

# L4: Pilha e Fila

Murilo Dantas

## Exercícios

1. Faça um programa que cadastre 5 números em uma fila dinâmica e mais 5 em uma pilha dinâmica. Em seguida, o programa deve mostrar 3 relatórios. O primeiro terá os números que estão nas duas estruturas. O segundo terá os que estão apenas na fila e o terceiro terá os que estão apenas na pilha.

2. Faça um programa que apresente o menu de opções abaixo:

### MENU

- 1- Cadastrar número
- 2- Mostrar números pares entre o primeiro e o último número cadastrado
- 3- Excluir número
- 4- Sair

### Observações:

- a. O programa deve ser implementado usando uma estrutura do tipo pilha.
- b. A opção 1 do menu cadastra um número de cada vez.
- c. Mostrar mensagem para opção inválida do menu.
- d. Cuidado com o intervalo de números formado pelo primeiro e pelo último número da pilha, pois este pode ser crescente, decrescente ou ainda ser o mesmo número.
- e. Quando a opção do menu não puder ser realizada, mostrar mensagem.

3. Faça um programa que apresente o menu de opções abaixo:

### MENU

- 1- Cadastrar aluno
- 2- Cadastrar nota
- 3- Calcular média de um aluno
- 4- Listar os nomes dos alunos sem notas
- 5- Excluir aluno
- 6- Excluir nota
- 7- Sair

### Observações:

- a. Na opção 1, deve ser cadastrado um aluno (número e nome) de cada vez em uma pilha. A mensagem disponível nesta opção é: Aluno cadastrado. Os números dos alunos devem ser gerados automaticamente, partindo do nº 1.
- b. Na opção 2, deve ser cadastrada uma nota (número do aluno e nota) em uma fila. Uma nota só pode ser cadastrada se pertencer a um aluno cadastrado na pilha de alunos. As mensagens disponíveis nessa opção são: Nota cadastrada e Aluno não cadastrado. Cada aluno pode ter várias notas cadastradas.
- c. Na opção 3, o usuário deve digitar o número de um aluno e o programa deve mostrar o nome dele e a média aritmética das notas desse aluno. As

**mensagens disponíveis nessa opção são: Aluno não cadastrado, Aluno sem notas e Média do aluno = valor calculado.**

- d. Na opção 4, os nomes dos alunos que não possuem notas devem ser listados. As mensagens disponíveis nesta opção são: A listagem dos nomes sem nota e Todos os alunos possuem notas.**
  - e. Na opção 5, um aluno da pilha de alunos de alunos deve ser excluído, respeitando duas regras: (i) um aluno só pode ser excluído se não possuir notas; e (ii) o usuário não deve escolher o aluno a ser excluído, pois a exclusão deve obedecer à lógica da pilha. As mensagens são: Aluno excluído, Pilha vazia e Este aluno possui notas, logo, não poderá ser excluído.**
  - f. Na opção 6, uma nota deve ser excluída, respeitando as regras de funcionamento da fila. As mensagens disponíveis são: Nota excluída e Fila vazia.**
  - g. A opção 7 é a única que sai do programa. Uma mensagem deve ser mostrada para opções inválidas.**
- 4. Faça um programa que cadastre em uma pilha vários números. A entrada deles será finalizada com a digitação de um número menor ou igual a zero. Posteriormente, o programa deve gerar duas filas: a primeira com os números pares e a segunda, com os números ímpares. A saída do programa deve apresentar a pilha digitada e as filas geradas. Caso alguma das filas seja vazia, deve-se mostrar a mensagem.**

- 5. Faça um programa que apresente o menu de opções abaixo:**

**MENU**

- 1- Cadastrar tipo**
- 2- Cadastrar produto**
- 3- Consultar o preço de um produto**
- 4- Excluir tipo**
- 5- Sair**

**Observações:**

- a. Mostrar mensagem de opção inválida do menu. A opção 5 é a única que sai do programa.**
- b. Para a implementação do programa acima, é necessário utilizar duas filas.**
- c. Na primeira fila serão armazenados os tipos de produtos com seus respectivos percentuais de impostos. Verifique que todos os tipos sejam diferentes e cada tipo é identificado com apenas uma letra.**
- d. Na segunda fila serão armazenados os produtos cujo número deve ser gerado automaticamente. O preço e o tipo devem ser digitados. Lembrando que um produto só pode ser cadastrado se for de um tipo também já cadastrado. Faça essa verificação.**
- e. Na primeira opção do menu serão cadastrados os tipos, um de cada vez: cada vez que o usuário escolhe a opção 1 do menu, ele tem a possibilidade de cadastrar um novo tipo (letra que representa o tipo e percentual de imposto). Nesta opção, a única mensagem disponível é: Tipo cadastrado.**
- f. Na segunda opção do menu serão cadastrados os produtos, um de cada vez: cada vez que o usuário escolhe a opção 2 do menu, ele tem a possibilidade de cadastrar um novo produto (número gerado automaticamente, preço e tipo**

digitados). Lembrando que um produto só pode ser cadastrado se o tipo ao qual ele pertence já existe na fila de tipos. Nesta opção, as mensagens disponíveis são: Produto cadastrado e Tipo de produto inexistente.

- g. Na terceira opção do menu, o usuário digita o número do produto que deseja consultar o preço e, se este existir na fila de produtos, o programa deve procurar por seu percentual de imposto, de acordo com o tipo do produto na fila de tipos, calcular e mostrar seu preço, ou seja, preço cadastrado menos o percentual de imposto. Nesta opção, as mensagens disponíveis são: Preço = valor calculado, Produto não cadastrado e Fila vazia.
- h. Na quarta opção, o programa deve excluir um tipo da fila de tipos, respeitando a forma de organização de uma fila. Lembrando que um tipo só pode ser excluído se não existir nenhum produto cadastrado para ele.

Questões extras para quem quiser ir mais fundo no assunto:

- 6. Para checar o balanceamento de parênteses de uma expressão matemática, pode-se utilizar uma pilha de caracteres da seguinte forma: percorre-se a string, de forma toda vez que for encontrado um parênteses, ele é colocado na pilha; quando for encontrado um parênteses de fechamento, retira-se um parênteses de abertura da pilha. Se ao final da expressão a pilha não estiver vazia, os parênteses estão desbalanceados. Implemente uma chamada classe Checador que contenha uma pilha de caracteres e um método chamado balanceada(char \*expressao) que retorne uma das seguintes constantes: OK ou FALTA\_ABRIR\_PARENTESES ou FALTA\_FECHAR\_PARENTESES. Teste suas classes.
- 7. Faça um programa semelhante que inclua a checagem de parênteses, colchetes e chaves, num único método. Para isso, estenda sua classe Checador em uma classe chamada ChecadorCompleto e sobreponha o método balanceada().
- 8. Utilize uma Fila para estudar o seguinte problema. Uma lei obriga que os bancos atendam os clientes em no máximo 30 minutos. Uma situação hipotética seria:
  - A cada minuto, podem entrar aleatoriamente até 5 clientes no banco.
  - Cada caixa demora em média 3 minutos para atender um cliente.
  - Existem 2 caixas disponíveis.

Crie uma classe Banco com uma fila de clientes (numerados em sequência e com “hora” de entrada) que tenha um método que simule a dinâmica do banco. O método de simulação deve receber como parâmetros o número máximo de clientes que pode entrar no banco por minuto, o tempo médio de atendimento do caixa, o número de caixas disponíveis e o tempo de duração da simulação – em minutos. Esse método irá gerar clientes a cada minuto aleatoriamente (até o valor dado) e gerar mensagens do que está acontecendo. Um exemplo de saída seria:

*O cliente 1 foi atendido imediatamente  
O cliente 2 foi atendido imediatamente  
O cliente 3 foi atendido após 2 minutos*

....

*O cliente 120 foi atendido após 1h e 49 minutos.*

---

*Atenção, este banco poderá sofrer até 95 processos no procon se não aumentar o número de caixas.*

*Além disso, foram atendidos apenas 156 clientes e ainda restam 87 clientes na fila*

**Faça a partir daí um programa em que o usuário possa testar várias configurações.**