Universidade Federal de Campina Grande
Centro de Engenharia Eletrica e Informática
Departamento de Sistemas e Computação
Graduação em Ciência da Computação

Exercício sobre TAD elementares: vetor, fila e pilha

Objetivo: Praticar a implementação de TADs simples.

O endereço do sistema de submissão é o **https://les.dsc.ufcg.edu.br:8443/EasyLabCorrection**.

Relembre os conceitos de fila e pilha visto em sala de aula. As políticas de acesso de cada TAD (LIFO ou FIFO) fazem a estrutura adequada para problemas diferentes.

Pilha (Stack) e fila (Queue) possuem interfaces diferentes (alguns métodos são em comum), cada um implementando a política específica de cada TAD.

Atividades necessárias antes de iniciar o exercício:

1. Crie um projeto no Eplipse chamado LEDA, por exemplo (pode ser qualquer outro nome que lhe convier);
2. Descompacte o arquivo baixado (exceto o PDF) na pasta dos fontes (normalmente **src**) do seu projeto LEDA criado no seu workspace. O arquivo baixado tem a seguinte estrutura:
   - adt
   -- stack
   --- Stack.java (INTERFACE DA PILHA GENÉRICA)
   --- StackImpl.java (IMPLEMENTACAO PARCIAL DA PILHA GENÉRICA)
   --- StackOverflowException.java (exceção de pilha cheia)
   --- StackUnderflowException.java (exceção de pilha vazia)
   -- queue
   --- Queue.java (INTERFACE DA FILA GENÉRICA)
   --- QueueImpl.java (IMPLEMENTACAO PARCIAL DA FILA GENÉRICA)
   --- CircularQueue.java (IMPLEMENTACAO PARCIAL DA FILA GENÉRICA CIRCULAR)
   --- QueueOverflowException.java (exceção de fila cheia)
   --- QueueUnderflowException.java (exceção de fila vazia)

3. No Eclipse, selecione a pasta dos fontes no projeto LEDA e faça um refresh (apertar F5). Note que deve aparecer um pacote adt.stack contendo or arquivos mencionados acima.

Agora voce está pronto para começar a trabalhar nas seguintes atividades:

1. Observe a interface Stack.java. Ela descreve os serviços de uma pilha genérica.
2. Observe também a existência implementação incompleta StackImpl. Voce precisa implementar os métodos incompletos.
3. Observe a interface Queue.java. Ela descreve os serviços de uma fila genérica.
4. Observe também a existência implementação incompleta QueueImpl. Voce precisa implementar os métodos incompletos.

5. Observe a classe CircularQueue.java. Ela representa a implementação de uma fila circular, para evitar shifts dos elementos na remoção (conforme visto em sala de aula). Voce precisa implementar os métodos incompletos.

Instruções para o envio

Ao terminar o exercício, faça os seguintes passos:

1. Compacte a pasta **adt** que existe nos fontes de seu projeto LEDA (**src**) e retire a classe TestStack.java desse arquivo compactado. A compactação DEVE ser feita a partir do diretório raiz de seus fontes de forma a preservar a estrutura de pastas que refletem a estrutura dos pacotes (package) Java. Por exemplo, voce deve ter um arquivo compactado NOME_COMPLETO_DO_ALUNO.ZIP com a seguinte estrutura:
   - adt
   -- stack
   --- StackImpl.java
   -- queue
   --- QueueImpl.java
   --- CircularQueue.java

2. Envie esse arquivo com sua solução para o sistema de submissão e verifique que o contador de submissões será alterado.

Observações finais:

- **A interpretação do exercício faz parte da atividade.**
- **A atividade é individual. A conversa entre alunos é proibida.**
- **É proibido coletar códigos prontos e adaptar. Implemente as questões. Isso é para seu aprendizado.**
- **Caso voce observe qualquer problema no sistema de submissão, contacte o professor imediatamente.**
- **Se voce nao compactar o arquivo seguindo a estrutura de diretórios a compilação não terá sucesso e o sistema mostrará isso. Erro de compactação serão de responsabilidade do aluno. O professor não ajudará o aluno nesse item. É só seguir as instruções deste arquivo.**