stampare i titoli dei libri sulla mensola (nell'ordine in cui sono)
- Per semplificare, omettiamo altre operazioni comuni come: contare quanti libri ci sono, togliere un libro dalla posizione p , etc.

Esercizio 5

- E se volessimo usare stessa la mensola per CD, VHS, DVD o altri oggetti?
- Usare Java Generics per descrivere una mensola di oggetti generici di tipo T , che espone i seguenti metodi:
 - inizializzare una mensola di oggetti di tipo T e di capacità c
 - verificare se la posizione p è libera
 - mettere un oggetto nella posizione p (se libera)
 - stampare i titoli degli oggetti sulla mensola (nell'ordine in cui sono)

Esercizio 6

- Implementare in Java una classe `VettoreUtil<T>` che espone un metodo `min(T[] items)` che ritorna il minimo in un vettore di oggetti confrontabili tra loro:
 - Vettore di interi o decimali → valore minimo
 - Vettore di stringhe → prima parola in ordine alfabetico
 - Vettore di oggetti `Book()` → libro con il prezzo di basso
 - Si assuma che esiste una classe `Book()` che implementa un metodo `getPrice()`
- Sfruttiamo l'interfaccia `Comparable`

Esercizio 7 (sul libro Es. 2.10)

- Sia dato un vettore di n elementi che possono assumere solo i tre valori VERDE, BIANCO e ROSSO.
- Si vuole ordinare il vettore in tempo lineare in modo che tutti i “verdi” precedano tutti i “bianchi” e tutti i “bianchi” precedano tutti i “rossi”.
- Le uniche operazioni ammesse sono l’esame di un elemento e lo scambio di due elementi, dati i loro indici (non si possono usare vettori di appoggio)