

Cotação:

2.5

2

2.5

2

Universidade de Trás-os-Montes e Alto Douro Engenharia Informática e Tecnologias de Informação e Comunicação **Programação Orientada a Objetos**

Segunda Frequência	31.5.2019
N° Nome: Duração: 100 min. + 15min.	Curso:
Parte Pr	rática
Ouestão: 1 2 3 4 5	6

2

1.5

Num campeonato de corridas vão ser realizadas diversas provas por 3n **Pilotos** pertencentes a **n** marcas. Em todas as provas 3 pilotos conduzem para a mesma **Marca** e recebem como pontuação o tempo de duração da prova em minutos inteiros. Pretende-se guardar a informação necessária para saber quais os **Pilotos** e qual a **Marca** que ganham o campeonato; ganha quem tiver menor pontuação.

Os <u>Pilotos</u> são diferenciados em três tipos: <u>Senior</u>, <u>Junior</u>, e <u>Rookie</u> e cada um dos tipos tem o seu próprio campeonato de Pilotos. Existirá um campeão em cada uma das categorias. O que distingue as categorias é o facto de ter peso extra ou não, e o facto de poder usar DRS ou não.

- Senior Peso extra de 20 kg e pode usar DRS.
- Junior Peso extra de 10 kg e não pode usar DRS.
- Rookie Sem peso extra nem pode usar DRS.

Nota1: O **Piloto** é uma representação abstrata dos pilotos, onde se regista o <u>nome do piloto</u>, <u>número</u> <u>de inscrição</u> e <u>pontos</u>, e possui dois métodos <u>PesoStatus</u> e <u>DRSStatus</u> virtuais puros.

1. Defina as classes <u>Piloto</u>, <u>Senior</u>, <u>Junior</u>, e <u>Rookie</u>, estabelecendo as relações que ache necessárias para permitir implementar um modelo de dados que dê suporte ao problema apresentado acima. Em cada classe implemente os seus construtores, destrutor e um método (**ReadFile**) para ler objetos da classe a partir de ficheiro (exemplo do ficheiro na página 4). **Nota:** Pode assumir a implementação dos métodos **Get** e **Set**.

Programação Orientada a Objetos	
	_
	_
	_
	_
	_
	_
	_

=	Programação Orientada a Objetos
C	Com a implementação da tipologia de pilotos, é necessário definir a classe Marca . Cada Marca tem
ļ	ma <u>designação</u> e tem 3 <u>Pilotos</u> associados. Tenha em consideração que cada Marca tem de competir
)	brigatoriamente com um piloto Junior e um Rookie , sendo o terceiro à escolha.
	2. Defina e implemente a classe Marca implementando os seus construtores, destrutor e um
	método para ler objetos da classe a partir de ficheiro (ReadFile).

	Programação Orientada a Objetos
	No ficheiro "marcas.txt" são guardadas todas as informações das Marcas que participam no campeonato.
3.	Desenvolva um programa para ler a partir do ficheiro " <i>marcas.txt</i> " toda a informação sobre as marca em competição e guardá-la numa estrutura de dados adequada. O ficheiro possui um registo (Marca por linha no seguinte formato: Designação; Piloto; Piloto; Piloto. Os dados de cada Piloto sã precedidos de um caracter designativo da sua categoria (S, J ou R).
	ZZTone;S:Neca Tone;123;200;20;sim;J:Zeca Tone;456;176;10;não;R:Nelia Tone;789;678;não BalaZZzz;J:Rato;321;300;10;não;R:Culatra;654;276;não;J:Bino;987;478;10;não
	EOF
_	Nota: Crie os métodos que achar necessários em cada classe

Prog	Programação Orientada a Objetos		
4.	Implemente o código necessário para calcular e apresentar no ecrã o número total de kg		
	extra a ser usados na competição, por cada marca.		

Pro	gramação Orientada a Objetos
5.	Implemente o código necessário para proceder à substituição de um piloto de uma determinada
	marca (a marca e o nome dos pilotos devem ser lidos através do teclado).

Prog	ramação Orientada a Objetos
6.	Implemente o código necessário para mostrar no ecrã o nome do Piloto vencedor de cada uma
	das categorias.

Programação Orientada a Objetos	

Programação Orientada a Objetos	