- ## Composição de classes

1 - Considere a classe Jogador que construiu na folha prática 5, 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()).

a) Construa a classe Equipa.

b) Teste a classe Equipa.

2 – Considere o exercício da aula teórica 3:

Construa uma classe que represente os Empregados de uma empresa. Um Empregado tem um número de segurança social, um nome e um salário. Defina os atributos, dois construtores à sua escolha, os métodos de consulta (getters) e os de modificação (setters), e o método toString. Construa ainda um método que permita subir o salário do empregado de uma dada percentagem dada como parâmetro.

a) Construa a classe Empregado, construindo também os métodos equals e clone.

b) Construa um programa que permita "gerir" os empregados de uma empresa. Para isso deverá armazenar os empregado num objecto do tipo ArrayList<Empregado>.

Deverá ter as opções:

1 – Criar empregado;
2 – Consultar todos os empregados;
3 – Modificar um empregado;
4 – Apagar um empregado.

- ## Escrever e ler objectos em ficheiros

3 – Teste os blocos de código listados abaixo:

```
import java.io.*;
import java.util.*;
…
      ArrayList<String> lista = new ArrayList<String>();
      lista.add("um");
      lista.add("dois");
      lista.add("três");
      lista.add("quatro");
      // Escrever um objeto para um ficheiro
      try {
         ObjectOutputStream os = new ObjectOutputStream(new
                                 FileOutputStream("d:\\teste\\ficheiroExemplo.dat"));
         // escrever o objeto lista no ficheiro
         os.writeObject(lista);
         os.flush();
      } catch (IOException e) {
         System.out.println(e.getMessage());
      }

      // Ler um objeto de um ficheiro
      ArrayList<String> lista2 = new ArrayList<String>();
      try {
         ObjectInputStream is = new ObjectInputStream(  new
                                   FileInputStream("d:\\teste\\ficheiroExemplo.dat"));

         lista2 = (ArrayList) is.readObject();

         System.out.println(lista2);
         }
         catch (IOException e){
               System.out.println(e.getMessage());
         }
         catch ( ClassNotFoundException e) {
               System.out.println(e.getMessage());
         }
```

4 - Modifique o exercício 5 de forma a escrever a lista de empregados num ficheiro. Assim quando termina a execução do programa os seus dados ficarão armazenados em disco. Ao voltar a executar o programa deverá ler os dados. Para poder escrever objectos num ficheiro a classe desses objectos terá de implementar a interface (!) **Serializable.**

O cabeçalho da classe Empregado deverá ser:

```
public class Empregado implements Serializable {
```

Para treinar:

5 – Reescreva a classe Fila que construiu na folha prática 5, exercício 3, usando agora uma ArrayList de objectos do tipo Integer em vez do array de inteiros que usou na implementação anterior.

6 – Teste a nova classe Fila.

7 – Verifique que alterações tem de fazer na classe que construiu para gerir a sua frota de autocarros se usar a nova implementação da classe fila.

8 – Implemente a nova solução com persistência de dados.