

Programmazione a oggetti - Java

Esercitazione 6

Contatti:

Prof. Angelo Gargantini – angelo.gargantini@unibg.it

Dott. Marco Radavelli – marco.radavelli@unibg.it

Eccezioni

Esercizio 31

- Scrivi un programma che inserito un numero intero da tastiera stampi a video il risultato della divisione di tale numero per un altro numero generato casualmente, compreso fra 0 e 3, intercettando l'eventuale eccezione della divisione per 0.

Esercizio 32

- Scrivi un metodo che prenda in input un'unica stringa contenente un elenco di voti separati da punto e virgola, li memorizzi in un array e ne calcoli media e massimo, intercettando l'eventuale eccezione per i voti non validi (per esempio negativi, o maggiori di 30).

Esercizio 33

- A partire dall'Esercizio 28 (ordinazioni ai tavoli di un ristorante), definisci, lancia e cattura opportunamente delle eccezioni nei seguenti casi:
 - Replicazioni di ID (nel caso di Tavolo e Piatto)
 - Inserimento di una persona con lo stesso nome e cognome di una esistente
 - Aggiunta di una persona a un tavolo già pieno
 - Aggiunta e rimozione di un piatto il cui ID è inesistente.
- Scrivere una classe di prova in cui si crea almeno un tavolo, e si testano tutti i metodi che lanciano eccezioni.
- Come eccezioni, lancia delle RuntimeException

Esercizio 34

- A partire dall'esercizio 29 (gestione voli), aggiungere almeno tre eccezioni di tipo `BoardingException`:
- `ExceedingPassengersCapacityException`: quando si tenta di aggiungere più passeggeri rispetto ai posti disponibili nella classe richiesta (nella stampa dell'eccezione, inserire anche la classe richiesta)
- `PassengerNotFoundException`: quando si vuole rimuovere un passeggero dal volo, tramite in IDBiglietto inesistente fra i passeggeri registrati su quel volo.
- `NoPassengersException`: si vuol fare decollare un volo senza alcun passeggero registrato.

Esercizio 35

- Aggiungere eccezioni gestore di negozi dell'esercizio 30, ogniquaľvolta:
- Viene inserito un ID uguale (SameIDException)
 - A livello di negozio (SameStoreIDException)
 - A livello di prodotto in un negozio (SameItemIDException)
- Un cliente acquista più della disponibilità (ExceedingAvailabilityException)
- Ricorda che un negozio può anche rifornire un prodotto (eventualmente lanciare eccezione NegativeRefillException se viene passata una quantità negativa) e rimuoverlo dalla lista dei suoi prodotti.
- Scrivi una classe di prova