Upskill - Java

Laboratório de Programação

Um clube desportivo regista informação básica acerca dos seus atletas para poder calcular o valor a pagar a cada atleta no final de cada mês. Para auxiliar a informatização deste processo, pretende-se que implemente em JAVA, um projeto com as classes necessárias para representar os atletas do clube desportivo.

Um atleta é caraterizado pelo: nome, número de identificação civil, género e idade. Os atletas também têm uma atividade associada que pode ser: *caminhada*, *corrida*, *ciclismo* ou *natação.*

Todos os atletas são sujeitos ao cálculo da Frequência Cardíaca Máxima (FCM) e da Frequência Cardíaca de Trabalho (FCT). A forma de cálculo da FCM é apresentada na Tabela 1 e, conforme se verifica, difere de acordo com a atividade e/ou género do atleta.

Tabela 1 - Cálculo da FCM

Atividade	Género	FCM
Caminhada **Corrida**	Feminino e Masculino	$208{,}75 - (0{,}73\ x\ idade)$
Ciclismo	Feminino	$189 - (0{,}56\ x\ idade)$
	Masculino	$202 - (0{,}72\ x\ idade)$
Natação	Feminino e Masculino	$204 - (1{,}7\ x\ idade)$

A FCT, cuja fórmula de cálculo se apresenta abaixo, envolve os valores da: FCM, Frequência Cardíaca em Repouso (FCR) e a percentagem de Intensidade do Treino (IT) necessária para o objetivo. A FCR é um valor que representa o número de batimentos cardíacos em repouso durante 1 minuto e a IT pode ter como valor: 0,6 quando o objetivo é a queima de gordura ou 0,75 quando o objetivo é trabalhar a capacidade cardiorrespiratória.

$$FCT = FCR + [\mathrm{IT}\ x\ (FCM - FCR)]$$

A disponibilidade para treinar e/ou competir faz com que os atletas possam ser considerados: profissionais ou não profissionais. No caso de serem não profissionais, podem ser ainda considerados: semiprofissionais ou amadores. Um atleta não profissional deve ter, também, associado a antiguidade (número de anos que está filiado ao clube). Todos os atletas têm associado um valor mensal relativo aos prémios que consigam arrecadar nas competições em que participam.

De seguida, são apresentadas algumas particularidades para cada tipo de atleta.

Atletas Profissionais: estes atletas recebem ao fim do mês o valor resultante da soma de duas parcelas: uma fixa e outra variável. A parcela fixa é diferente entre atletas e pode ser atualizada, enquanto que a parcela variável está dependente do valor mensal arrecadado em prémios pelo atleta. O valor da parcela variável é de 20% do valor mensal arrecadado. No entanto, o valor desta percentagem pode ser atualizado.

Atletas Semiprofissionais: estes atletas recebem ao fim do mês o valor resultante da soma de duas parcelas: uma fixa e outra variável. A parcela fixa é igual para todos estes atletas e pode ser atualizada, enquanto que a parcela variável está dependente da antiguidade. O valor da parcela variável é uma percentagem sobre o valor da parcela fixa, de acordo com a antiguidade (ver Tabela 2).

Atletas Amadores: estes atletas recebem ao fim do mês o valor resultante da soma de duas parcelas variáveis associadas ao valor mensal arrecadado em prémios pelo atleta. A primeira parcela é uma percentagem do valor mensal arrecadado, de acordo com a antiguidade (ver Tabela 2). A segunda parcela é também uma percentagem do valor mensal arrecadado, tendo um valor igual para todos os atletas de 7%. Por indicação do clube, todos os atletas amadores recebem pelo menos 5 euros por mês. No entanto, o valor da percentagem da segunda parcela e o valor mínimo a receber pode ser atualizado.

Tabela 2 - Percentagens mediante a antiguidade

Antiguidade	Percentagem[1]
[5,10] anos	**2%**
]10,20] anos	**8%**
mais que 20 anos	**20%**

As classes criadas (com exceção da classe principal) devem obedecer ao seguinte conjunto de especificações:

- implementação de construtores (pelo menos o construtor completo e o construtor sem parâmetros);

[1] Estes valores podem ser atualizados.

- implementação de métodos que sejam relevantes para aceder e modificar o valor dos atributos;
- reescrita do método *toString*;
- implementação do método equals.

Adicionalmente devem ser implementadas, nas respetivas classes, funcionalidades para o cálculo do valor a pagar mensalmente a cada atleta.

Apesar de, neste momento, existirem apenas estas três categorias de atletas poderão surgir outras no futuro.

Todos os atletas que auferem uma componente fixa (neste momento, apenas os Profissionais e Semiprofissionais) devem descontar, sobre essa componente, uma taxa imutável de 10% para o IRS.

Deverá existir também uma classe instanciável de nome **ClubeDesportivo** que tem por atributos o nome, a data de fundação (Data) e um contentor de objetos do tipo **Arraylist** para armazenar os atletas. Esta classe deverá permitir às suas instâncias disponibilizar as seguintes funcionalidades:

- Retornar o nome do clube;
- Inserir um novo atleta no contentor;
- Retornar uma lista de atletas, ordenada alfabeticamente por nome, usando a interface nativa do Java Comparable;
- Retornar uma lista de atletas, ordenada inversamente pelo valor dos prémios usando a interface nativa do Java Comparator;
- Retornar o valor total, para efeitos de IRS, da totalidade dos atletas;
- Retornar uma lista de atletas do clube, ordenada alfabeticamente por categoria, modalidade e nome.

Deve ser apresentado um diagrama UML para representar o descrito em cima.

Na classe principal, o código a implementar deve preencher os seguintes requisitos:

- Criação de uma instância da classe ClubeDesportivo;
- Armazenamento nessa instância de 3 objetos de cada uma das categorias (Profissional, Semiprofissional e Amador);
- Criação de listagens separadas, sobre o contentor, para:
  - obter o nome, a FCM e as FCT de cada atleta semiprofissional e amador;
  - obter o nome e o valor a pagar de cada atleta.

- Apresentação das quantidades de instâncias de atletas amadores e profissionais criadas, sem percorrer o contentor;
- Calcular e apresentar o valor total a pagar a cada tipo de atleta (profissional, semiprofissional e amador), percorrendo apenas uma vez o contentor. Deve ser também calculado e apresentado o valor total a pagar a todos os atletas.
- Executar as funcionalidades implementadas na classe ClubeDesportivo e visualizar o seu resultado.

Realize o levantamento e as especificação dos requisitos funcionais e não funcionais, bem como a análise OO, que compreende a produção de:

- Glossário;
- Modelo de casos de uso
  - Diagrama de casos de uso;
  - Casos de Uso (formato completo);
  - Diagrama de Sequência do Sistema (SSD's);
- Especificação suplementar (FURPS+) ;
- Análise OO
  - Modelo de domínio (MD).

Todo o código produzido deve ter sempre em consideração os principais princípios da programação orientada por objetos: abstração, encapsulamento, herança e polimorfismo.

O núcleo principal do software deve ser implementado em Java. Com o intuito de aumentar a manutenibilidade do software, devem ser adotadas boas práticas de análise e design de software OO.

Aplique o processo de desenvolvimento de software designado por **Test Driven Development** (**TDD**) na implementação das classes.

Dever-se-á utilizar o *plugin* Maven JaCoCo (Java Code Coverage) no IDE Netbeans para verificar a cobertura de testes.

O código deverá conter os comentários necessários para que possa ser gerada a documentação usando a ferramenta Javadoc.

A implementação do software deve adotar normas de codificação (e.g. Camel case) e de controlo de versões. O controlo de versões será conseguido, usando o GitHub.

O trabalho deverá ser realizado por **um grupo de dois formandos**.

Deverá ser submetido no Moodle do UPskill, um ficheiro ZIP com: o projeto Maven com o seguinte formato: **JavaNºTurma_PrimeiroUltimoNome_PrimeiroUltimoNome**, como por exemplo, Java1_IsabelBras_ArturSilva; e um pdf com o diagrama UML e a especificação dos requisitos funcionais e não funcionais e a análise OO.

O projeto Maven deve ser implementado recorrendo a um repositório do GitHub, criado e configurado por um dos elementos do grupo. Na turma Java1, os professores Alexandre Gouveia (aas@upskill.pt) e Jorge Santos (ajs@upskill.pt) têm de ser adicionados à lista de elementos com acesso ao repositório. Na turma Java2, os professores Jorge Duarte (fjd@upskill.pt) e Nuno Melo e Castro (anc@upskill.pt) têm de ser adicionados à lista de elementos com acesso ao repositório.

O trabalho deverá ser submetido no Moodle até às 17:30 do dia 26 de novembro (quinta-feira).