



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Algoritmos e Estruturas de Dados

Trabalho realizado por:

Nuno Afonso Anjos Pereira - up202007865
Pedro Miguel Magalhães Nunes - up202004714
Vitor Manuel da Silva Cavaleiro - up202004724

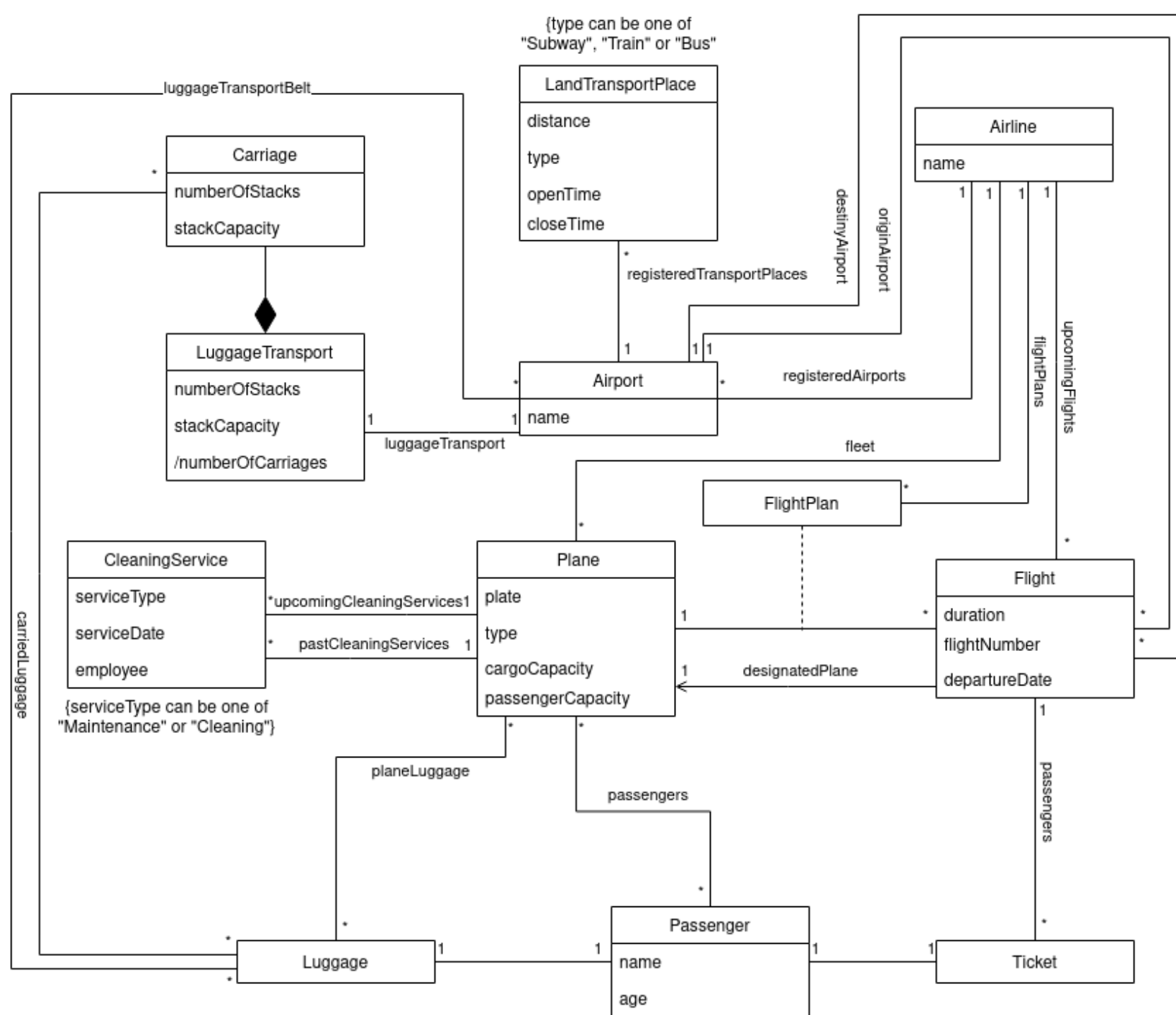
Descrição do Problema

Implementar um sistema de gestão de informação para uma companhia aérea, capaz de guardar e gerir informação relativa a aviões, voos, passageiros, bagagens e aeroportos.

...

Solução

- Implementámos um menu de forma a facilitar a interação com o programa;
- O núcleo central do programa é a classe Airline, que gere as interações entre aviões, aeroportos, voos e passageiros;
- Dentro das soluções aos problemas apresentados, que implementámos, eis dois exemplos:
- No tratamento dos serviços criamos duas filas(serviçosAcabados, serviçosMarcados), quando um novo serviço era criado, era colocado nos serviçosMarcados e depois de ser cumprido, era colocado nos serviçosAcabados.
- No tratamento das malas criamos um sistema de transporte, composto por carruagens que, à medida que estas ficam cheias, adicionam novas carruagens. Cada carruagem possui 4 stacks e cada stack tem a capacidade máxima de 5 malas.



Estrutura de Ficheiros

- Incluimos no nosso projeto uma estrutura de ficheiros dos seguintes elementos:
- Aviões(tipoAvião, matrícula, capacidade, capacidadeCargo);
- Aeroportos(nomeAeroporto, aviõesAterrados);
- Voos(número, data, duração, matrícula, origem, destino);
- Locais de transporte terrestre(Aeroporto, tipoTransporte, distância, horaAbertura, horaFecho);
- Serviços de limpezas(matrículaAvião, tipo, data, empregado).
- Nota:Os serviços de limpeza estão organizados por avião e por ordem cronológica.

Funcionalidades Implementadas

No que toca nas operações CRUD:

- Na classe Flight, implementámos todas as operações CRUD e a mesma possui listagem parcial;
- Na classe LandTransportPlace, implementámos as operações create, read e update, a mesma tem listagem parcial;
- Na classe CleaningService, implementámos as operações create e read, a mesma tem listagem parcial;
- Na classe Plane, implementámos as operações CRUD, a mesma tem listagem total;
- Na classe Airport, implementámos as operações create, read e update, a mesma tem listagem total;

Funcionalidades Implementadas(cont)

- Algoritmos de pesquisa usados:
- `std::find_if` (STL);
- Recorreremos também à pesquisa linear com recurso a iteradores.
- Algoritmos de ordenação usados:
- `std::sort` (STL);
- `std::remove_if` (STL);
- `Std::list.sort` (STL).

Destaque de Funcionalidade

- A funcionalidade que gostaríamos de realçar seria a que nos permite a realização de voos e também de serviços de limpeza/manutenção com base na data atual, o que dá uma sensação de "realismo" à aplicação em si.

Principais dificuldades

- Ao longo do trabalho, tivemos dificuldades em relacionar as classes devido à maneira como estas estavam especificadas no enunciado, levando a dependências circulares no início (Nomeadamente, Flights e Planes). A solução foi criar a classe FlightPlan.
- Tivemos também problemas a gerir os objetos das classes. A solução foi usar pointers.

Esforço de cada membro

Vitor Cavaleiro: 33%

Pedro Nunes: 33%

Nuno Pereira: 33%