



Escola de Engenharia  
Departamento de Informática

Licenciatura em Engenharia Informática

# Projecto Java - FitnessUM

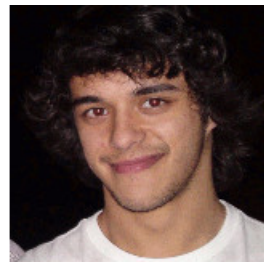
Programação Orientada aos Objectos



69303  
Bruno Pereira



66822  
Miguel Guimarães



69854  
João Mano

Braga, Junho de 2014

# Conteúdo

<b>1</b>	<b>Estrutura da aplicação</b>	<b>2</b>
1.1	Actividades . . . . .	2
1.1.1	Classe abstracta Activity . . . . .	3
1.1.2	Indoor,Outdoor e actividades desportivas . . . . .	3
1.1.3	Comparadores e Interfaces . . . . .	3
1.2	Utilizadores . . . . .	4
1.2.1	Classe abstracta Person . . . . .	4
1.2.2	Classes User e Admin . . . . .	5
1.2.3	Comparators . . . . .	5
1.3	Eventos . . . . .	5
1.4	Classe abstracta Event . . . . .	5
1.5	. . . . .	5

# 1 Estrutura da aplicação

## 1.1 Actividades

Foram definidas as seguintes actividades desportivas para a nossa aplicação:

- Yoga
- Aerobics
- Swimming
- IndoorCycling
- Handball
- Basketball
- TableTennis
- Boxing
- Badminton
- VolleyBallIndoor
- Football
- VolleyBallBeach
- Running
- Skating
- Sailing
- Walking
- Tennis
- Skiing
- Cycling
- MountainBiking
- Orienteering
- Snowboarding
- Polo

Para a implementação destas actividades foi usada a seguinte estrutura:

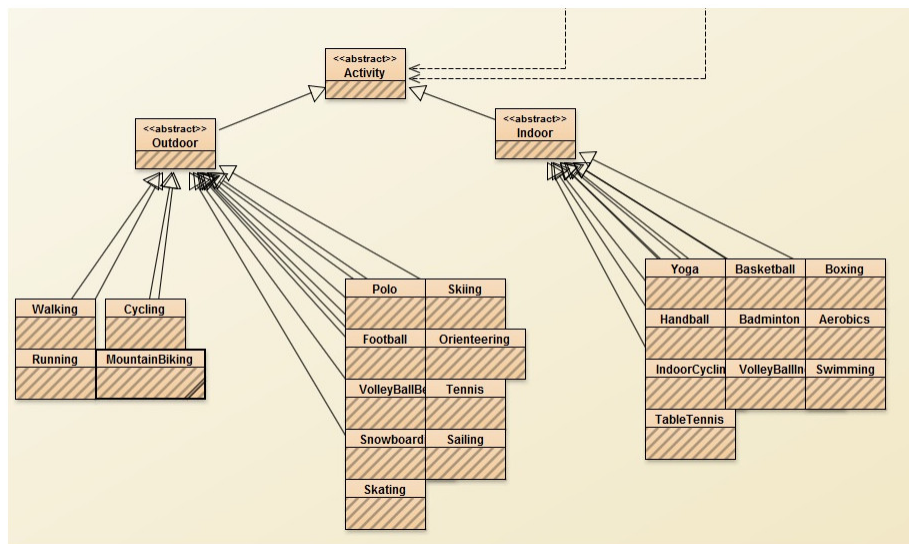


Figura 1: Estrutura das actividades

### 1.1.1 Classe abstracta Activity

Esta é a classe mais abstracta que contém o conceito de actividade. Contém variáveis comuns a todas as actividades:

- *String name*, nome da actividade criada.
- *GregorianCalendar date*, data de quando se realizou a actividade.
- *double timeSpent*, tempo gasto na actividade.
- *double calories*, campo preenchido pela aplicação de uma fórmula.

tal como os construtores, *getters* e *setters*.

### 1.1.2 Indoor,Outdoor e actividades desportivas

Todas as actividades desportivas tem um aspecto importante,o clima caso sejam praticadas ao ar livre. Devido a este aspecto foram criadas duas classes abstractas,subclasses de *Activity*,para essa distinção.

- Outdoor,contém a variável: *String weather*
- Indoor

Todas as actividades desportivas são subclasses de *Indoor* ou *Outdoor* como exemplificado na figura 1.

### 1.1.3 Comparadores e Interfaces

Para organizar as actividades criaram-se dois tipo de comparadores:

- CompareActivity- Compara a actividade pela data da realização da mesma.
- CompareActivityByTime- Compara a actividade pelo tempo gasto na realização desta.

Como certos desportos usam

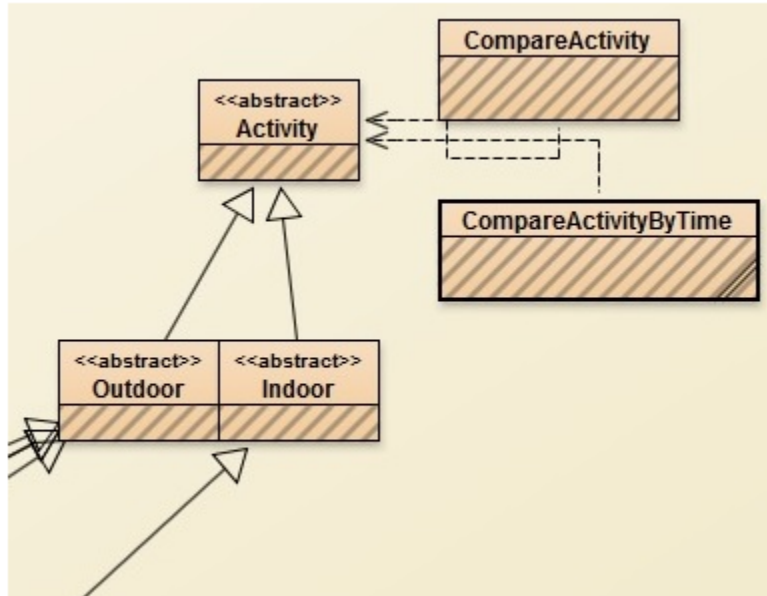


Figura 2: Comparador Activity

## 1.2 Utilizadores

Para distinguir utilizadores regulares de administradores criou-se a seguinte estrutura:

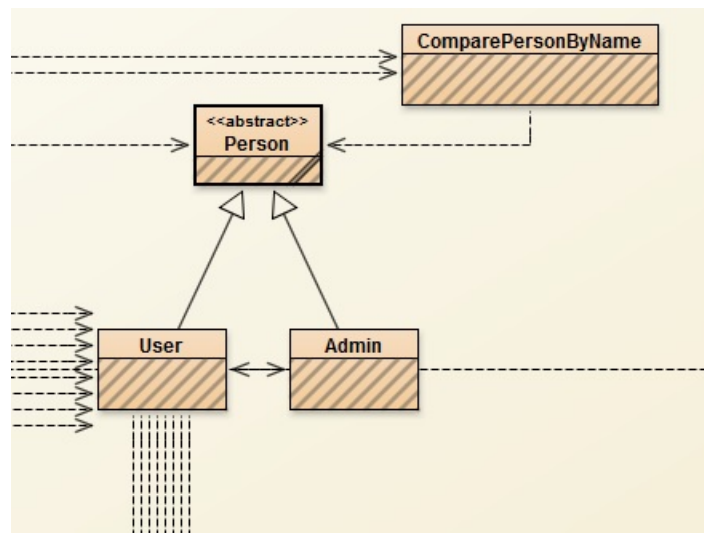


Figura 3: Estrutura das classes User e Admin

### 1.2.1 Classe abstracta Person

Classe geral para todo tipo de utilizador. As suas variáveis são:

- *String email;*

- *String password;*
- *String name;*
- *char gender;*
- *GregorianCalendar dateOfBirth;*

### 1.2.2 Classes User e Admin

As subclasses de Person referem-se a dois possíveis tipos de utilizador, utilizador normal ou utilizador com privilégios de administrador.

A classe Admin não tem métodos ou variáveis adicionais, visto que este tipo de utilizador apenas opera sobre a base de dados da aplicação.

A classe User adiciona as seguintes variáveis:

- *int height;*
- *double weight;*
- *String favoriteActivity;*
- *TreeSet<Activity> userActivities;*
- *TreeSet<String> friendsList;*
- *TreeMap<String, ListRecords> records;*
- *TreeSet<String> messageFriend;*

Respectivos métodos *getters* e *setters*, construtores e métodos auxiliares para a gestão de amigos/pedidos de amizade, recordes pessoais, das suas actividades e estatísticas relevantes. Ainda contém funções auxiliares para a simulação de eventos.

### 1.2.3 Comparators

O tipo Person tem apenas um comparator:

- ComparePersonByName - que ordena por ordem alfabética do seu nome.

## 1.3 Eventos

### 1.4 Classe abstracta Event

Classe com o conceito mais abstracto de Evento, contém as variáveis *name*, *tipoActivity*, *location*, *maxParticipants*, *participants*, *deadline*, *date*, *duration*, *participantsList*, *ranking*, *desistentes* e *simula*, respetivos *getters* e *setters* e os vários contrutores. Ainda tem métodos auxiliares para adicionar um *User*, *ranking*,

## 1.5