

Soluzioni master parte teorica

Domanda 1:

- 1 punto: metodi e campi pubblici (e talvolta protected)
- 1 punto: documentazione (javadoc e contratti)
- 1 punto: nome e tipo dei campi
- 1 punto: firma dei metodi
- 1 punto: la firma e' composta da nome metodo ed elenco di parametri con loro tipo

Domanda 2:

- 1 punto: tipi passati come parametri a classi o metodi
- 1 punto: possono essere usati al posto degli altri tipi dichiarati
- 1 punto: in particolare, come tipo di parametri, valori di ritorno, campi, variabili locali
- 1 punto: si dichiarano tra < > dopo il nome della classe
- 1 punto: si puo' restringere il tipo generico con la keyword extends quando viene dichiarato in modo che debba essere sottotipo del tipo dato dopo l'extends

Domanda 3:

- 1 punto: try {blocco1} catch(Exception1 e) {blocco2} finally {blocco3}
- 1 punto: si esegue prima di tutto il blocco 1
- 1 punto: se termina normalmente si esegue il blocco3
- 1 punto: se lancia un'eccezione sottotipo di Exception1, si esegue il blocco 2 e poi il blocco 3
- 1 punto: se lancia un'eccezione non sottotipo di Exception1, si esegue il blocco 3 e si rilancia l'eccezione

Soluzioni master parte pratica

Esercizio 1:

- 1 punto: i tipi dei campi sono corretti
- 1 punto: costruttore e setter della forza dichiarano di lanciare l'eccezione `ValueException`
- 1 punto: l'eccezione `ValueException` e' lanciata correttamente nei casi previsti nel costruttore
- 1 punto: i campi ad eccezione della forza del terremoto non sono modificabili dopo la costruzione
- 1 punto: l'implementazione della modifica della forza e' corretta (vede il valore iniziale +- 1.0 e non l'ultimo valore!)

Esercizio 2:

- 1 punto: l'interfaccia `Damages` ha due metodi getter che ritornano valori interi
- 1 punto: `EqarthquakeWithDamages` extends `Earthquake` implements `Damages`
- 1 punto: il costruttore della sottoclasse e' corretto
- 1 punto: il costruttore della sottoclasse dichiara di lanciare la `ValueException`
- 1 punto: il setter dei morti/edifici caduti e' corretto e controlla di incrementare il numero

Esercizio 3:

- 1 punto: il costruttore inizializza una collection vuota
- 1 punto: esiste un metodo che permette di aggiungere un `Earthquake` ed e' corretto
- 1 punto: i due metodi iterano correttamente sulle collection
- 1 punto: i due metodi effettuano i controlli appropriati
- 1 punto: il secondo metodo controlla che sia instanceof `Damages` e fa il cast dell'oggetto per accedere il metodo che ritorna il numero di morti