

ESQUEMA AULA PRÁTICA 6

Composição de classes versus Herança

□ **Composição de classes**

1 - Considere a classe Jogador que construiu na folha prática 4, a partir da qual quer agora construir uma equipa de futebol.

- Uma equipa tem como atributos o nome da equipa e uma lista de Jogadores. Considere para isso um objeto do tipo ArrayList<Jogador>” isto é uma lista dinâmica onde são armazenados os jogadores da equipa.

A classe Equipa terá um construtor com o nome da equipa como parâmetro e cada instância deverá poder responder às seguintes mensagens:

- inserir um novo jogador na equipa;
- remover um jogador da equipa dada a sua posição na lista de jogadores;
- dar a conhecer o número de jogadores da equipa
- dado o nome de um jogador, verificar este pertence ou não à equipa;
- devolver o nome do jogador que marcou mais golos no campeonato.
- mostrar sob a forma de texto o estado de um objecto do tipo equipa (método toString)
- comparar dois objectos (método boolean equals (Object o)).
- criar um cópia do objecto (método Object clone()).

2 - Construa uma classe de Teste em que verifique a correcção dos métodos da classe Equipa.

□ **Herança**

3 – Considere o esquema da classe abaixo:

a) Implemente a classe construindo cada um dos seus métodos.

```
public class Base {
    private int a;
    private int b;

    public Base(int a, int b) {
```

```
    this.a=a;   this.b=b;
}
public int soma(){
    return a+b;
}
public int soma(int x){
    return a+b+x;
}
public int getA(){...}
public int getB(){...}
public void setA(int a){...}
public void setB(int b){...}
public String toString {...}
public boolean equals (Object o ){ ...}
public Object clone () {...}
}
```

b) Construa uma classe teste para testar a classe Base.

4 - Defina uma classe, Derivada, como subclasse da classe Base. Na subclasse Derivada, defina um atributo **a do tipo int**, um atributo **c do tipo int**, e um construtor com a assinatura **Derivada (Base b, int a, int c)**.

- Na subclasse Derivada construa ainda os seguintes métodos:

a) Getters e setters;

b) Método toString.

c) Método que calcule a soma de todas as variáveis de instância de um objeto da classe Derivada.

d) Método equals.

e) Método clone.

5 - Construa uma classe teste para testar a classe Derivada.

6 - Modifique a classe Base declarando os atributos **a** e **b** com o modificador de acesso **protected**.

- Explore o que pode agora mudar no exercício 4?

Classe java.util.ArrayList:

```
ArrayList() // construtor vazio, dimensão inicial zero.  
boolean add(Object element)  
// adiciona o elemento especificado ao final da lista  
void add( int index, Object obj)  
//insere o elemento especificado na posição index  
Object remove(int index )//remove o elemento da posiçã index  
boolean remove( Object o)  
//remove a primeira ocorrência do objecto dado como parâmetro  
Object set (int position, Object obj )  
// substitui o elemento da posição index pelo elemento dado  
Object get (int position)//devolve o elemento da posição index  
void clear() // remove todos os elementos da lista  
Object clone() // devolve uma cópia da lista  
boolean contains(Object element)  
// devolve true se a lista contém o elemento especificado  
boolean equals ( Object obj)  
// permite comparar duas listas  
int indexOf(Object element)  
// procura o índice da 1ª ocorrência de elemento  
boolean isEmpty() // verifica se a lista não tem componentes  
int size() // devolve a dimensão actual  
String toString ()
```