HOMEWORK 3

Java Collection Framework

Esercizio 0

- Chi non avesse concluso la scrittura dei test, lo faccia in questo homework, prima di fare le modifiche al codice

Esercizio 1

- Sostituire tutti gli array utilizzati nelle classi `Stanza` e `Borsa` con opportune collezioni (`List`, `Set`, `Map`)
  - Assumere che non possano esistere due oggetti `Attrezzo` con lo stesso nome in stanze dello dello stesso `Labirinto`
  - Eliminare il vincolo che al massimo 10 attrezzi possano essere collocati nella borsa (ma mantenere quello sul peso massimo)
  - provare ad usare (in alternativa) `List` e `Map` per implementare la collezione di attrezzi nella borsa (vedi POO-14).
    Quale implementazione risulta più semplice?
- Queste modifiche impattano solo sull'implementazione: dopo averle effettuate verificare con i test sviluppati negli homework precedenti la correttezza del codice

Esercizio 2

- Rivisitare il codice delle nuove tipologie di stanze introdotte nel precedente homework:
  - La stanza buia: se nella stanza non è presente un attrezzo con un nome particolare (ad esempio "lanterna") il metodo `getDescrizione()` di una stanza buia ritorna la stringa "*qui c'è un buio pesto*"
  - La stanza bloccata: una delle direzioni della stanza non può essere seguita a meno che nella borsa non sia presente un oggetto con un nome particolare (ad esempio "passepartout")

- Subiscono modifiche consequenziali ai cambiamenti effettuati nella realizzazione di `Stanza`?

Esercizio 2 (continua)

- Ampliare i test di tutte le classi della gerarchia che ha radice in `Stanza`: `Stanza`, `StanzaMagica`, `StanzaBuia` e `StanzaBloccata`

- Eliminare dal codice delle classi `Borsa`, `Stanza`, `StanzaMagica`, `StanzaBuia` e `StanzaBloccata` (estensioni di `Stanza`) ogni ciclo di ricerca da un collezione (ad es. di un attrezzo per nome, o di una stanza per direzione)

Esercizio 3

- Aggiungere alla classe `Borsa` dei metodi di interrogazione del suo contenuto:
  - `List<Attrezzo> getContenutoOrdinatoPerPeso();`

restituisce la lista degli attrezzi nella borsa ordinati per peso e quindi per nome a parità di peso

  - `List<Attrezzo> getContenutoOrdinatoPerNome();`

restituisce la lista degli attrezzi nella borsa ordinati per nome

  - `Map<Integer,Set<Attrezzo>> getContenutoRaggruppatoPerPeso()`

    restituisce una mappa che associa ad un intero con un insieme (comunque non vuoto) di attrezzi: tutti gli attrezzi nell'insieme hanno lo stesso peso ed è pari all'intero che figura come chiave nella mappa

- Utilizzare questi metodi per migliorare la stampa del contenuto della `Borsa`

Esercizio 3 (notazione es.)

- Si utilizzi “piombo”:10 per indicare un riferimento ad un oggetto `Attrezzo` di nome “piombo” e peso 10
- Per brevità scriviamo *piombo* al posto di “piombo”:10 quando non è utile ripetere i dettagli
- Si utilizzi
  - { *piombo*, *piuma*, *libro*, *ps* } per indicare un `Set` di attrezzi
  - [*piuma, libro, ps, piombo*] per indicare una `List` di attrezzi
  - (5, { *libro,ps* }) per indicare una coppia chiave/valore di una `Map<Integer,Set<Attrezzi>>`

Esercizio 3 (esempio)

- Si consideri una `Borsa` contenente questo insieme di riferimenti ad oggetti `Attrezzo`:

{“piombo”:10, “piuma”:1, “libro”:5, “ps”:5”}

- Allora i metodi di cui prima, invocati sullo stesso oggetto `Borsa`:

— `List<Attrezzo> getContenutoOrdinatoPerPeso();`

deve restituire: [*piuma, libro, ps, piombo*]

— `List<Attrezzo> getContenutoOrdinatoPerNome();`

deve restituire: [*libro, piombo, ps, piuma*]

— `Map<Integer,Set<Attrezzo>> getContenutoRaggruppatoPerPeso()`

deve restituire una `Map` contenente tutte e sole le seguenti coppie chiave/valore: (1, { *piuma* }) ; (5, { *libro, ps* }) ; (10, { *piombo* }),

Esercizio 4

- Utilizzando JUnit, scrivere una batteria di test-case *minimali* per verificare la correttezza delle soluzioni prodotte nell'esercizio precedente
  - minimali: ovvero facenti utilizzo delle più semplici collezioni possibili (più piccole possibili e con Attrezzi di nomi/pesi in configurazioni minimali) utili alla verifica

- Ampliare i test-case di sopra con altri non minimali per migliorarne la copertura

TERMINI E MODALITA' DI CONSEGNA

- La soluzione deve essere inviata al docente entro le 21:00 del 25 maggio 2014 come segue:
  - Svolgere in gruppi di max 2 persone
  - Esportare (con la funzione File->Export di Eclipse) il progetto realizzato nel file **homework3.zip**
  - Inviare il file **homework3.zip** all'indirizzo di posta elettronica poo.roma3@gmail.com
  - Nel corpo del messaggio riportare: eventuali malfunzionamenti noti, ma non risolti, osservazioni, critiche, suggerimenti in merito all'esperienza
  - L'oggetto del messaggio deve iniziare con la stringa **[2014-HOMEWORK3]** seguita dalla matricola/e mittente/i