

Soluzioni master parte teorica

Domanda 1:

- 1 punto: contratto come qualcosa che viene richiesto/garantito dal codice
- 1 punto: restringe gli stati possibili del programma/oggetto
- 1 punto: preconditione e' qualcosa che deve essere garantito quando si invoca un metodo
- 1 punto: postcondizione e' qualcosa che e' garantito da un metodo dopo la sua esecuzione
- 1 punto: invariante di oggetto e' qualcosa che vale sempre (prima o dopo l'esecuzione di suoi metodi) durante la vita di un oggetto

Domanda 2:

- 1 punto: interfaccia definisce un insieme di firme di metodi pubblici
- 1 punto: una classe deve dichiarare "implements <interface>"
- 1 punto: una classe deve implementare tutti i metodi definiti dall'interfaccia
- 1 punto: si possono avere anche campi ma questi sono static a final
- 1 punto: una implementazione di default permette di scrivere il codice di un metodo

Domanda 3:

- 1 punto: Collection rappresenta un "insieme" di elementi dello stesso tipo specificato tramite tipo generico
- 1 punto: in una Collection, si possono accedere, aggiungere e rimuovere elementi
- 1 punto: List puo' contenere duplicati, ciascun elemento "ha un indice"
- 1 punto: Set non puo' contenere duplicati
- 1 punto: List e Set sono sottotipo di Collection

Soluzioni master parte pratica

Esercizio 1:

- 1 punto: il tipo dei campi e' corretto
- 1 punto: ci sono solo i sei campi specificati
- 1 punto: il costruttore dichiara di lanciare la WeatherValueException
- 1 punto: c'e' il controllo che $\text{min} \leq \text{max}$ nel costruttore
- 1 punto: ci sono altri due controlli di integrita' dei dati, tra cui nome e data non null, vento e precipitazioni ≥ 0 , temperatura "plausibile"

Esercizio 2:

- 1 punto: l'interfaccia WeatherAlert contiene due getter la cui firma e' corretta
- 1 punto: la classe WindWeatherAlert implements WeatherAlert e implementa correttamente i metodi di WeatherAlert
- 1 punto: la classe WindWeatherAlert contiene due campi per memorizzare data e luogo e nient'altro
- 1 punto: il costruttore inizializza i propri campi in maniera appropriata
- 1 punto: il costruttore dichiara di lanciare l'eccezione e la lancia nei casi previsti

Esercizio 3:

- 1 punto: AlertSystem contiene un solo metodo statico e nessun campo
- 1 punto: la firma del metodo statico e' corretta (static e con parametri e tipi di ritorno giusti - Collection e non piu' specifici!)
- 1 punto: l'iterazione sugli alert passati come parametro e' corretta
- 1 punto: il confronto tra le date e' corretta
- 1 punto: la costruzione e popolamento del valore tornato dal metodo e' corretto