

Projeto Prático – 2022/2023 Gestão de Fábrica

OBS: Este tipo de problema pode ser usado em várias situações (Controlo de ESCOLAS, Centralina de um automóvel, Sistema de segurança/gestão de uma fábrica, etc.)

Pretende-se um programa que faça a Simulação da gestão (ao nível da domótica e sistema de controlo de motores) de uma dada fábrica. Esta fábrica tem vários tipos de utilizadores (pessoas com níveis de acesso diferentes e responsabilidades diferentes). Existem na fábrica vários tipos de motores (que é preciso monitorizar), e vários tipos de sensores, que permitem manter um bom estado das instalações

Os motores têm estados típicos: STOP, START, RUN, AVARIADO

Os motores quando em funcionamento, podem aquecer!. Considerem 3 classes de temperatura.

VERDE, AMARELO e VERMELHO.

VERDE: O motor está funcionando bem!

AMARELO: O motor está a aquecer, deve ser enviado um aviso ao gestor;

VERMELHO: O motor está bem quente, então deve ser avisado o Gestor e o motor para se proteger deve ficar no estado STOP.

Para tornar o problema mais próximo da realidade, será dado um ficheiro em XML com a configuração da fábrica e com todo o conteúdo a gerir. Os intervalos de temperatura para cada um dos estados da temperatura será dada no ficheiro XML

Considerações:

A Fábrica tem uma hora de abertura e hora de encerramento;

Quando existe algum alerta/aviso (da parte dos motores ou sensores) o gestor da fábrica deve ser logo informado, para desse modo actuar!;

Cada dispositivo (SENSOR, MOTOR) está numa dada posição (X, Y) na fábrica.

Algumas definições dos dispositivos:

SENSOR (ID, MARCA, VALOR, PROBABILIDADE_AVARIA)

A fábrica tem vários tipos de sensores (LUZ, HUMIDADE, FUMO, MÍSSEIS)

Este sensor de MÍSSEIS, permite detetar, se um dado míssil vem na direção da fábrica....

MOTOR (ID, MARCA, CONSUMO_HORA, TEMPERATURA_AVISO,

TEMPERATURA_PARAGEM, PROBABILIDADE_AVARIA)

Todos os motores têm os métodos RUN; START; RESTART; STOP e ESTOU_QUENTE. Estes métodos podem ser re-implementados por cada um dos motores específicos.

USER (ID, NOME)

ADMIN (Pode fazer: ADD, REMOVE, LISTAR, RUN, MANUTENCAO)

NORMAL (Pode fazer: ADD, LISTAR, RUN, MANUTENCAO)

VISITANTE (Pode fazer: LISTAR)

Funcionalidades Gerais:

- Construtor da Fabrica, sendo dado o USER actual;

Fabrica::Fabrica(USER *ut)

- As configurações da Fábrica são dadas em ficheiro XML, com todas as informações.

bool Fabrica::Load(const string &ficheiro)

- O Simulador deve estar sempre a correr e quando se pretende introduzir alterações, deve-se carregar

- numa tecla 'M' de modo a aparecer um menu (nesse instante o simulador deve estar parado, até a opção ser executada!)
- Adicionar Utilizadores (ter em atenção as regras!)
bool Fabrica::Add(User *ut)
 - Adicionar Motores (Ver quem pode!)
bool Fabrica::Add(Motor *m)
 - Listar o estado atual da fábrica;
void Fabrica::Listar(ostream &f = std::cout)
 - Desligar um motor específico, dado o seu ID;
void Fabrica::Desligar(int id_motor)
 - Saber o estado de um dado Motor, dado o seu ID;
ESTADO_MOTOR Fabrica::Get_Estado(int id_motor)
 - Listar e devolver todos os Motores de um dado Tipo;
List<Motor *> *Fabrica::Listar_Tipo(string Tipo, ostream &f = std::cout)
 - Manutenção de todos os dispositivos (Motores e Sensores!), A manutenção de um motor poderá ser por exemplo alterar a temperatura para um valor (aleatório) dentro da temperatura VERDE do motor;
bool Fabrica::Manutencao()
 - Quais as Marcas de motores/sensores que mais avariaram, deve devolver uma lista (ordenada) do que mais avaria para o mais fiável;
List<string> *Fabrica::Ranking_Dos_Fracos()
 - Quais motores que mais trabalharam, deve devolver uma lista (ordenada);
List<Motor *> *Fabrica::Ranking_Dos_Mais_Trabalhadores()
 - Enviar um relatório em XML, do estado da fábrica;
 - O relatório deverá ter a informação do total de consumo (do motor) nesse dia;**void Fabrica::Relatorio(string fich_xml)**
 - Quando um sensor de Humidade envia um aviso, todos os motores que estão numa dada vizinhança (X) dele devem ser desligados!, o método deve devolver (por parametro) a lista dos Motores que foram desligados, também deve devolver o número de motores que foram desligados;
int Fabrica::Aviso_Humidade(list<Motor *> &lm)
 - Quando um sensor de FUMO envia um aviso, todos os motores devem ser desligados e deve ser invocado um video mostrando as zonas de saída da fábrica;
int Fabrica::Aviso_Fumo(list<Motor *> &lm, string fich_video)
 - No caso de o sensor de LUZ enviar um aviso (por exemplo quando a luz é inferior a um dado valor!) deve passar um video desejando um bom regresso a casa!, pois o tempo está a escurecer e aproxima-se a hora de dormir.
int Fabrica::Aviso_Luz(string fich_video)
 - No caso de o sensor de MÍSSEL enviar um aviso (por exemplo quando o valor for 1) deve passar um video mostrando os abrigos e a fábrica deve desligar-se completamente, guardando o estado atual!.
void Fabrica::Aviso_Missel(string fvideo, string festado = "Estado.txt")

Observações:

- Podem e devem criar novos métodos auxiliares (privados) com os nomes que acharem por bem! E chamar nos métodos principais (públicos);
- Sempre na metodologia orientada a objetos;

Avaliação / Observações:

- Se a gestão da memória, não estiver correcta, haverá uma penalização de 5 valores;
- Se for detetado "copiar", trabalho anulado!
- Os grupos são constituídos por 2 a 3 alunos;
- O código deve estar comentado;
- Eventuais dúvidas serão esclarecidas pelos docentes da disciplina.

Entrega:

- 19/12/2022; para avaliação em época normal (Avaliação ocorre no período das aulas de compensação)
- 02/02/2023; para avaliação em época recurso;

Na página seguinte apresenta-se um exemplo de um ficheiro de configuração de uma fábrica.

```
<DADOS>
  <DEFINICOES>
    <NOME_EMPRESA>Empresa da Minha Terra</ NOME_EMPRESA >
    <HORA_INICIO>8</HORA_INICIO>
    <HORA_FECHO>16</HORA_FECHO>
    <VIZINHANCA_AVISO>10</VIZINHANCA_AVISO>
    <DIMENSAO_FABRICA>50,80</DIMENSAO_FABRICA>
    <MELETRICO>
      <VERDE>0, 80</VERDE>
      <AMARELO>81, 100</AMARELO>
      <VERMELHO>101, 500</VERMELHO>
      <PROB_AVARIA>1</ PROB_AVARIA>
    </MELETRICO>
    <MCOMBUSTAO>
      <VERDE>0, 80</VERDE>
      <AMARELO>81, 100</AMARELO>
      <VERMELHO>101, 250</VERMELHO>
      <PROB_AVARIA>10</ PROB_AVARIA>
    </MCOMBUSTAO>
    <MINDUCAO>
      <VERDE>0, 50</VERDE>
      <AMARELO>51, 70</AMARELO>
      <VERMELHO>71, 100</VERMELHO>
      <PROB_AVARIA>1</ PROB_AVARIA>
    </MINDUCAO>
  </DEFINICOES>
  <MOTORES>
    <MCOMBUSTAO>
      <ID>100</ID>
      <MARCA>Roy</MARCA>
      <CONSUMO_HORA>5</CONSUMO_HORA>
      <POSICAO>10,20</POSICAO>
    </MCOMBUSTAO>
    <MCOMBUSTAO>
      <ID>101</ID>
      <MARCA>HIRTH-SOLO</MARCA>
      <CONSUMO_HORA>3</CONSUMO_HORA>
      <POSICAO>0,10</POSICAO>
    </MCOMBUSTAO>
    <MCOMBUSTAO>
      <ID>102</ID>
      <MARCA>F200</MARCA>
      <CONSUMO_HORA>4</CONSUMO_HORA>
      <POSICAO>15,22</POSICAO>
    </MCOMBUSTAO>
    <MELETRICO>
      <ID>103</ID>
      <MARCA>EFACEC</MARCA>
      <CONSUMO_HORA>1250</CONSUMO_HORA>
      <POSICAO>10,18</POSICAO>
    </MELETRICO>
  </MOTORES>
  <SENSORES>
    <STEMPERATURA>
      <ID>200</ID>
      <MARCA>M1</MARCA>
      <VALOR_AVISO>50</VALOR_AVISO>
      <POSICAO>12,40</POSICAO>
      <PROB_AVARIA>1</PROB_AVARIA>
    </STEMPERATURA>
```

```
<STEMPERATURA>
  <ID>201</ID>
  <MARCA>M1</MARCA>
  <VALOR_AVISO>50</VALOR_AVISO>
  <POSICAO>12,10</POSICAO>
  <PROB_AVARIA>5</PROB_AVARIA>
</STEMPERATURA>
<SFOGO>
  <ID>200</ID>
  <MARCA>M1</MARCA>
  <VALOR_AVISO>1</VALOR_AVISO>
  <POSICAO>8,25</POSICAO>
  <PROB_1>5</PROB_1>
  <PROB_AVARIA>15</PROB_AVARIA>
</SFOGO>
<SMISSEL>
  <ID>300</ID>
  <MARCA>Molotof-Expresso</MARCA>
  <VALOR_AVISO>1</VALOR_AVISO>
  <POSICAO>49,79</POSICAO>
  <PROB_1>3</PROB_1>
  <PROB_AVARIA>4</PROB_AVARIA>
</SMISSEL>
</SENSORES>
</DADOS>
```