



Map & Files

Alessandro Midolo, Ph.D. Student
alessandro.midolo@phd.unict.it



Tutorato Ingegneria del Software

A.A. 2021/2022



java.util.HashMap

Una HashMap è una collezione che associa ad una **CHIAVE** un **VALORE**

- Non possono esserci chiavi duplicate
- Per ogni chiave è possibile associare un solo valore
- Le HashMap sono composte da tre insiemi: l'insieme delle chiavi, l'insieme dei valori e l'insieme delle associazioni "chiave-valore"
- Non garantisce l'ordine degli elementi
- Non è sincronizzata

Metodi della classe HashMap

- **put(K chiave, V valore)** : associa alla chiave specificata il valore specificato, se la chiave è già presente, sostituisce il valore precedente
- **get(Object chiave)** : restituisce il valore per cui è associata la chiave specificata, ritorna **null** se la chiave specificata non è presente
- **remove(Object chiave)** : rimuove l'associazione chiave-valore prevista per la chiave specificata
- **keySet()** : ritorna un insieme contenente tutte le **chiavi**
- **values()** : ritorna un insieme contenente tutti i **valori**
- **entrySet()** : ritorna un insieme contenente tutte le coppie **chiave-valore**

java.io.File

La libreria **File** di Java mette a disposizione delle funzionalità (API) per la gestione di file system

E' possibile leggere e scrivere all'interno di un file, creare e cancellare un file

La maggior parte delle operazioni eseguite dalla classe File devono essere inserite all'interno di un costrutto ***try catch***

```
"File nuovoFile = new File("nuovoFile.csv");
```

Metodi della classe File

- **createNewFile()** : questa operazione permette di creare un nuovo file. Se l'operazione va a buon fine ritorna true, se il file è già presente, ritorna false
- **delete()** : elimina il file in questione; è possibile eliminare anche una directory ma deve essere vuota
- **exists()** : verifica se il file è presente nella cartella o meno
- **list()** : se il percorso specificato è una cartella, ritorna i nomi dei file presenti
- **mkdir()** : crea una nuova cartella
- **canRead()** or **canWrite()** : verifica se il file può essere letto o scritto (dipende dai permessi)

Scrittura e Lettura di un File

- Per **scrivere** all'interno di un file si utilizza una libreria java:
java.io.FileWriter
 - La libreria mette a disposizione il metodo ***write()*** che permette di scrivere all'interno di un file
- Per **leggere** il testo da un file si utilizza una libreria java:
java.io.BufferedReader
 - La libreria mette a disposizione il metodo ***readLine()*** la quale legge una riga del testo, fermandosi al carattere “\n” o “\r”