Soluzioni master parte teorica

Domanda 1:

- 1 punto: a campi, metodi e classi
- 1 punto: una classe final non puo' essere estesa
- 1 punto: non si puo' fare l'override di un metodo final
- 1 punto: non si puo' assegnare piu' di una volta un campo final
- 1 punto: un campo final deve essere inizializzato nel costruttore (o nella sua definizione)

Domanda 2:

- 1 punto: si' si puo' implementare un metodo
- 1 punto: e' necessario definirlo di default
- 1 punto: si' si possono definire campi
- 1 punto: i campi sono static e final
- 1 punto: l'implementazione dei metodi non ha restrizioni

Domanda 3:

- 1 punto: uno e' un tipo valore e l'altra e' un tipo reference (numero/oggetto)
- 1 punto: Integer eredita i metodi di Object
- 1 punto: Integer e' un wrapper di int
- 1 punto: si' e' possibile
- 1 punto: autoboxing

Soluzioni master parte pratica

Domanda 1:

- 1 punto: struttura dati adeguata per memorizzare le informazioni richieste (e.g., Map per gli orari)
- 1 punto: informazioni dell'orario settate all'interno del costruttore
- 1 punto: identificatore settato all'interno del costruttore
- 1 punto: campi privati e nessun setter
- 1 punto: metodo tipo Date getTime(String stazione) pubblico e metodo tipo int getTrainNumber() pubblico
- 1 punto: c'e' un metodo tipo Collection<String> getClasses() in treno
- 1 punto: il metodo getClasses e la classe Treno sono astratti
- 1 punto aggiuntivo: i campi sono final

Domanda 2:

- 1 punto: le due classi estendono Treno
- 1 punto: l'implementazione di getClasses in TrenoRegionale e FrecciaRossa sono corrette
- 1 punto: il metodo getClasses nelle due implementazioni ritorna sempre lo stesso valore (collezione con quegli elementi, anche se ogni volta crea una nuova istanza va bene)
- 1 punto: le classi TrenoRegionale e FrecciaRossa hanno dei costruttori che invocano semplicemente il costruttore della superclasse
- 1 punto aggiuntivo: utilizzo annotazione @Override

Domanda 3:

- 1 punto: la classe biglietto ha un costruttore che riceve le informazioni richieste
- 1 punto: la classe biglietto ha i campi richiesti private (classe, id treno, e boolean per la validazione)
- 1 punto: viene definita un'eccezione ad hoc per gestire il biglietto gia' validato
- 1 punto: il metodo che valida contiene il check del tipo dinamico del treno e setta il campo booleano appropriatamente