

Trabalho Final

Transportes espaciais

1. Enunciado geral:

O ano é 2050. Além da propulsão de impulso subluz, a humanidade desenvolveu uma propulsão mais rápida que a velocidade da luz e viaja a diversos espaço-portos pela galáxia.

O sistema de localização espacial utiliza o espaço cartesiano 3D, em que cada ponto possui coordenadas X, Y e Z; sendo que a Terra é considerada como o ponto de origem (0,0,0) e a escala é de 1 : 1 ano-luz.

A velocidade é medida em Warp (ano-luz/hora).

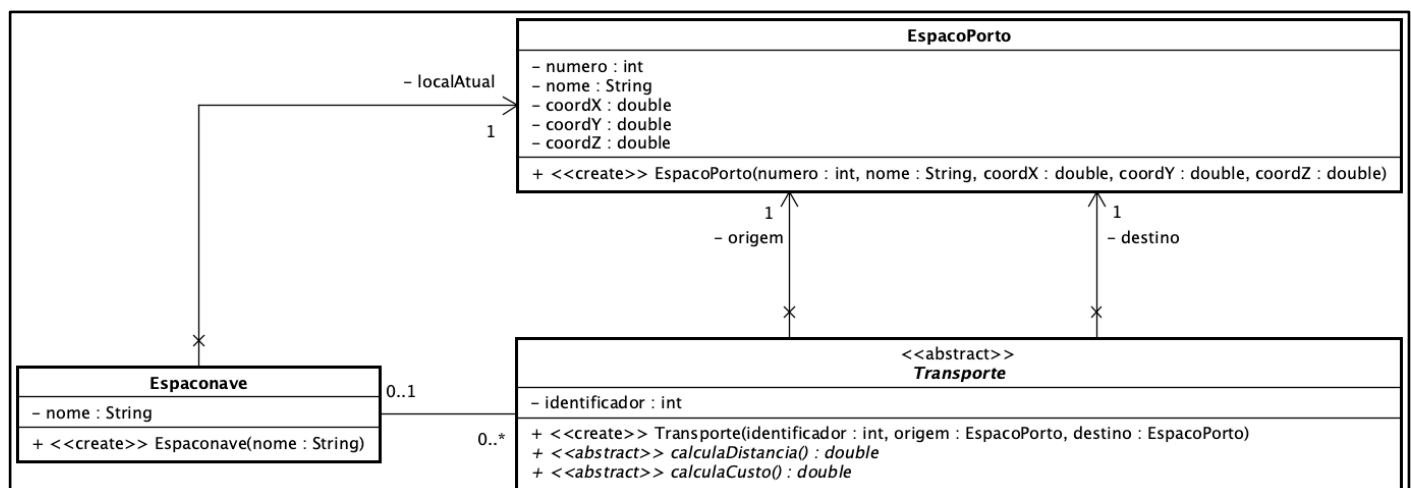
A unidade monetária existente é Créditos (C\$).

Incrivelmente a linguagem Java continua existindo, na versão 33.

A ACMESpace é uma corporação que gerencia espaçonaves para o transporte espacial de pessoas e cargas.

Esta empresa deseja que seja desenvolvido um sistema para gerenciar suas espaçonaves e transportes.

O analista de sistemas definiu as seguintes classes iniciais:



Na definição do sistema o analista também identificou as seguintes características:

- Todo espaço-porto possui um número único, um nome e suas coordenadas (X, Y e Z).
- Toda espaçonave possui um nome único e está em algum espaço-porto (o espaço-porto *default* é na Terra). Uma espaçonave só pode ser designada para um transporte de cada vez, mas em seu histórico pode ter realizado vários transportes.
- Uma espaçonave pode ser Subluz ou FTL.
 - Se for subluz, há a informação adicional de velocidade máxima de impulso (valor limite: 0,3 Warp) e combustível (nuclear ou íon).
 - Se for FTL, há as informações adicionais de velocidade máxima Warp e de limite de transporte (quantidade máxima de pessoas ou carga).
- Todo transporte possui um identificador único, o espaço-porto de origem e o espaço-porto de destino.
- Um transporte pode ser de pessoas ou de materiais.

- Se um transporte for de pessoas, há a informação adicional de quantidade de pessoas transportadas.
- Se um transporte for de material, há as informações adicionais de descrição do material, carga (quantidade em toneladas).
- Um transporte pode ser estar em um dos estados: PENDENTE, TRANSPORTANDO, CANCELADO, FINALIZADO. Ao ser cadastrado um transporte fica no estado PENDENTE. Se uma espaçonave for designada para o transporte, este muda para o estado TRANSPORTANDO. Ao chegar ao destino ele torna-se FINALIZADO. Um pedido pode ser CANCELADO se estiver PENDENTE ou TRANSPORTANDO. Se estiver no estado CANCELADO ou FINALIZADO não pode mais mudar.
- A operação *calculaDistancia()* de um transporte calcula a distância entre o espaço-porto de origem e o de destino.
- O cálculo básico do custo de um transporte (operação *calculaCusto()*) é: (custo da distância) X (custo do transportado).
 - O custo da distância é calculado por: distância entre origem e destino X fatorDistância
 - fatorDistância: é 1.000.000 se a distância for menor que 0,5 ano-luz; ou 100 se a distância for igual ou maior a 0,5 ano-luz.
 - O curso do transportado, depende do tipo:
 - pessoas: quantidade de pessoas X 100
 - material: carga X 500

A ACMESpace deseja automatizar o seu negócio e quer que o sistema tenha uma tela cíclica com o usuário para as seguintes funcionalidades:

- **Cadastrar novo espaço-porto** (cadastra os dados do espaço-porto. [Se já houver espaço-porto com o número indicado, apresenta uma mensagem de erro]).
- **Cadastrar nova espaçonave** (cadastra os dados da espaçonave. [Se já houver espaçonave com o nome indicado apresenta uma mensagem de erro]).
- **Cadastrar novo transporte** (solicita o identificador do transporte [se o identificador já existir, apresenta uma mensagem de erro], solicita o tipo do transporte; coloca o novo transporte em uma fila de transportes pendentes; ao final apresenta na tela os dados do novo transporte cadastrado, incluindo os dados da origem e destino, distância e custo).
- **Consultar todos os transportes** (apresenta uma lista com todos os transportes cadastrados: todos os dados dos transportes, incluindo os dados da origem e destino, distância e custo; e dados da espaçonave (se alguma foi designada) [se não há transportes, apresenta uma mensagem de erro]).
- **Alterar estado de transporte** (solicita o identificador de um transporte; mostra os dados do transporte; solicita o novo estado [se não há transporte com o identificador indicado, apresenta uma mensagem de erro; se um transporte estiver no estado CANCELADO ou FINALIZADO, não pode ser alterado e apresenta uma mensagem de erro]).
- **Carregar dados iniciais** (solicita o nome do arquivo (sem extensão); carrega os dados dos arquivos para o sistema; os transportes devem ser carregados em uma fila de transportes pendentes; ao final, mostra todos os dados de espaço-portos, espaçonaves, transportes, [se houver problemas na carga de dados, apresenta uma mensagem de erro]).
- **Designar transportes** (a partir da fila de transportes pendentes; verifica se é possível designar alguma espaçonave disponível e apta para cada transporte, e atualiza o seu estado. Se não for possível designar alguma espaçonave, retorna para a fila de transportes pendentes. [se não há transportes na fila de transportes pendentes, apresenta uma mensagem de erro]).
- **Salvar dados** (solicita ao usuário um nome de arquivo (sem extensão) e salva todos os dados cadastrados em um ou mais arquivos [se houver algum problema no salvamento apresenta uma mensagem de erro]).

- **Carregar dados** (solicita ao usuário um nome de arquivo (sem extensão) e carrega todos os dados de um ou mais arquivos [se houver algum problema no carregamento apresenta uma mensagem de erro]).
- **Finalizar sistema** (termina a execução do sistema).

2. Definição do trabalho:

O objetivo do trabalho é implementar um sistema de gerenciamento de espaçonaves e transportes, capaz de atender as necessidades da empresa descrita no enunciado geral, e que atenda as restrições que seguem:

- É permitida a criação de novas classes, métodos, atributos e relacionamentos, mas as informações definidas no diagrama de classes original não podem ser removidas.
- As classes de negócio (entidades¹) devem possuir teste unitário.
- Deve haver tratamento de exceções para que não haja falhas na execução.
- Deve utilizar coleções da API Java (exceto array []) para o armazenamento de objetos.
- Deve possuir interface gráfica com o usuário.
- Deve ser entregue um relatório do projeto contendo as seguintes seções:
 1. Diagrama de classes – diagrama com as classes do projeto do sistema;
 2. Coleções de dados – descrição das TADs (listas, filas, pilhas, etc.) que foram utilizadas e em que partes do sistema;
 3. Armazenamento (persistência) de dados – descrição da forma escolhida de armazenamento de dados em arquivos.

3. Critérios de avaliação:

- O trabalho será avaliado conforme os seguintes critérios:
 - Nota do Trabalho = 10% apresentação + 90% implementação
 - A implementação envolve:
 - Relatório do projeto: 1 ponto.
 - Testes unitários para as classes de negócio (entidades): 1 ponto.
 - Persistência de dados em arquivos: 1 ponto.
 - Tratamento de exceções: 1 ponto.
 - Interface gráfica com o usuário: 1 ponto.
 - Uso de herança, polimorfismo e coleções: 1 pontos.
 - Implementação e execução do sistema conforme a descrição: 4 pontos.
- Ponto extra (opcional): será atribuído 1 (um) ponto extra, considerando o valor máximo total de 10 (dez) pontos na avaliação, se o sistema ler e gravar os dados em arquivo-texto no formato JSON ou XML.

4. Entrega:

- A entrega do Trabalho Final envolverá:
 - códigos-fonte do sistema (e demais arquivos necessários para a compilação do sistema).
 - relatório do projeto.
- Deverá ser gerado um arquivo compactado (.zip ou .rar), com os itens acima, e entregue pelo Moodle da disciplina.
- A entrega deve ser realizada individualmente, para confirmação do desenvolvimento pelo componente da equipe.
- Data da entrega: **30 / 11 / 2022**.

5. Apresentação:

- Datas de apresentação: **30 / 11 / 2022**.

6. Considerações finais:

¹ Considere como classes de negócio (entidades) aquelas cujos dados são salvos em arquivos.

- O trabalho pode ser desenvolvido individualmente, em dupla ou trio.
- A implementação deve seguir o Java Code Conventions para nomes de identificadores e estrutura das classes.
- Não serão aceitos trabalhos com erros de compilação. Programas que não compilarem corretamente terão nota zerada.
- A cópia parcial ou completa do exercício terá como consequência a atribuição de nota 0 (zero) aos exercícios dos alunos envolvidos. Para análise de similaridade será utilizado o MOSS (<https://theory.stanford.edu/~aiken/moss/>).

Apêndice: exemplos de arquivos de carga de dados iniciais. A partir do nome do arquivo (exemplo: nome do arquivo “EXEMPLO”), cada arquivo de dados terá um sufixo:

- ‘-espacoportos.dat’ para o arquivo de dados de espaço-portos.
 - Formato de cada linha: número do espaço-porto:nome do espaço-porto:coordenada X: coordenada Y: coordenada Z

EXEMPLO-espacoportos.dat

```
numero:nome:x:y:z
11:Terra:0.000000000000000:0.000000000000000:0.000000000000000
22:Marte:-0.00000369002000:0.00000084530000:0.00000900900000
33:Venus:0.00001900000000:0.00001890000000:-0.00002200000000
44:Ganimedes:0.00003400000000:-0.00001600000000:0.00005400000000
55:Alfa Centauri A:2.00200000000000:2.40000000000000:-3.05000000000000
66:Proxima Centauri:3.03000000000000:2.80000000000000:1.00000000000000
77:Epsilon Indi:7.01000000000000:-5.00410000000000:8.10099000000000
88:Glise 412:-10.08000000000000:8.70540000000000:9.00990000000000
99:Sirius :4.09000000000000:5.52000000000000:-5.14000000000000
111:Beta Hydri:15.00090000000000:-14.59900000000000:12.38100000000000
222:Gamma Ceti:-51.23900000000000:26.81950000000000:-55.99590000000000
```

- ‘-espaconaves.dat’ para o arquivo de dados de espaçonaves.
 - Formato de cada linha: tipo da espaçonave (1-subluz, 2-FTL):nome:número do espaço-porto:velocidade de impulso ou WARP:(se tipo 1: combustível, se tipo 2:limite)

EXEMPLO-espaconaves.dat

```
tipo:nome:espacoporto:velocidade:combustivel_limite
1:Alfa:11:0.1:ion
1:Bravo:33:0.3:nuclear
2:Charlie:22:2.1:200
2:Delta:44:3.3:100
```

- ‘-transportes.dat’ para o arquivo de transportes pendentes.
 - Formato de cada linha: tipo do transporte (1-pessoas, 2-material):identificador do transporte:número do espaço-porto da origem:número do espaço-porto de destino:se tipo 1: quantidade de pessoas ou se tipo 2: quantidade de carga:(se tipo 2: descrição))

EXEMPLO-transportes.dat

```
tipo:identificador:origem:destino:quantidade_carga:descricao
1:111:11:22:100
1:222:11:222:200
2:333:22:33:100.10:Dilithium
2:444:33:55:200.22:Kyber cristal
```