**APLICAÇÃO DO PADRÃO PRODUTOR-CONSUMIDOR**

A empresa M&M comercializa móveis por meio da Web de acordo com o seguinte modelo de negócio:

* Há um catálogo único de móveis comercializados pela empresa.
* Há um conjunto de lojas habilitadas a vender os móveis de todo o catálogo.
* Um cliente compra qualquer móvel do catálogo por meio de qualquer uma das empresas habilitadas.
* Os móveis nunca ficam em estoque: são produzidos sob demanda, conforme ocorrem as vendas. Assim que um móvel é produzido, é diretamente transportado para o respectivo comprador.
* Os móveis são produzidos por um conjunto de fabricantes conveniados: cada um dos fabricantes é capaz de produzir qualquer um dos móveis do catálogo.
* Um móvel vendido a um cliente pode ser produzido por qualquer um dos fabricantes conveniados.
* O transporte de um móvel para o respectivo comprador é realizado por um conjunto de transportadoras conveniadas, de forma que qualquer uma das transportadoras pode atender uma demanda de transporte.
* Há uma fila única para os pedidos de compra:
  + Todas as lojas inserem nessa fila
  + Todos os fabricantes removem dessa fila
  + Essa fila é infinita
* Há uma fila única para os pedidos de transporte:
  + Todos os fabricantes inserem nessa fila
  + Todas as transportadoras removem dessa fila
  + Essa fila é finita
* Um fabricante pode fabricar diversos móveis em paralelo, sendo que cada fabricante possui um limite próprio.
* Uma transportadora pode transportar diversos móveis em paralelo, sendo que cada transportadora possui um limite próprio.
* O tempo entre duas vendas consecutivas realizadas por qualquer uma das lojas varia dentro de um intervalo de tempo conhecido.
* O tempo de fabricação de um móvel pode variar dentro de um intervalo conhecido, sendo que cada fabricante pode definir um intervalo de tempo de produção diferente para cada tipo de produto.
* O tempo de entrega de um móvel pode variar dentro de um intervalo conhecido, sendo que cada transportadora pode definir um intervalo de tempo de entrega.

Escreva uma aplicação que implemente o modelo de negócio da empresa M&M.

1. Cada loja é uma thread.
2. Cada fabricante é uma thread.
3. Cada transportadora é uma thread.

A aplicação deve ser simulada com a seguinte configuração:

* Catálogo com oito tipos de produtos: A, B, C, D, E, F, G e H.
* Oito instâncias de loja: A, B, C, D, E, F, G e H.
* Quatro instâncias de fabricante: A, B, C e D.
* Duas instâncias de transportadora: A e B.
* Intervalo de tempo entre duas vendas consecutivas na mesma loja: entre 10 e 150 minutos.
* Intervalo de tempo (em minutos) para a produção de um móvel:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tipo do Produto** | **Fabricante A** | **Fabricante B** | **Fabricante C** | **Fabricante D** |
| A | 600 a 1000 | 400 a 600 | 1000 a 1200 | 800 a 1000 |
| B | 200 a 400 | 800 a 1000 | 1200 a 1400 | 600 a 800 |
| C | 1000 a 1200 | 1200 a 1400 | 400 a 600 | 400 a 600 |
| D | 400 a 600 | 800 a 1000 | 600 a 800 | 1000 a 1200 |
| E | 800 a 1000 | 200 a 400 | 400 a 600 | 1200 a 1400 |
| F | 1400 a 1600 | 1000 a 1200 | 400 a 600 | 800 a 1000 |
| G | 400 a 600 | 1000 a 1200 | 1000 a 1200 | 600 a 800 |
| H | 800 a 1000 | 600 a 800 | 400 a 600 | 1200 a 1400 |

* Intervalo de tempo de entrega de um produto:
  + Transportadora A: entre 100 e 200 minutos
  + Transportadora B: entre 400 e 600 minutos
* Quantidade máxima de produção simultânea de móveis por cada fabricante:
  + Fabricante A: 4
  + Fabricante B: 1
  + Fabricante C: 4
  + Fabricante D: 4
* Quantidade máxima de entrega simultânea de móveis por cada transportadora:
  + Transportadora A: 10
  + Transportadora B: 20

Cada pedido de compra é identificado de maneira única, com a composição da identificação da loja e um número sequencial de venda realizada pela loja. Por exemplo: A10 é a identificação do décimo pedido de compra feito na loja A.

O sistema considera duas medidas de tempo para cada item vendido:

1. Tfabricação: tempo entre o momento do fechamento da venda de um item e o momento da conclusão da fabricação desse item.
2. Tentrega: tempo entre o momento da conclusão da fabricação de um item e o momento da sua entrega ao comprador.

A aplicação deve gerar um relatório que mostre a evolução de Tfabricação  e de Tentrega. Preferencialmente, o relatório deve conter gráficos que exibam essa evolução. Além disso, o relatório deve mostrar a evolução das filas.

**IMPLEMENTAÇÃO INCREMENTAL:**

**VERSÃO 1 [4 pontos]**

1. Classe Venda:
   1. um objeto dessa classe é criado toda vez que uma loja realiza uma venda de um produto do catálogo
   2. deve definir atributos que identifiquem a venda (nome da loja concatenado com um número sequencial fornecido pela loja) e o produto vendido
2. Classe FilaVenda:
   1. há somente um objeto dessa classe, criado no "main", para representar a *fila de vendas*
   2. implementa uma fila que contém instâncias da classe Venda
3. Classe Loja
   1. executa como uma thread
   2. há somente um objeto dessa classe, criado no "main"
   3. possui ao menos dois atributos: nome da loja e um contador de vendas
   4. executa em loop infinito, criando objetos da classe Venda e inserindo-os na fila de vendas
4. Classe Fabricante
   1. executa como uma thread
   2. há somente um objeto dessa classe, criado no "main"
   3. executa em loop infinito, removendo objetos da fila de vendas
5. Método "main":
   1. cria uma instância de FilaVenda para representar a fila de vendas
   2. cria semáforos para controlar o acesso concorrente à fila de vendas
   3. cria uma instância de Loja, fornecendo como parâmetro a fila de vendas e os semáforos para controle de acesso
   4. cria uma instância de Fabricante, fornecendo como parâmetro a fila de vendas e os semáforos para controle de acesso

**VERSÃO 2 [2 pontos]**

1. Classe Entrega:
   1. um objeto dessa classe é criado toda vez que um fabricante termina de confeccionar um produto que foi vendido pela loja
   2. deve definir atributos que identifiquem a entrega a ser realizada, incluindo o identificador da correspondente venda e o produto a ser entregue
2. Classe FilaEntrega:
   1. há somente um objeto dessa classe, criado no "main", para representar a *fila de entregas*
   2. implementa uma fila que contém instâncias da classe Entrega
3. Classe Fabricante:
   1. para cada objeto removido da fila de vendas, cria um objeto da classe Entrega e insere-o na fila de entregas
4. Classe Transportadora:
   1. executa como uma thread
   2. há somente um objeto dessa classe, criado no "main"
   3. executa em loop infinito, removendo objetos da fila de entregas
5. Método "main":
   1. cria uma instância de FilaVenda para representar a fila de vendas
   2. cria semáforos para controlar o acesso concorrente à fila de vendas
   3. cria uma instância de FilaEntrega para representar a fila de entregas
   4. cria semáforos para controlar o acesso concorrente à fila de entregas
   5. cria uma instância de Loja, fornecendo como parâmetro a fila de vendas e os semáforos para controle de acesso
   6. cria uma instância de Fabricante, fornecendo como parâmetro a fila de vendas, a fila de entregas e os semáforos para controle de acesso
   7. cria uma instância de Transportadora, fornecendo como parâmetro a fila de entregas e os semáforos para controle de acesso

**VERSÃO 3 [1 ponto]**

1. Método "main":
   1. cria uma instância de FilaVenda para representar a fila de vendas
   2. cria semáforos para controlar o acesso concorrente à fila de vendas
   3. cria uma instância de FilaEntrega para representar a fila de entregas
   4. cria semáforos para controlar o acesso concorrente à fila de entregas
   5. cria oito instâncias de Loja, fornecendo como parâmetro a fila de vendas e os semáforos para controle de acesso
   6. cria quatro instâncias de Fabricante, fornecendo como parâmetro a fila de vendas, a fila de entregas e os semáforos para controle de acesso
   7. cria duas instâncias de Transportadora, fornecendo como parâmetro a fila de entregas e os semáforos para controle de acesso

**VERSÃO 4 [2 pontos]**

1. Classe Fabricacao:
   1. thread que corresponde à fabricação de um produto vendido
   2. uma instância dessa classe é criada pelo código da thread Fabricante para cada objeto removido da fila de vendas
   3. depois de dormir por um tempo aleatório para simular um tempo de fabricação, a thread deve inserir uma nova instância da classe Entrega na fila de entregas
2. Classe Transporte:
   1. thread que corresponde à entrega de um produto fabricado
   2. uma instância dessa classe é criada pelo código da thread Transportadora para cada objeto removido da fila de entregas
   3. a thread deve dormir por um tempo aleatório para simular um tempo de entrega
3. Classe Fabricante:
   1. cria uma instância da classe Fabricacao para cada objeto removido da fila de vendas
   2. deve incluir um limitador (Padrão Score Board) para o número de threads da classe Fabricacao em execução simultânea
4. Classe Transportadora:
   1. cria uma instância da classe Transporte para cada objeto removido da fila de entregas
   2. deve incluir um limitador (Padrão Score Board) para o número de threads da classe Transporte em execução simultânea

**VERSÃO 5 [1 ponto]**

1. Gera um relatório que mostra a evolução de Tfabricação  e de Tentrega considerando certo período de vendas
   1. para cada dia *i* do período, devem ser calculados os seguintes tempos:
      1. Pi : média entre os tempos de fabricação de todos os itens vendidos no dia i
      2. Ei : média entre os tempos de entrega de todos os itens vendidos no dia i
   2. deve ser construído um gráfico que mostra a evolução de Pi no período
   3. deve ser construído um gráfico que mostra a evolução de Ei no período
2. Gera um relatório que mostra a evolução da fila de vendas e da fila de entregas.