0.1 Actividades

0.1.1 Classe abstracta *Activity*

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

- *String name*;
- *GregorianCalendar date*;
- *double timeSpent*;
- *double calories*;

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

0.2 Utilizadores

0.2.1 Classe abstracta *Person*

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

- *String email* - Email do utilizador;
- *String password* - Password da conta;
- *String name* - Nome do utilzador;
- *char gender* - Género do utilizador;
- *GregorianCalendar dateOfBirth* - Data de nascimento do utilizador;

e contém os métodos construtores *getters* e *setters*

0.2.2 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* - Actividades realizadas pelo utilizador;
- *TreeSet¡String¿ friendsList* - Lista dos amigos do utilizador;
- *TreeMap¡String, ListRecords¿ records* - Lista dos seus recordes pessoais;
- *TreeSet¡String¿ messageFriend* - Lista de pedidos de amizade;

Respectivos métodos *getters* e *setters*, construtores e métodos auxiliares para gerir os seus amigos, recordes, as suas actividades e estatísticas relevantes. Ainda contém funções auxiliares para a simulação de eventos.

0.2.3 Comparador

O tipo *Person* tem apenas um comparador:

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

0.2.4 Statistics

A classe *Statistics* é usada para mostrar ao utilizador dados relevantes das suas actividades, estes podem ser descriminados por um dado mês ou por um ano. As suas variáveis são:

- *double timeSpend* ;
- *double calories*;
- *double distance*;

contém os respectivos métodos *getters* e *setters* e construtores.

0.2.5 Classe abstracta *Record*

Esta classe representa todos os recordes que o utilizador pode bater. Contém apenas uma variável:

- *String name*;

métodos construtores, *getName()* e *isEmpty()* que verifica se esse recorde existe ou não.

0.2.6 *DistancePerTime* e *TimePerDistance*

Estas classes simboliza os dois diferentes tipos de recordes.

DistancePerTime é um recorde em que o objectivo é fazer a maior distância para um dado tempo. As suas variáveis são:

- *double recordTime* - Tempo do recorde;
- *double distance* - Distância registada;

Enquanto que *TimePerDistance* representa um recorde de menor tempo para uma certa distância. As suas variáveis são:

- *double recordDistance* - Distância do recorde;
- *double time* - Tempo registado;

Estas duas classes têm os mesmos métodos, no entanto os métodos *update()* e *setStatistics()*, estão implementados de maneiras diferentes, tendo em conta que em *DistancePerTime*, quanto maior a distância melhor é o recorde, e no caso do *TimePerDistance*, o melhor recorde é o de menor tempo.

0.2.7 *ListRecords*

Classe que agrupa todos os recordes de uma actividade. Tem como variáveis:

- *String name* - Aqui o nome simboliza o tipo de actividade (Ex: Running, Walking...);
- *ArrayList<Record> recs* - Lista dos recordes;

Tem implementado métodos construtores, *getters*, *setters* e ainda um método *updateList()* que aplica a função *update()* a todos os objectos *Record* da lista. (Substitui na lista original caso recorde da segunda lista seja melhor).

0.3 Eventos

0.3.1 Classe abstracta *Event*

Classe com o conceito mais abstracto de Evento, contém as variáveis:

- *String name* - Nome do evento;
- *String tipoActivity* - Tipo de actividade (Running, Walking, ...);
- *String location* - Onde se realiza a prova;
- *int maxParticipants* - Número máximo de participantes;
- *int participants* - Número actual de participantes ;
- *GregorianCalendar deadline* - Data limite de inscrição;
- *GregorianCalendar date* - Data de realização;
- *double duration* - Duração da prova; ¡— NÃO ESQUECER DE FALAR DESTA VARIÁVEL
- *TreeSet<User> participantsList* - Lista de participantes;
- *TreeSet<Ranking> ranking* - Classificação dos que acabaram a prova;
- *TreeSet<Ranking> desistentes* - Participantes que desistiram da prova;
- *TreeSet<Simulacao> simula* - Informação relevante para simular cada concorrente;

respectivos *getters* e *setters* e os vários construtores. Ainda tem métodos auxiliares para, adicionar um *User*, *Ranking* (desistente ou não) e *Simulacao* aos respectivos *Sets* e para mostrar a classificação geral do evento.

0.3.2 Tipo de Evento

Subclasses de Evento (*Marathon*, *HalfMarathon*, *MarathonBTT* e *Trail*), todas estas contem mais uma variável *distance*, que nos casos de *Marathon* e *HalfMarathon* são variáveis *final*, porque este tipo de eventos tem distâncias especificas. Não tem métodos auxiliares para além de *getDistance()*.

0.3.3 Simulação

Para guardar dados relevantes à simulação de cada utilizador para um evento, foi criada a classe *Simulacao*. A simulação de cada evento é feita actualizando os dados desta classe a cada km.

- *double tempoGeral* - Tempo acumulado do utilizado na realização da prova.
- *double tempoMedio* - Tempo médio por km.
- *int kmDesiste* - Número de km que o utilizador aguenta durante a prova.
- *User u* - Utilizador associado á simulação.

esta classe, para além dos métodos construtores e *getters* e *setters*, contém apenas um método *actualiza*, que simula a passagem de uma distância (passada como argumento), usando o tempo médio por km e aplicando um factor aleatório (usando *Math.random()*.

0.3.4 Ranking

Cada evento, para organizar a sua classificação, utiliza duas colecções de objectos da classe *Ranking*. Uma delas, usada para organizar todos os participantes, que concluíram a prova, por ordem de chegada, a outra onde estão os aqueles que não terminaram, organizados por número de quilómetros realizados.

Esta classe usa as seguintes variáveis:

- *double time* - Tempo de realizado no evento;
- *int km* - Número de quilómetros realizados;
- *User athlete* - Utilizador;

Das variáveis *time* e *km*, apenas uma irá ter algum valor para cada utilizador, visto que esta classe é usada para ordenar classificações finais, cada pessoa tem ou um tempo de conclusão do evento ou o número do quilómetro em que desistiu. *Ranking* contém os métodos *getters* e *setters* relevantes, construtores, e para além dos métodos essenciais, foram implementados dois métodos *toString* alternativos, para os dois casos.