

# Test Gestione di Errori e Assertion

- 1) Quanti e quali tipi di errori sono presenti in java?

Esistono 2 tipi di errori principali, Error e Exception. La famiglia di Error di solito non è gestibile (si pensi ad errori di tipo "low memory" etc.) mentre la famiglia Exception è gestibile da dei gestori delle eccezioni

- 2) In caso di errore, java permette di uscire dal metodo con un percorso che non restituisce nessun valore?

È corretto, in java si può uscire dal metodo che non restituisce nessun valore

- 3) Cosa sono le eccezioni?

Le eccezioni sono degli errori che si possono presentare all'interno del programma e che possono essere gestiti

- 4) Quali sono le gerarchie che partano da Throwable? Scrivere in dettaglio.

Throwable è la classe in cui è definito tutto ciò che può essere sollevato dal compilatore. Ha due gerarchie, Error e Exception. Error descrive errori interni del sistema di esecuzione e quindi non sono gestibili. Exception descrive una situazione imprevista che si può avere nel programma e quindi sono gestibili.

- 5) Qual'è la differenza tra checked e unchecked Exception e come derivano?

Le exception unchecked sono quelle che derivano da una altra eccezione chiamata RuntimeException

Le exception checked sono quelle che non derivano da RuntimeException, queste sono le exception che vincolano il programmatore

- 6) Possiamo lanciare delle eccezioni personalizzate?

Sì, e per farlo bisogna creare una classe che estende la classe Exception.

- 7) Throw e return lavorano insieme per la gestione degli errori?

☐ Sì

☒ No

- 8) Come possiamo intercettare le eccezioni?

Le eccezioni vengono catturate dal blocco try-catch. Il blocco try è dove si mette il codice che potrebbe lanciare l'eccezione, nel blocco catch invece si inserisce il codice per la gestione dell'eccezione. A volte nel blocco catch viene rilanciata l'eccezione per far sapere al chiamante che ci è stato un errore

- 9) È possibile intercettare eccezioni multiple?

Sì, basta mettere multipli blocchi catch

- 10) Scrivere la giusta interpretazione per il meccanismo vincolato dell'eccezione

Gestione dell'eccezione => try/catch

Propagazione dell'eccezione => throws

11) Possiamo estendere il tipo di un'eccezione della classe super?

☐ Si

☒ No

12) A cosa serve finally?

Il blocco finally contiene il codice che verrà eseguito sia che venga lanciata una eccezione sia che il programma funzioni come ci si aspettava

13) Scrivere il blocco di codice dove lanci e gestisci una eccezione.

```
try{  
    //codice che lancia eccezione  
} catch(Exception e1){  
    //codice per risolvere eccezione  
}
```

14) Nel Try catch, quando possiamo utilizzare l'istruzione break, continue e return? A cosa servono?

Le istruzioni break, continue, return vengono eseguite solo dopo l'esecuzione del codice presente nel blocco finally. Break serve per interrompere il proseguimento di un loop, return ha il compito di ritornare al chiamante un valore specificato N.B: se la funzione ritorna un tipo void, allora return ha il compito di interrompere l'esecuzione del metodo; continue è utilizzato anch'esso all'interno dei loop e viene usato per passare alla prossima iterazione.

15) Cosa si intende per assertion e come lo possiamo usare in termini di linguaggio java?

Assertion è una keyword che viene utilizzata per verificare se i valori dei dati in una certa parte del programma è pari a quelli che ci si aspetta