



Universidade do Minho

Braga, Portugal

TRABALHO PRÁTICO - RELATÓRIO

TRIVIAL FITNESS

Programação Orientada a Objetos

Departamento de Informática

Engenharia Informática 2023/24

Equipa de Trabalho:

A104537 - Afonso Pedreira

A104344 - Dário Guimarães

A104534 - Hugo Rauber



Índice

1. Resumo do trabalho	1
2. Introdução	1
3. Arquitetura de classes utilizada	1
4. Diagrama de Classes	2
5. Descrição da aplicação	2
6. Conclusão e aspetos a melhorar	7

1. Resumo do trabalho

Aqui aborda-se a criação de uma aplicação de gestão de atividades físicas e planos de treino, no qual se refere a sua estrutura, funcionalidade e a capacidade de manter o estado e de o rever ao longo do tempo."

Palavras-chave: diagrama, classes, atividade, planos...

2. Introdução

Este relatório apresenta a conceção e implementação de uma aplicação de gestão de atividades físicas e planos de treino, enfatizando a arquitetura de classes, funcionalidades desenvolvidas e a capacidade de salvar o estado da aplicação. Destaca-se a abordagem de programação orientada a objetos, que foi imposta, a robustez do sistema e a sua capacidade de atender a uma variedade de necessidades de utilizadores.

3. Arquitetura de classes utilizada

Quanto à arquitetura de classes adotada para este projeto, priorizou-se uma estrutura coesa e modular, baseada em princípios de programação orientada a objetos (POO), como foi requerido nas diretrizes do enunciado. A hierarquia de classes foi organizada em diferentes pacotes, cada um representando uma parte fundamental do sistema.

No pacote **activity**, concentram-se as classes relacionadas às atividades físicas, como **DistanceActivity**, **RepetitionsActivity** e **WeightRepetitionsActivity**. Cada uma dessas classes, encapsula comportamentos específicos das atividades, como cálculo de calorias, repetições, pesos e atributos relacionados.

Dentro da pasta **user** vemos abrangidas as classes relacionadas aos diferentes tipos de utilizadores, como **AmateurUser**, **CasualUser**, e **ProfessionalUser**. Essas classes encapsulam informações individuais de cada utilizador, como nome, endereço e peso, além de implementarem métodos para cálculo do fator de multiplicação de aptidão física.

O pacote **controller** engloba as classes responsáveis pela lógica de controlo da aplicação, como **AppController**. Estas classes tratam da interação entre o utilizador e o sistema, incluindo funcionalidades como login, registo de utilizadores e controlo de sessão.

O **state** abarca as classes relacionadas ao estado da aplicação, o **AppState**. Aqui, armazena-se e trata-se do estado global da aplicação, incluindo informações sobre utilizadores, atividades e planos de treino.

Dentro do pacote **trainingPlan**, estão as classes **TrainingPlan** e **TrainingPlanActivity**, nos quais se podem tratar dos planos de treino e as atividades associadas de forma correta.

No **view**, encontra-se a classe **View** que é responsável pela interação com o utilizador. Contém também os **Handlers** de interação com o utilizador.

A **App**, no final, que contém o método **main** e coordena a execução da aplicação, interagindo com o utilizador e chamando os métodos apropriados.

O estado é guardado num ficheiro `appState.bin` da forma que foi imposta nas diretrizes do enunciado.

4. Diagrama de Classes

(Aqui podemos ver um diagrama geral repartido em pacotes que resume a arquitetura da aplicação. (acima vai o diagrama geral).

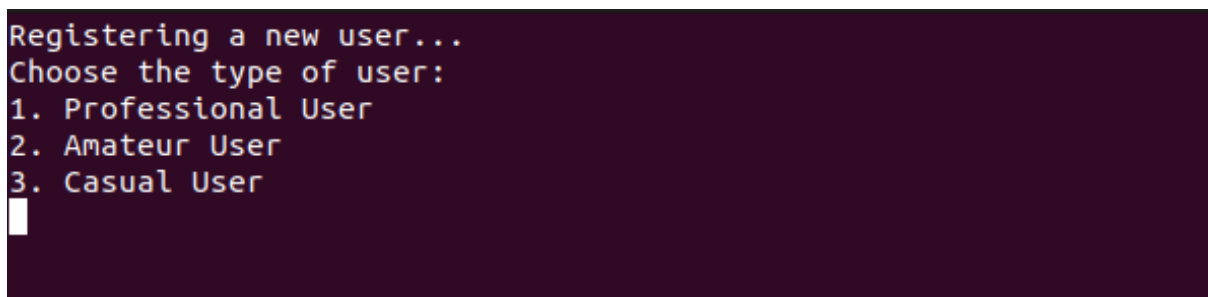
Aqui pode-se ver de uma forma mais específica, cada classe, os seus atributos e as ligações entre si.)

5. Descrição da aplicação

Inicialização da Aplicação: A aplicação é iniciada e temos as opções de interação disponíveis.



Registo de utilizadores: O utilizadores regista-se na aplicação. Pode-se registar como utilizador amador, casual ou profissional, dando as suas informações como nome, endereço, e-mail e peso. A aplicação verifica se o ID do utilizador já existe e, caso contrário, regista-o.



```
Enter a new user ID: user_test
Enter your name: user
Enter your address: address_example
Enter your email: user_test@gmail.com
Enter your weight (in kg): 81
```

Login de utilizadores: Após o registo ou em sessões subsequentes, o utilizador pode fazer login na aplicação fornecendo o seu ID de utilizador (neste caso `user_test`). A aplicação verifica se o ID de utilizador fornecido corresponde a um utilizador registado, e permite, ou não, o seu login, pedindo uma nova tentativa.

```
Login successful.

Welcome, user

1. View Training Plans
2. View Activities
3. Add New Activity
4. Add New Training Plan
5. Generate Training Plan
6. Advance Time
7. Check distance traveled
8. Check altimetry total
9. Check most famous activity type
10. Check status
11. Check User with most calories burned
12. Check User with most activities done
13. Save
0. Logout

Choose an option:
```

Agora é possível que o utilizador adicione uma atividade já feita. Adiciona uma atividade de distância (por exemplo).

```
1. Distance
2. Distance and Altimetry
3. Repetitions
4. Repetitions with Weight

Choose an activity type:
```

Coloca a data em que se realizou, a duração em minutos e os batimentos cardíacos médios. Depois, a atividade em questão.

```
1. Rowing
2. Running
3. Swimming
4. Surfing
5. Indoor Cycling
6. Walking

Choose an activity: █
```

Escolhe, por exemplo, `running` e os km percorridos, 12km neste caso.

```
Activity added successfully.
```

Adiciona outra atividade. (tipo: Repetições, atividade: pushups, repetições: 50).

Agora, podem-se verificar todas as atividades realizadas

```
Past activities:
    Running on 2024-05-10 <-> 12 Km of distance;
    Push Ups on 2024-05-09 <-> 50 repetitions;

Press any key to continue...
```

Adicionando uma nova atividade de distância na qual se tenham percorrido, por ex, 3 km. Vemos agora que a distância total percorrida entre duas datas (ou desde o início das atividades, se for preferido), é de 15km (a soma das duas atividades de distância).

```
Total distance traveled between 2024-03-01 and 2024-05-12 is 15 km
Press any key to continue...
```

Se adicionar agora, uma atividade de altimetria na qual se tenham subido 250m, será visível. Além de que a distância total, terá aumentado também. Isto é possível também com o resto dos tipos de atividades (adicionaremos uma atividade de repetição mais tarde para ser visível depois do `logout`).

```
Total altimetry climbed between the beggining and now is 250 mt.  
Press any key to continue...
```

É possível ver outras estatísticas:

```
Most famous activity type is Distance with 2 activities.  
Press any key to continue...
```

Figura 2: Atividade mais realizada pelo User

```
User status:  
    Number of past activities: 4;  
    Number of training plans: 0;  
    Total calories spent: 4009.0447000000004;  
    Average heart rate: 92;  
  
Press any key to continue...
```

Figura 3: Estatísticas gerais

É também possível avançar no tempo, por exemplo, aqui avançamos 360 dias.

```
Time advanced successfully. Current date: 2025-05-07  
  
Welcome, user  
  
1. View Training Plans  
2. View Activities  
3. Add New Activity  
4. Add New Training Plan
```

Se adicionarmos outro utilizador, `user2`, e lhe pusermos 5 atividades, podemos verificar quem a estatística de quem tem mais atividades de todos os users, fica a nome de `user2`.

```
Enter the ending date of the period (yyyy-MM-dd) or press enter to check until today:  
User user2 has the most activities between the beggining and now with 5 activities.  
Press any key to continue...
```

Podem-se verificar as atividades feitas deste utilizador.

```
Past activities:
  Running on 2024-04-12 <-> 58 Km of distance;
  Burpees on 2024-02-01 <-> 150 repetitions;
  Walking on 2024-01-01 <-> 12 Km of distance;
  Jump Rope on 2024-02-08 <-> 200 repetitions;
  Bench Press on 2024-04-05 <-> 45 repetitions with 30 Kg;

Press any key to continue...
```

Assim como dar Logout com este utilizador. Login com o `user_test` anterior e verificar que os dados foram guardados.

```
Past activities:
  Running on 2024-05-10 <-> 12 Km of distance;
  Push Ups on 2024-05-09 <-> 50 repetitions;
  Surfing on 2024-05-08 <-> 3 Km of distance;
  Mountain Bike on 2024-05-03 <-> 12 Km of distance and 250 Mt of elevation gain;

Press any key to continue...
```

Geração de planos de treino

Vamos agora criar um novo utilizador para demonstrar a criação de planos de treino automática e manual.

Podemos gerar um plano de treino novo de forma manual:

Inserimos o número de atividades pretendidas (4).

Colocamos as datas nas quais o plano de treino se vai encaixar, por exemplo, desde 2024-05-12 até 2024-05-19.

```
Enter the number of activities in the training plan: 4
Enter the beginning date of the training plan (yyyy-MM-dd): 2024-05-12
```

```
Enter the ending date of the training plan (yyyy-MM-dd): 2024-05-19
```

Novamente o tipo de atividade (Distance).

```
1. Distance
2. Distance and Altimetry
3. Repetitions
4. Repetitions with Weight

Choose an activity type:
```


Quantos dias queremos a atividade (7). E a distância a percorrer (8km). Adicionamos as restantes 3 atividades e o plano fica criado.

Agora vamos realizar de forma automática.

Colocamos a data 2024-05-20 até 2024-05-27.

O máximo de atividades por dia (3) e o máximo de atividades diferentes(7).

```
Enter the maximum number of activities per day: 3
Enter the maximum number of different activities: 7
```

O número máximo de atividades por semana (7) e o objetivo de calorias (1500).

Escolher a opção de atividades difíceis (yes):

```
Do you want to include hard activities? (yes/no):
```

Escolhemos a atividade Distance, e já temos um plano de distância semanal criado.

Podemos ver agora os dois planos de treino criados:

```
Training plans:

Training plan from 2024-05-12 to 2024-05-19 with 4 exercises:
    Running on SUNDAY;
    Climbing on WEDNESDAY;
    Squats on FRIDAY;
    Bench Press on MONDAY;

Training plan from 2024-05-20 to 2024-05-27 with 3 exercises:
    Surfing on MONDAY;
    Surfing on THURSDAY;
    Surfing on SUNDAY;

Press any key to continue...
```

6. Conclusão e aspetos a melhorar

Os maiores desafios para o grupo foram tentar garantir que o código seguisse os princípios de POO como o encapsulamento. Isto, enquanto tornávamos o código versátil para novas atividades e tipos de atividades serem adicionadas facilmente, evitando ter que navegar por todo o código para adicionar novas coisas.

O lado positivo foi termos a oportunidade de colocar em prática e em larga escala os conceitos de POO, permitindo treinar e aprofundar temas como o encapsulamento e abstração

Em termos de melhorias futuras: Seria ideal otimizar ainda mais a criação de novas atividades. Ainda assim o código esta um pouco preso aos tipos atuais de atividades.

Outro aspeto a melhorar seria a estrutura de dados usadas, que maioritariamente são sempre arrays. Poderíamos usar estruturas como árvores, hashes e heaps para as estatísticas. Em geral, outros aspetos que pudessem melhorar a performance do programa e a legibilidade e facilidade de escrever código.