

Trabalho sobre Métodos de Pesquisa e Ordenação e Linguagem de Programação Orientada a Objetos em Java – Primeira Etapa

Valor

Primeira Etapa: 5,0 pontos na C2

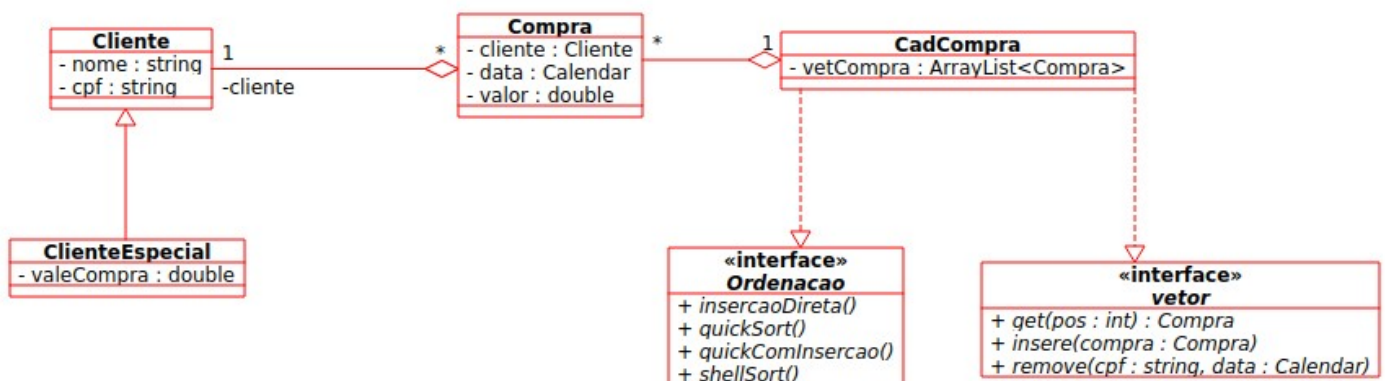
Objetivos

- Avaliar competências e conhecimentos desenvolvidos nas disciplinas de Pesquisa e Ordenação e Linguagem de Programação Orientada a Objetos, por meio de um trabalho experimental.
- Oportunizar o desenvolvimento e escrita de uma atividade nos moldes de um trabalho acadêmico, bem como aperfeiçoar a capacidade de criar, planejar, trabalhar e decidir em grupo de forma cooperativa.
- Aplicar conceitos de POO e de LPOO no desenvolvimento de um programa.
- Aprimorar-se na programação Java.

Obs.: O grupos em que **nenhum dos componentes faz a disciplina LPOO** não precisam cumprir os requisitos de Orientação a Objetos, embora, se cumprirem, a programação será menos exaustiva.

Primeira Etapa

- Implementar o diagrama de classes a seguir, sendo que esse é o padrão mínimo, ou seja, o grupo deverá discutir e implementar métodos que não constam desse diagrama. Por exemplo, na classe CadCompra, o grupo deverá implementar um método para inserir uma Compra no ArrayList



- Data limite para entrega: **30/10** (sábado).
- **O grupo perde 1 ponto por cada aula de atraso na entrega, limitado a duas aulas.**
- Qualquer componente do grupo deve ser capaz de apresentar/defender todas as ideias contidas no trabalho. É fundamental que o trabalho seja feito de forma cooperativa, e não, simplesmente, que cada componente cuide de uma parte do mesmo para depois juntar e formar o todo. As apresentações serão realizadas após a entrega da segunda parte.

- O trabalho deve ser postado no AVA, o código e a parte escrita.

Parte Escrita

A parte escrita deverá ser feita de forma bastante sucinta e deverá conter:

- Uma introdução, descrevendo brevemente cada um dos métodos de Pesquisa e Ordenação implementados (Inserção Direta, Shellsort, Quicksort, Quicksort com Inserção).
- O corpo que deve começar descrevendo a máquina em que o programa for rodado e os arquivos utilizados para teste.
- Fazer quadros de comparação entre o tempo de processamento de cada método implementado, para cada arquivo gerado.
- Conclusões geradas a partir da análise dos quadros comparativos de tempo e verificar se elas são compatíveis ou não com a teoria. Se não for compatível, investigue as causas.
- Se for necessário fazer pesquisas em outros materiais, listar os títulos que auxiliaram o desenvolvimento do trabalho, segundo as normas da ABNT

Segunda Etapa

- A segunda etapa, que trata dos métodos de Pesquisa, será enviada e avaliada na C3.

Critérios avaliativos

- Organização do documento impresso e formatação segundo as normas de trabalho acadêmico.
- Utilização adequada de estruturas de dados e métodos de ordenação.
- Programa correto.
- Legibilidade do código e indentação.
- Entrevista individual e/ou coletiva.
- Trabalho em equipe e de forma cooperativa.
- Pontualidade na entrega (a cada dia de atraso, o trabalho valerá 1,0 ponto a menos).

Texto nas normas da FAESA	0,5
Descrição dos métodos e conclusões	0,5
Implementação dos métodos	3,0
Apresentação (essa nota será dada em novembro, depois da apresentação)	1,0
TOTAL	5,0

Problema a ser implementado

Serão disponibilizados 15 arquivos do tipo texto, contendo 500, 1.000, 5.000, 10.000 e 50.000 registros, dispostos de forma aleatória, ordenada e invertida.

Os registros representarão um cadastro de compras, conforme diagrama acima. O formato do arquivo será

nome;cpf;valeCompra;dataDaCompra;valor

Se for ClienteEspecial. Caso seja Cliente, será:

nome;cpf;dataDaCompra;valor

sendo que dataDaCompra estará no formato dd/mm/aaaa

O trabalho consiste em:

- 1) Comece a contar o tempo.
- 2) Carregue o vetor com o arquivo de 500 elementos aleatórios.
- 3) Use o método **Inserção Direta** para ordenar os registros pelo cpf. Se tiver cpfs iguais, ordene pela data da compra. Grave o resultado da ordenação em um arquivo. (O nome do arquivo pode ser InsDir500Alea.txt)
- 4) Repita 4 vezes o processo de 2 e 3. **Você deve rodar o processo 5 vezes no total.** Os arquivos gerados podem ser regravados, ou seja, no final você terá apenas um arquivo para cada tamanho, tipo e método.
- 5) Termine de contar o tempo, faça uma média e armazene este resultado.
- 6) **Faça os itens de 1 a 5 para cada um dos tamanhos (500, 1000, 5000, 10000 e 50000), para cada tipo de arquivo (aleatório, ordenado e invertido) e para cada método (Inserção Direta, ShellSort, QuickSort e Quick com Inserção).** Ao todo, o programa rodará 60 vezes e criará 60 arquivos.
- 7) Compare os tempos de todos os algoritmos em cada tamanho e tipo de arquivo e gere conclusões.

Obs.:

- 1) Para computar o tempo, utilize o método `System.currentTimeMillis()` que retorna o tempo da máquina em milissegundos, ou `System.nanoTime()` que retorna o tempo da máquina em nanossegundos.