

Trabalho Prático

Estruturas de Dados 1

Prof. Paulo Henrique Ribeiro Gabriel

O objetivo deste trabalho é implementar os TADs Fila e Pilha com o objetivo de aprofundar os conceitos vistos em aula.

Descrição

Observação: Esse trabalho é uma adaptação do trabalho prático proposto pelo professor Dr. Elias P. Duarte Jr. (UFPR).

Um restaurante funciona da seguinte maneira: cada cliente escolhe diferentes itens do cardápio. O cardápio tem todos os itens disponíveis, que são descritos em uma **lista** (por exemplo, um *array*). Cada item aparece na lista com um número de identificação (inteiro), a descrição (nome do item) e o preço (real). Na saída do restaurante, os clientes fazem uma **fila** invariavelmente grande e são atendidos pela gerente do restaurante.

Cada cliente tem uma **comanda** com a lista dos produtos consumidos. Para cada cliente atendido, a gerente informa o preço total, somando os valores dos itens da comanda. Além disso, o restaurante está agradecendo os clientes com um pequeno chocolate. A gerente mantém uma **pilha** de chocolates diversos e entrega ao cliente atendido o chocolate do topo da pilha.

Tarefa

Escreva um programa que implementa o sistema acima. Cada grupo deve preparar cuidadosamente o cardápio do restaurante. Preencha a fila de pessoas, cada uma com sua comanda com a lista dos produtos consumidos. Deve-se encher a pilha com chocolates diversos. Ao final, mostre qual chocolate cada cliente ganhou, bem como quanto pagou pela refeição.

Para as implementações dos Tipos Abstratos de Dados podem ser utilizadas tanto uma estrutura estática sequencial quanto uma estrutura dinâmica encadeada. Cada grupo pode escolher a implementação que julgar mais conveniente. O programa deve ter opções para imprimir a lista, fila ou pilha a qualquer momento.

Execute *testes exaustivos*, mostrando que seu programa funciona nas mais diversas situações (por exemplo, pilha vazia, fila cheia, etc.). Recomenda-se que os dados sejam lidos de arquivos.

Entrega

Cada grupo deverá criar um repositório no GitHub. Nesse repositório, devem estar presentes os seguintes itens:

1. Relatório de como foi feito o trabalho e quais foram os resultados obtidos. Use desenhos, diagramas, figuras, todos os recursos que permitam ao professor compreender como o grupo estruturou o trabalho e quais resultados obteve. O relatório pode ser tanto um arquivo PDF quanto um arquivo README do GitHub (formato .md). Outros formatos não serão aceitos.
2. Código-fonte **comentado**. Se o programa for dividido em diferentes módulos (recomendado), todos os arquivos .c e .h devem estar presentes no repositório.
3. Arquivos do tipo texto (.txt) contendo os casos de teste.
4. Link para um vídeo curto (de 5 a 10 minutos) mostrando a execução de alguns casos de teste.

O link para o GitHub deverá ser disponibilizado para o professor via Moodle, na caixa de texto associada a essa tarefa. Nessa mesma caixa de texto devem estar relacionados o **nome completo** e o **número de matrícula** de **cada membro do grupo**. Caso algumas dessas informações esteja incorreta, será descontado ponto. **Todos os membros do grupo devem submeter essas informações.**

Critérios de Avaliação

O projeto será avaliado principalmente levando em consideração:

1. Funcionamento do programa;
2. Realização das tarefas descritas;
3. Bom uso das técnicas de programação e do TAD;
4. Boa indentação e uso de comentários no código;
5. Boa estruturação e modularização do código;
6. Qualidade da apresentação em vídeo.

Observações

1. O trabalho deve ser feito em **grupos de duas ou três pessoas** e em Linguagem C. Não serão aceitos trabalhos individuais nem desenvolvidos por grupos com mais de três discentes.

2. Não deverá ser utilizada qualquer variável global nem bibliotecas com funções prontas (a não ser aquelas para entrada, saída e alocação dinâmica de memória).
3. Todas as submissões são checadas para evitar plágio; portanto, evite problemas e implemente o seu próprio código.
4. Ainda sobre plágio, a detecção de cópia de parte ou de todo código-fonte, de qualquer origem, implicará reprovação direta no trabalho. Compartilhem ideias, modos de resolver o problema, mas não o código. Qualquer dúvida entrem em contato com o professor.
5. Comente o seu código com uma explicação rápida do que cada função ou trecho importante de código faz (ou deveria fazer). Os comentários e a boa modularização do código serão checados e valem nota.
6. Todos dos trabalhos serão testados. Desculpas do tipo “na minha máquina compilava” serão ignoradas.
7. O vídeo deve ser carregado no Microsoft Stream. Caso o grupo queira, pode torná-lo privado, permitindo acesso apenas aos que possuírem o link. O professor se compromete a não tornar o conteúdo público.
8. O professor pode tirar dúvidas sobre o enunciado, ou sobre a lógica por trás do programa. No entanto, ele não olhará código-fonte em busca de erros! Procure o(s) monitor(es) da disciplina nesse caso.
9. Não serão aceitos trabalhos encaminhados por outro meio que não seja o Moodle.
10. Caso o professor ache necessário, alguns grupos serão convidados a fazer uma apresentação presencial do trabalho.