Dipartimento di Informatica Sistemistica e Comunicazione
Corso di Laurea in Informatica

Programmazione 2

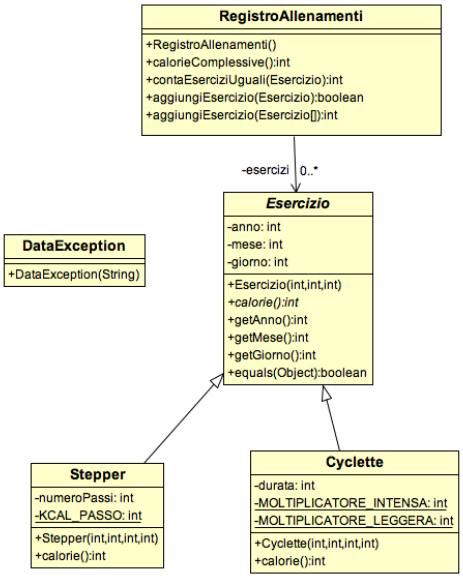
18 Gennaio 2021 – Esame completo

Testo parte di pratica

Una persona svolge degli esercizi per tenersi in forma e registra le sue sessioni di allenamento in un registro in modo da poter fare delle statistiche. Ogni tipologia di esercizio è caratterizzata dal proprio metodo di calcolo delle calorie spese. Si implementi la specifica indicata di seguito. Si provino le classi realizzate con JUnit utilizzando i test forniti nella classe `EsameTest`.

Suggerimento: si eviti di eseguire i test solo alla fine del lavoro, quando ormai sarebbe tardi per apportare correzioni; Piuttosto si eseguano i test man mano che si procede con l'implementazione, per verificare incrementalmente il lavoro via via fatto.

Le classi devono essere implementate in modo coerente al diagramma e alla specifica che seguono. **I metodi sono tutti specificati dal diagramma**: non occorre aggiungere altro. Le operazioni **mostrano il tipo ma non il nome** di ogni parametro che può essere scelto arbitrariamente dallo studente purché il **nome scelto rifletta il significato del parametro**. La classe `EsameTest` è fornita e deve essere usate senza modifiche.



Descrizione delle classi

- `DataException` è una classe che implementa una eccezione che puo' essere sollevata nel codice
- Classi `Esercizio`, `Stepper` e `Cyclette`

  `Esercizio` è una classe astratta che modella un esercizio fisico caratterizzato da una data espressa attraverso una tripletta di interi (anno, mese e giorno) e da un metodo di calcolo delle calorie (`calorie`) che è definito astratto e che ogni sottoclasse implementa in maniera opportuna.

  Il costruttore di `Esercizio` crea un esercizio inizializzando la data. A tal proposito accetta in ingresso nell'ordine specificato di seguito 3 interi: anno (un numero intero compreso tra 2000 e

2020, estremi inclusi), mese (un numero intero compreso tra 1 e 12, estremi inclusi) e giorno (un numero intero compreso tra 1 e 31, estremi inclusi). Se i valori passati in ingresso non rientrano negli intervalli specificati lancia una eccezione di tipo `DataException`. Non viene richiesto un controllo relativamente all'esattezza del numero di giorno rispetto al mese. Ad esempio, se viene inserito 2020 2 30, il sistema lo accetta nonostante febbraio abbia 28 giorni. L'importante è che i singoli valori di anno, mese e giorno siano negli intervalli specificati sopra.

Due esercizi sono uguali (metodo `equals`) se sono dello stesso tipo e se hanno permesso di consumare lo stesso numero di calorie.

La classe Esercizio implementa anche tre metodi getter che ritornano il valore di giorno, mese ed anno.

Si considerano in particolare due tipi di esercizio: `Stepper` e `Cyclette` che si differenziano per il modo in cui permettono di bruciare calorie.

- `Stepper` restituisce il valore calcolato come prodotto del numero di passi (`numeroPassi`) per le calorie bruciate a passo (costante `KCAL_PASSO`, da dichiarare ed inizializzare a 2).
- `Cyclette` restituisce il valore calcolato rispetto alla durata dell'allenamento: se vengono eseguiti fino a 20 minuti, allora le calorie consumate sono calcolate come prodotto dei minuti per la costante `MOLTIPLICATORE_LEGGERA` (da dichiarare ed inizializzare a 2), se sono superiori a 20, le calorie consumate sono calcolate come prodotto dei minuti per la costante `MOLTIPLICATORE_INTENSA` (da dichiarare ed inizializzare a 3).
- Al fine di permettere il calcolo, i costruttori delle classi `Stepper` e `Cyclette`, oltre ad accettare in ingresso l'anno, il mese e il giorno, accettano in ingresso un ulteriore parametro che rappresenta il numero di passi per la classe `Stepper` e la durata in minuti della pedalata per la classe `Cyclette`.

- Classe `RegistroAllenamenti`

  La classe permette di memorizzare gli esercizi svolti attraverso un `ArrayList`.

  Dettagli ulteriori:

  - Il costruttore inizializza opportunamente un `RegistroAllenamento`
  - Il metodo `calorieComplessive` restituisce le calorie complessive consumate grazie a tutti gli esercizi registrati.
  - Il metodo `contaEserciziUguali` restituisce il numero totale di esercizi che sono uguali (secondo il metodo `equals`) all'esercizio passato in ingresso come parametro.
  - Il metodo `aggiungiEsercizo` che accetta in ingresso un esercizio aggiunge al registro l'esercizio solo se è diverso da `null`. Restituisce `true` se l'esercizio è stato inserito, `false` in caso contrario.
  - Il metodo `aggiungiEsercizo` che accetta in ingresso un array di esercizi aggiunge al registro i vari esercizi nell'array solo se sono diversi da `null`. Restituisce il numero di esercizi che ha inserito nel registro.