

---

---

**PARADIGMAS DE PROGRAMAÇÃO IV – LESI**  
e  
**PROGRAMAÇÃO ORIENTADA AOS OBJECTOS - LCC**  
**2º ANO/2º SEMESTRE – 2006/2007**

Exame de Recurso – 23 de Julho de 2007  
Cotação - 20 valores      Duração - 2h00m

---

---

Considere a classe **PontoGps**, que representa as coordenadas GPS de um dado ponto, designadamente, latitude, longitude e altura, classe parcialmente definida como:

```
public class PontoGps {
    private double lat, long, alt;
    .....
    // considere que construtores, getX(), setX(..), toString() e clone() estão disponíveis
}
```

Considere ainda a classe **InfoGps** que representa o conjunto de informação que se pretende associar a um dado ponto GPS, designadamente, o continente, país e cidade, e ainda uma tabela que associa a uma dada *string* identificativa de um tipo de serviço (cf. farmácia, hospital, restaurante, hotel, etc.) um conjunto contendo as localizações (*nome e endereço* do tipo *String* cf. classe **LocalServ**) de serviços desse tipo. A variável de classe **serviços** contém todos os nomes de serviços considerados válidos. As instâncias de **InfoGps** podem não conter informações sobre todos os serviços definidos mas apenas sobre os que de facto existem (por exemplo, não existem hospitais ou piscinas).

```
public class InfoGps {
    // de classe
    public static HashSet<String> servicos = new HashSet<String>();
    public static void addServ(String serv) { servicos.add(serv); }
    // de instância
    private String cont, pais, cidade;
    private TreeMap<String, HashSet<LocalServ>> infos;
    .....
    // considere que construtores e métodos getX() "deep" e setX(..) estão disponíveis
}
```

Chegamos assim à classe principal que se pretende implementar, a classe **MapaGps**. A classe **MapaGps** é representada por um **TreeMap** que associa um *ponto Gps* a uma *ficha de informação sobre esse ponto*, cf.:

```
public class MapaGps {
    private TreeMap<PontoGps, InfoGps> mapa;

    // considere que construtores e métodos getX() "deep" e setX(..) estão disponíveis
}
```

A classe **MapaGps** deverá ser agora completada com um conjunto de métodos que permitam realizar as seguintes operações:

- 1) Método que determina o número total de pontos referenciados entre 2 latitudes dadas;
- 2) Método **getMapa()** que devolve uma “deep copy” da variável de instância *mapa*;
- 3) Método que determine quantos continentes estão representados pelos pontos GPS;

- 4) Método que devolva a lista de tipos de serviços existentes num dado ponto GPS (cf. hotel, etc.);
- 5) Método que dadas duas latitudes dê como resultado todas as cidades que podem ser encontradas;
- 6) Método que determine se existe algum ponto GPS de altitude superior a um valor dado;
- 7) Método que determine todas as cidades referenciadas acima de uma altitude dada;
- 8) Método que determine a lista de pontos GPS que dizem respeito a um dado país;
- 9) Método que insira um novo serviço e lista de localizações associado ao ponto GPS dado;
- 10) Método que devolva todos os pontos GPS definidos entre duas longitudes;
- 11) Método que determine um TreeMap contendo todos os serviços e respectivas informações que podem ser encontrados num país de nome dado;
- 12) Método que dado um ponto GPS determine o hospital mais próximo (país e cidade);
- 13) Método que crie um HashMap no qual a cada continente é associada a lista de todos os pontos GPS que se lhe referem;
- 14) Método que crie um Map no qual a cada informação de hospital (LocalServ) se associe o seu ponto GPS;
- 15) Existem vários tipos de *coordenadas GPS*, designadamente militares e civis, e ainda diferenciais. As coordenadas GPS militares e civis têm a si associado adicionalmente um *tempo de precisão*. As diferenciais são relativas a um ponto GPS fixo e pré-definido, sendo apenas guardadas por ponto as suas diferenças para este ponto referencial. Tomando em consideração a classe PontoGPS apresentada, defina as classes PontoGPSMilitar e PontoGPSDiferencial, apresentando a sua estrutura e os construtores completo e de cópia;

**Prof. F. Mário Martins**