

ESQUEMA AULA PRÁTICA 5

1 - A Federação Portuguesa de Futebol encomendou um programa para gerir o número de golos marcados pelos jogadores de cada equipa da 1ª divisão. Para isso pretende-se desenvolver a classe Jogador e a classe Equipa (esta última será desenvolvida numa folha prática a estudar mais à frente. Em cada uma destas classes deve definir os modificadores de acesso de forma a que apenas os métodos da classe acedam aos atributos da classe e que qualquer classe que tenha acesso ao package da classe possa aceder aos seus métodos.

- Cada instância da classe Jogador terá como atributos o nome do jogador (cadeira de caracteres) o seu salário (um valor do tipo double) e um array com 34 valores inteiros onde serão armazenados o número de golos que o jogador marcou. (Na posição 0, número de golos marcados na 1ª jornada, na posição 1, o número de golos marcados na 2ª jornada e assim por diante.

A classe deverá ter dois construtores, um com o parâmetro nome e outro com os parâmetros nome e salário.

Cada instância da classe Jogador deverá poder responder a mensagens para:

- consultar o nome do jogador;
- modificar o nome do jogador;
- consultar o salário do jogador;
- modificar o salário do jogador;
- consultar o array de golos do jogador;
- modificar o array de golos do jogador;
- aumentar o salário de uma dada percentagem (double, entre 0 e 100);
- modificar o número de golos na n-ésima jornada;
- consultar o número de golos na n-ésima jornada;

- a) Construa a classe Jogador de forma a responder às especificações anteriores.
- b) Construa para a classe Jogador o método toString
- c) Construa uma classe de teste que permita verificar os dois construtores, e os métodos de instância definidos.

2 – Construa uma classe de Teste para estudar a classe **String** predefinida na linguagem Java.

Teste cada um dos métodos seguintes:

```
char charAt(int);  
int length();  
int indexOf(String);
```

String concat(String);
String substring(int);
int compareTo (String);

- Construa um programa onde deve definir uma String com um dos textos abaixo, à sua escolha:

“The competent programmer is fully aware of the limited size of his own skull. He therefore approaches his task with full humility, and avoids clever tricks like the plague.”

[Edsger Dijkstra, 1972](#)

“A programmer – someone who creates algorithms and codes them up – is a minor god, creating universes at will. You could even say that the God of Genesis himself is a programmer: language, not manipulation, is his tool of creation. Words become worlds. Today, sitting on the couch with your laptop, you too can be a god. Imagine a universe and make it real.”

[Pedro Domingos, 2016](#)

- O programa deverá:

- a) Contar quantos caracteres tem a String (incluindo espaços);
- b) Contar quantos caracteres (sem contar os espaços) tem a String;
- c) Contar quantas palavras tem a String;
- d) Contar o número de vezes que ocorreu cada uma das letras do alfabeto na String.
- e) Dada uma palavra verificar se ele pertence ao texto e em que posição está.

3 – Construa uma classe Fila que defina uma estrutura de dados do tipo fila com o comportamento tradicional em que o primeiro elemento a entrar é o primeiro elemento a sair (First In First Out). A implementação da fila deverá ser feita através de um array de inteiros, com utilização circular e cuja dimensão será indicada pelo utilizador.

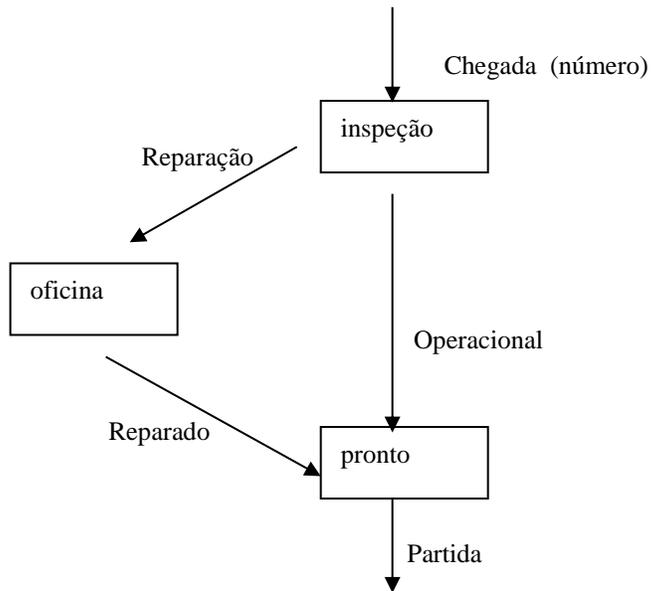
(Veja a implementação da aula teórica, TP04 – pág. 28)

a) Implemente as operações:

- criar uma fila com dimensão **n**;
- inserir um elemento na fila;
- retirar um elemento da fila;
- verificar se a fila está vazia;
- verificar se a fila está cheia;
- implemente o método toString de forma a que apenas mostre os valores que fazem parte da fila.

b) Construa uma classe de Teste para verificar a correcção da classe anterior.

c) Usando a classe Fila pretende-se simular um sistema de recolha e reparação de autocarros. Cada autocarro é identificado por um número. À chegada de um autocarro este é colocado numa fila de **inspeção**. Após esta, ou está operacional, seguindo para a fila **pronto**, ou segue para a oficina para reparação, sendo colocado na fila **oficina**:



Construa um programa que mostre ao utilizador quais as operações disponíveis e que, de acordo com a escolha do utilizador, movimente o autocarro nas filas correspondentes.

Opção 1 – Chegada (adicionar um autocarro à fila de **inspeção**, o autocarro é identificado por um numero inteiro)

Opção 2 - Reparação (um autocarro sai da frente da fila de **inspeção** e entra nas costas da fila de **oficina**)

Opção 3 - Reparado (um autocarro sai da frente da fila de **oficina** e entra nas costas da fila **pronto**)

Opção 4 – Operacional (um autocarro sai da frente da fila de **inspeção** e entra nas costas da fila **pronto**)

Opção 5 – Partida (sai o autocarro da frente da fila **pronto**)

- Após cada opção do utilizador, e respectiva movimentação, o programa deverá mostrar o conteúdo de cada uma das filas.