

RELAZIONE PROGETTO PROGRAMMAZIONE AD OGGETTI

Nome del progetto: *QFit*



Victor Dutca

Matricola: 1122137

Laurea in Informatica



DIPARTIMENTO
MATEMATICA

Anno Accademico 2019/2020

Indice

1	Ambiente di sviluppo	2
2	Scopo del progetto	2
3	Gerarchie utilizzate	3
3.1	Gerarchia della logica	3
4	Gerarchia della Vista	4
5	Utilizzo del codice polimorfo	5

1 Ambiente di sviluppo

- Sistema operativo
 - Ubuntu GNU/Linux 19.04
- Compilatore
 - Ubuntu: *g++* 8.3.0
- Versione Qt: 5.5.

2 Scopo del progetto

L'applicazione *QFit* permette all'utente di inserire e tenere traccia dei propri allenamenti che possono essere di quattro tipologie:

- Corsa
- Ciclismo
- Nuoto
- Triathlon

Una volta inseriti i dati relativi a ciascun allenamento l'utente visualizza le calorie consumate durante e la quantità dei grassi bruciati durante tutto lasso di tempo indicato. L'applicativo permette seguentemente di modificare l'allenamento, visualizzare i dati rispettivi all'allenamento e infine eliminarlo. Il tutto avviene tramite lettura e scrittura da file *.xml*. Il programma fa uso di un contenitore templetizzato *Dlista <Workout>*.

3 Gerarchie utilizzate

Sono presenti due tipi di gerarchie per la parte logica e grafica rispettivamente.

3.1 Gerarchia della logica

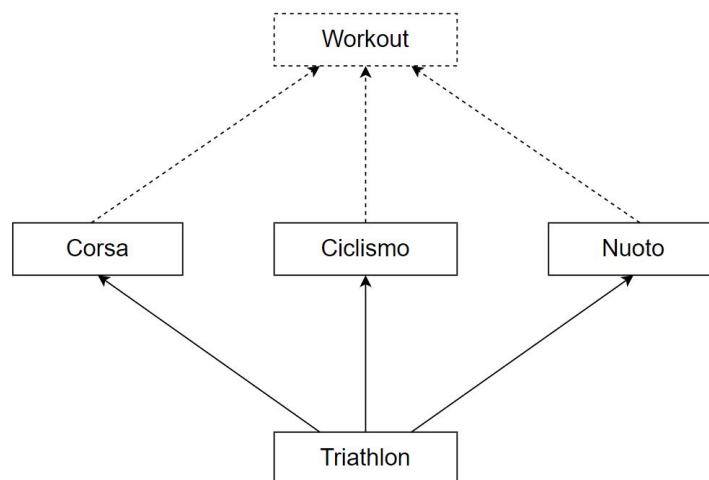


Figura 1: Gerarchia parte logica

Workout è la classe base virtuale astratta la quale dà forma ad un generico allenamento inserito dall'utente tenendo conto della durata di quest'ultimo. In base alla durata vengono concretizzati tutti gli altri allenamenti mediante la definizione dei seguenti metodi virtuali:

- `~Workout() = default;`
- `unsigned int GrassiBruc() const =0;`
- `unsigned int calorie() const = 0;`
- `bool operator==(const Workout &) const;`
- `bool operator<=(const Workout &) const;`
- `bool operator>=(const Workout &) const;`

- `unsigned int get_durata() const;`
- `void set_durata(unsigned int);`

Come menzionato in precedenza, dalla classe madre *Workout* derivano le seguenti classi concretizzate: Ciclismo, Corsa, Nuoto e Triathlon. Tutti metodi virtuali delle classe base vengono overrideati ad eccezione del distruttore.

Il metodo *GrassiBrucchi()* indica la quantità di grassi bruciati durante lasso di tempo dell'allenamento.

Il metodo *calorie()* calcola le calorie consumate dall'allenamento.

4 Gerarchia della Vista

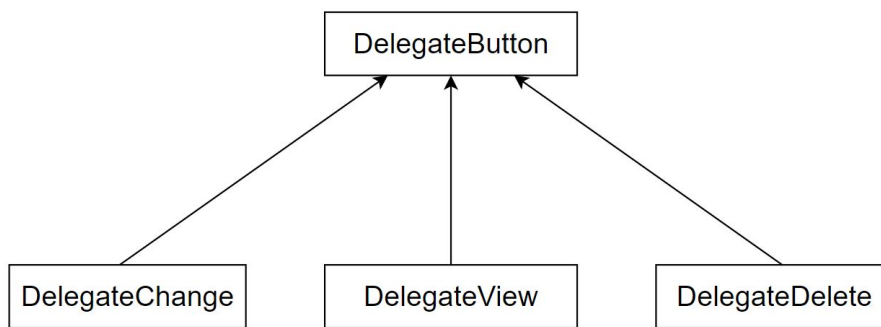


Figura 2: Gerarchia parte logica

DelegateButton è la classe base della gerarchia della vista la quale a sua volta è derivata da *QItemDelegate* dalla quale i metodi *paint()*, *editorEvent()* e i distruttore sono overrideati in *DelegateButton*. Le sottoclassi derivate da *DelegateButton* sono:

- 1) *DelegateView*: gestisce l'input per la visualizzazione del record selezionato nella *tableView* iniziale;
- 2) *DelegateChange*: gestisce l'input per la modifica del record selezionato nella *tableView* iniziale;
- 3) *DelegateDelete*: gestisce l'input per la cancellazione del record selezionato nella *tableView* iniziale.

5 Utilizzo del codice polimorfo

I metodi virtuali polimorfi della classe *Workout* vengono chiamati nella seguente parte del codice:

- > **get_durata()**: linea 40 della classe *ModelWorkout*;
- > **calorie()** linea 42 della classe *ModelWorkout*;
- > **GrassiBruc()** linea 44 della classe *ModelWorkout*;
- > **operator==** riga 119 della classe *qfitciclismo*;
- > **operator==** linea 119 della classe *qfitcorsa*;
- > **operator==** linea 109 della classe *qfitnuoto*;
- > **operator==** linea 246 della classe *qfittriathlon*.

6 Manuale utente