

Soluzioni master parte teorica

Domanda 1:

- 1 punto: si puo' usare su classi, metodi e campi
- 1 punto: un campo final non puo' essere assegnato nei metodi
- 1 punto: un campo final puo' essere assegnato solo nel costruttore o nella dichiarazione del campo stesso
- 1 punto: non e' possibile fare l'override di un metodo final
- 1 punto: non e' possibile estendere una classe final

Domanda 2:

- 1 punto: un oggetto di una classe puo' sostituire un oggetto di un'altra classe (in linea generale)
- 1 punto: l'interfaccia dell'oggetto sostituito deve essere piu' piccola dell'interfaccia dell'oggetto che lo sostituisce
- 1 punto: una classe che estende un'altra classe ha un'interfaccia piu' ampia
- 1 punto: una classe che estende un'altra classe e' suo sottotipo
- 1 punto: una classe che estende un'altra classe la puo' quindi sostituire

Domanda 3:

- 1 punto: un sistema nuovo e' compatibile con uno vecchio/legacy
- 1 punto: sulle librerie significa che forniscono una interfaccia piu' ampia della precedente
- 1 punto: ciascuna versione e' identificato da un numero univoco
- 1 punto: un numero di versione e' composto da piu' numeri distinti
- 1 punto: la major cambia l'interfaccia e non e' backward compatible

Soluzioni master parte pratica

Esercizio 1:

- 1 punto: c'è un'interfaccia Nazione con due metodi definiti e non implementati
- 1 punto: la firma dei metodi (parametri e valore di ritorno è corretta)
- 1 punto: c'è un'interfaccia NazioneAmicaInterfaccia che implementa Nazione
- 1 punto: l'interfaccia NazioneAmicaInterfaccia implementa il metodo ereditato con un default che ritorna true
- 1 punto: la classe NazioneAmica implementa NazioneAmicaInterfaccia
- 1 punto: la classe NazioneAmica ha un campo di tipo stringa che memorizza il nome della nazione
- 1 punto: la classe NazioneAmica implementa il metodo che ritorna il nome della nazione correttamente
- 1 punto: il campo che contiene il nome della nazione non può essere modificato dopo la costruzione dell'oggetto (final e/o private e mai assegnato se non nel costruttore)

Esercizio 2:

- 1 punto: la classe UnioneDiNazioni contiene un campo che memorizza una collection con tipo generico Nazione
- 1 punto: il campo collection è inizializzato con una qualche collection vuota
- 1 punto: viene rimossa correttamente la nazione (si controlla il nome o tramite equals/hashCode)
- 1 punto: il metodo di rimozione ritorna true solo se la classe è stata effettivamente rimossa
- 1 punto: viene controllato correttamente se tutte le nazioni ritengono la nazione data amica
- 1 punto: viene controllato che la nazione non sia già presente nell'unione (tramite equals/hashCode oppure controllando i nomi delle nazioni già presenti)
- 1 punto: il metodo ritorna true solo se tutte le nazioni sono amiche ed effettivamente la nazione non è presente nell'unione