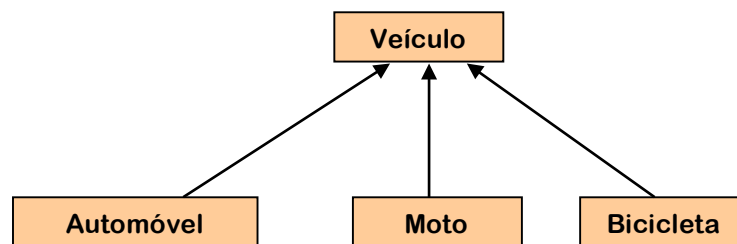


Aula prática nº 11**Tópicos**

- Problemas de POO
- Acesso a ficheiros
- Exceções

Exercícios

1. Considere as seguintes entidades:



- a) Construa classes para representar estas entidades tirando partido do conceito de generalização. Tenha em consideração as características e comportamento dos objetos apresentados como, por exemplo, todos os veículos têm uma matrícula, um ano e uma cor base; todos os automóveis e motos têm potência. Considere que dois veículos são iguais se possuírem a mesma matrícula.
- b) Para além dos elementos apresentados pretende-se que alguns dos veículos acima possam ter um motor elétrico (Automóvel, Moto), motor de explosão (Automóvel, Moto), ou ambos (Automóvel). As viaturas com motor eléctrico devem apresentar o método `double getAutonomiaO`; enquanto que as de motor de combustão devem incluir `double getEmissaoCO2O`; . Implemente construções adequadas a este problema. Note que para obter uma solução poderá ter de modificar a hierarquia de classes, acrescentando novas classes e/ou interfaces.
- c) Desenvolva um programa para testar todas as funcionalidades implementadas nas alíneas anteriores. No seu programa, considere a existência de ficheiros com informação sobre vários veículos com o formato descrito de seguida. O programa deverá ler o conteúdo de um ficheiro especificado pelo utilizador e, como resultado da execução, gravar um novo ficheiro contendo a informação sobre as viaturas ordenadas por matrícula (crescente).

Deverá também ser impresso no terminal a informação sobre todos os veículos mas agrupados por tipo (ex. todas as bicicletas, todas as motos elétricas, todos os híbridos, etc.).

- d) Inclua mecanismos de controlo de exceções em todos os pontos do programa em que possam surgir situações imprevistas (ex. ficheiros).

Considere como exemplo o ficheiro “viaturas.csv”:

```
CarroHibrido;00-AA-00;Verde;2000;150;400;50
CarroEletrico;ZZ-00-00;Azul;2010;75;300
CarroCombustao;00-ZZ-00;Vermelho;2017;120;100
Bicicleta;55-55-CC;Branco;2015
MotoEletrica;11-11-BB;Preto;2010;500;50
MotoCombustao;99-99-ZZ;Amarelo;2018;750;75
```

O ficheiro resultante após o processamento do ficheiro “viaturas.csv” seria o seguinte:

```
Carro [potencia=150.0, Veiculo [cor=Verde, ano=2000,
matricula=00-AA-00]], autonomia=400.0, co2=50.0
Carro [potencia=120.0, Veiculo [cor=Vermelho, ano=2017,
matricula=00-ZZ-00]], co2=100.0
MotoEletrica [autonomia=50.0, Moto [cilindrada=500.0, Veiculo
[cor=Preto, ano=2010, matricula=11-11-BB]]]
Bicicleta [Veiculo [cor=Branco, ano=2015, matricula=55-55-CC]]
MotoCombustao [co2=75.0, Moto [cilindrada=750.0, Veiculo
[cor=Amarelo, ano=2018, matricula=99-99-ZZ]]]
Carro [potencia=75.0, Veiculo [cor=Azul, ano=2010,
matricula=ZZ-00-00]], autonomia=300.0
```

2. Considere as seguintes entidades:

- País: caracterizado por um nome (String), uma capital (Localidade) e por um conjunto de regiões (Região).
- Região: caracterizada por um nome (String) e uma população (int)
- Estado: caracterizado por um nome (String), uma população (int) e uma capital (Localidade do tipo TipoLocalidade.CIDADE)
- Província: caracterizada por um nome (String), uma população (int) e um governador (String)
- Localidade: caracterizada por um nome (String), uma população (int) e um tipo (TipoLocalidade.CIDADE, TipoLocalidade.VILA, TipoLocalidade.ALDEIA)

- a) Construa um programa que represente estas entidades. Crie construtores, os métodos *set/get/hasCode>equals (...)* que lhe pareçam adequados e outros que sejam fundamentais para uma boa modulação e para o bom funcionamento do programa.
- b) Teste as classes desenvolvidas com a função *main* seguinte.

```
public static void main(String[] args) {

    Localidade cid1 = new Localidade("Szohod", 31212,
        TipoLocalidade.Cidade);
    Localidade cid2 = new Localidade("Wadesdah", 23423,
        TipoLocalidade.Cidade);
    Localidade cid3 = new Localidade("BedRock", 23423,
        TipoLocalidade.Vila);
    Estado est1 = new Estado("North Borduria", 223133, cid1);
    Estado est2 = new Estado("South Borduria", 84321, cid2);

    Pais p1 = new Pais("Borduria", est1.getCapital());
    Pais p2 = new Pais("Khemed", cid2);
    Pais p3 = new Pais("Aurelia");
    Pais p4 = new Pais("Atlantis");
    p1.addRegiao(est1);
    p1.addRegiao(est2);
    p2.addRegiao(new Provincia("Afrinia", 232475, "Aluko Pono"));
    p2.addRegiao(new Provincia("Eriador", 100000, "Dumpgase Liru"));
    p2.addRegiao(new Provincia("Laurania", 30000, "Mukabamba Dabba"));

    List<Pais> org = new ArrayList<Pais>();
    org.add(p1);
    org.add(p2);
    org.add(p3);
    org.add(p4);

    System.out.println("----Iterar sobre o conjunto");
    Iterator<Pais> itr = org.iterator();
    while (itr.hasNext())
        System.out.println(itr.next());

    System.out.println("-----Iterar sobre o conjunto - For each (java 8)");
    for (Pais pais: org)
        System.out.println(pais);
    // ToDo:
    // adicionar, remover, ordenar, garantir elementos únicos
}
```

Resultado esperado:

```
----Iterar sobre o conjunto
Pais: Borduria, População: 307454 (Capital: Cidade Szohod, população 31212)
Pais: Khemed, População: 362475 (Capital: Cidade Wadesdah, população 23423)
Pais: Aurelia, População: 0 (Capital: *Indefinida*)
Pais: Atlantis, População: 0 (Capital: *Indefinida*)
-----Iterar sobre o conjunto - For each (java 8)
Pais: Borduria, População: 307454 (Capital: Cidade Szohod, população 31212)
Pais: Khemed, População: 362475 (Capital: Cidade Wadesdah, população 23423)
Pais: Aurelia, População: 0 (Capital: *Indefinida*)
Pais: Atlantis, População: 0 (Capital: *Indefinida*)
```