PROGRAMAÇÃO ORIENTADA AOS OBJECTOS

TRABALHO PRÁTICO DE 2007/2008

2º ANO: LEI E LCC

ENTREGA: JUNHO DE 2008

AEROGEST – SISTEMA DE GESTÃO DE VOOS

O problema centra-se na implementação do **AEROGEST** – Software de Controlo e Gestão de Voos de um Aeroporto. Pretende-se criar um sistema software para controlo e gestão de parte da logística associada a um aeroporto, em especial a gestão diária dos **voos** que partem de um dado aeroporto.

O sistema AEROGEST recebe a qualquer momento solicitações de criação de voos, ou seja, especificações completas sobre um voo a criar para um dado dia, dada hora e dado destino, etc.

Não é esta, no entanto, a parte do problema que iremos tratar neste trabalho. Aqui vamos admitir que quando o dia D começa, o AEROGEST activa, ou seja põe em execução, um **Mapa de Voos** contendo informações sobre todos os voos a controlar e colocar no ar nesse dia. O AEROGEST vai registar e controlar todas as etapas necessárias à efectivação de cada um dos voos, podendo mesmo, em dadas circunstâncias, atrasá-los ou cancelá-los.

A entidade **voo** é uma entidade fundamental na resolução deste problema pelo que se desenvolveu um diagrama de estados e operações relacionados com a criação de um voo, que se apresenta na Fig.1, e que se aconselha seja consultado ao longo da apresentação dos requisitos do sistema a implementar.

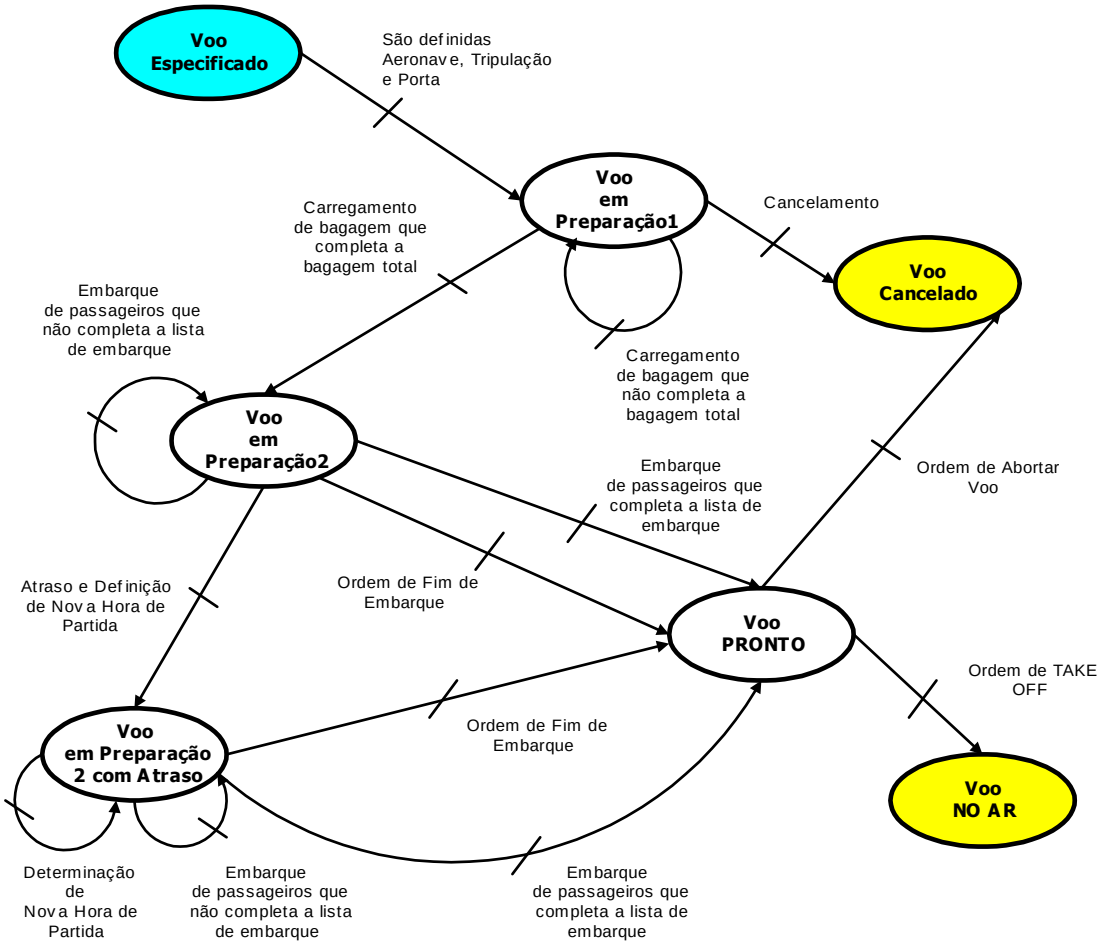

Fig. 1 – Diagrama de Estados de um VOO

Um **voo** é de facto, inicialmente, uma entidade semi-abstracta, uma especificação, que apenas terá existência concreta quando se garantirem todos os requisitos e procedimentos relacionados com entidades muito concretas (de informação ou físicas) que lhe devem ser correctamente associadas, qualquer que seja o seu tipo. Um voo poderá ser classificado como sendo do tipo: **comercial, militar, governamental ou privado, ou outros**.

No **Mapa de Voos** do dia do AEROGEST, cada voo é identificado por um código de voo e, independentemente do seu tipo, associa-se a uma companhia aérea, um conjunto de passageiros afectos a tal voo, uma eventual lista de espera de passageiros substitutos (que pode portanto existir ou não), uma carga (definida numa **lista de carga de produtos** a

embarcar), um destino, e um tempo de partida (hora/minuto). Uma **aeronave** específica capaz de realizar tal voo e uma **tripulação**, ser-lhe-ão posteriormente associadas também.

Conforme a Fig.1 indica, um voo encontra-se inicialmente no estado de **Especificado**, ou seja, tem código de voo, lista de passageiros e de carga, hora de partida e destino mas não lhe foi ainda atribuída a aeronave, a tripulação e a porta, e não começou ainda o embarque nem da carga nem de passageiros. Assim, o voo passa para o estado de **Em Preparação 1** quando se realiza a operação de atribuição de uma aeronave, de tripulação e de porta.

Uma **Aeronave** é uma entidade genérica capaz de voar, que poderá representar um helicóptero, um avião de passageiros, um avião de carga, um avião de combate, etc. Porém, qualquer aeronave deverá possuir os seguintes atributos: matrícula (12 caracteres), designação, capacidade máxima de passageiros, capacidade máxima de carga (em toneladas) e velocidade máxima.

Uma **Tripulação** é apenas uma estrutura de dados contendo **Fichas de Tripulação** onde se identifica a função, o nome e a nacionalidade de cada um dos tripulantes de um dado voo (exº Comandante, Luís Silva, Portugal, Co-piloto, Pedro Luís, Brasil, Assistente, Pedro Garcia, Espanhol, etc.). As funções devem ser validadas a partir de uma lista de funções válidas previamente criada, sendo certo que nenhum voo pode ser criado sem um Comandante e um Co-Piloto.

Quando às 00H00 de um novo dia o **AEROGEST** arranca, assume-se (de forma irreal por questões de simplificação do problema) que tudo o que havia a fazer no dia anterior foi resolvido. Assim, o AEROGEST activa um completamente novo **Mapa de Voos** a controlar e gerir nesse dia. Neste **Mapa de Voos** inicial todos os voos estão em estado de **Especificado**. As várias operações a realizar poderão levar o voo ao estado **NO AR** ou **CANCELADO**.

A passagem do estado de **Em Preparação 1** para **Em Preparação 2** e de Em Preparação 2 (com ou sem atraso) para o estado de **PRONTO** dependerá do carregamento da carga e do embarque de passageiros estar completo ou não, tendo em atenção em cada momento o que já foi carregado ou embarcado e as listas de carga e de passageiros iniciais. Estes carregamentos de carga e embarque de passageiros podem portanto ser feitos por partes, associando-se a cada um uma sublista de carga ou de passageiros associada à operação.

Uma **Lista de Passageiros** para um dado voo é apenas uma lista contendo os códigos, nomes e nacionalidades dos passageiros. Uma **Lista de Embarque** é exactamente a mesma estrutura, ainda que possivelmente parcial, mas que deve ser validada relativamente à lista de passageiros especificada para tal voo. Uma **Lista de Espera** para um dado voo tem a mesma estrutura da lista normal de passageiros.

Uma **Lista de Carga** consiste numa lista contendo os códigos e as características de todas as cargas a embarcar no avião. Qualquer **Carga** possui um código, um peso, um volume e uma descrição. As cargas são de momento divididas em categorias bem identificadas, designadamente: normal (cf. malas), animal, militar, química, alimentar, veículo e outra. Estas espécies de carga devem ser muito bem caracterizadas pois podem ser importantes para fins estatísticos. A solução deverá permitir de futuro criar outros tipos de carga. Cada tipo de carga tem também a si associado um tempo médio de carregamento que é proporcional ao seu peso e volume (deve ser criada uma fórmula razoável para determinar este tempo) ou considere que se trata apenas de um atributo da carga pré-definido. A **Lista de Carregamento** é, para cargas, o equivalente à lista de embarque para passageiros.

Em qualquer momento, o AEROGEST pode fazer passar um voo do estado de **Em Preparação 1** para **CANCELADO** através de uma operação para tal efeito. Esta operação registará ainda no Mapa de Voos, em relação ao respectivo voo, uma descrição textual indicando a razão do cancelamento.

Um voo no estado de **Em Preparação 2** pode passar para **PRONTO** se, mesmo que a lista de embarque não esteja completa (mesmo tendo já considerando os passageiros em lista de espera) for dada ordem de **FIM DE EMBARQUE**.

Um voo no estado de **Em Preparação 2** pode passar para **ATRASADO** se o AEROGEST determinar (através de um algoritmo que não é aqui solicitado) não ser possível satisfazer a hora de partida especificada, e activando a operação ATRASO E DEFINIÇÃO DE NOVA HORA.

Um voo no estado de **PRONTO** pode passar para **CANCELADO** se o AEROGEST activar a operação **ABORTAR VOO** que registará ainda, em associação com o voo no Mapa de Voos, uma descrição textual indicando a razão.

O **Mapa de Voos** tem, naturalmente, registado para cada voo o seu estado actual. No entanto, apenas os voos de tipo COMERCIAL terão a sua informação projectada no **PLACARD** do aeroporto, informação essa que terá a seguinte aparência e conteúdo:

AEROPORTO DE GUALTAR – HORA ACTUAL: 14H30

Voo	Destino	Partida	Porta	Estado	Obs
TP198	Lisboa	9.00	A19	Concluído	
PG333	P. Maiorca	10.00	A11	Cancelado	Condições Climatéricas
TP205	Madrid	11.35	A23	Concluído	
BA222	Londres	14.30	B21	Atrasado	Nova Hora: 15.30
KLM101	Frankfurt	14.50	B33	Embarque	
ALT398	Roma	15.00	A45	Em Preparação	

Note que o ESTADO de um voo apresentado no PLACARD é uma interpretação textual do estado real apresentado no diagrama da Fig.1. Assim, Em Preparação significa no estado 1 ou 2 da PREPARAÇÃO, ATRASADO, CONCLUÍDO e CANCELADO são óbvios a partir do diagrama de estados apresentado.
Todas as classes desenvolvidas neste projecto deverão poder ser guardadas numa ObjectStream e escritas como texto numa Writer, ou seja, todas deverão ser Serializable.

Para além destas funcionalidades de controlo e gestão de voos, que deverão fazer parte de um menu especial do programa principal, o AEROGEST deverá ainda implementar as seguintes funcionalidades (eventualmente agrupadas igualmente por sub-funcionalidades a apresentar noutros submenus) que correspondem a consultas e outras operações importantes para a gestão:

- Inicar a hora actual do sistema AEROGEST;
- Fazer a actualização o PLACARD do Aeroporto (apenas para voos COMERCIAIS);
- Imprimir o actual Mapa de Voos;
- Determinar o estado de um voo de código dado;
- Devolver uma lista de códigos de voos que se encontram num dado estado;

- Devolver o total de voos, de todos os tipos, já concluídos no dia;
- Determinar o tempo esperado para concluir a carga de um dado voo;
- Devolver, numa estrutura adequada, o total de voos do dia nos estados de: NO AR, CANCELADO, PRONTO e EM PREPARAÇÃO (com ou sem atraso);
- Guardar o estado do AEROGEST numa ObjectStream;
- Devolver uma lista com todos os códigos de voos de um dado tipo (cf. MILITAR, COMERCIAL, etc.) escalonados para este dia;
- Determinar dados estatísticos relevantes tais como: nº total de passageiros embarcados em voos de tipo comercial até essa hora, peso total da carga embarcada, etc.
- Determinar, para todos os voos de tipo COMERCIAL (!!) a diferença entre o número de passageiros previsto e o número de passageiros de facto embarcados;

EXECUÇÃO DO PROJECTO

- Desenvolva iterativamente e teste todas as classes usando o ambiente BlueJ;
- Não se esqueça de definir e implementar os métodos de alteração do estado interno dos objectos e os métodos de consulta desse estado interno.
- Não se esqueça de implementar os métodos toString(), equals() e clone() das classes que implementou, sempre que tal se justifique. Por exemplo, não se justifica comparar dois Mapas de Voo. Há no entanto várias operações importantes sobre Listas de Passageiros, de Embarque e de Carga.
- Crie em BlueJ a documentação das classes desenvolvidas (APIs) e o Diagrama de Classes;
- Crie um programa principal Java que realize o carregamento inicial de dados das diversas classes (com um mínimo de 10 voos comerciais e outros em diferentes situações) e que devolva como resultado um estado inicial aceitável para o AEROGEST ser testado na apresentação do trabalho. Pode também considerar a hipótese de leitura do estado inicial a partir de uma ObjectStream.

A estrutura do programa principal será a seguinte:

```
public class TEST_AEROGEST {
  public static Aerogest main() {
    Aerogest aeroGualtar = new Aerogest();
    // criação das mais diversas instâncias
    return aeroGualtar;
   }
}
```

APRESENTAÇÃO DO PROJECTO

♦ O projecto a entregar deverá ter um pequeno relatório, na capa do qual se identificam os elementos do grupo por nome e fotografia digitalizada, contendo a documentação gerada em BlueJ (APIs) e as decisões tomadas durante a execução do projecto (cf. simplificações, interpretações, etc.);

♦ Cada grupo terá cerca de 20 minutos para apresentar o seu trabalho.

♦ Situações de erro de execução ou de impossibilidade de execução serão de imediato canceladas, não havendo lugar a entregas posteriores. Idealmente os grupos deverão trazer os seus projectos em "flash disk" ou nos seus computadores portáteis.

Prof. F. Mário Martins