|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **PREENCHA SEUS DADOS NOS CAMPOS COM X** |  | |
|  |
| Disc.: | **ECM306 – tópicos avançados em estrutura de dados** | |
| Curso: | **ENGENHARIA DA COMPUTAÇÃO** | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Aluno: | | **X** | | | | | | | | |  |
| Curso: | **X** | | | Série: | **X** | Período: | **X** | RG: | **X** | |
| São Caetano do Sul, 18 de setembro de 2023. | | | | | | | | | RA: | **X** |
| Assinatura: | | |  | | | | | | Nota: |  |

**Instruções da Prova**

1. Esta prova é individual e prática, devendo ser realizada nos computadores do IMT, sendo permitido ao aluno, se assim desejar, utilizar seu próprio computador, sob sua inteira responsabilidade;
2. Não poderá haver acesso à Internet, sob nenhuma circunstância, exceto ao Open LMS do próprio aluno e, mesmo assim, somente em duas etapas: para receber (“baixar”) as questões de sua prova na máquina em uso; e para entregar (“subir”) as resoluções das questões de sua prova no devido local de entrega, na mesma plataforma.

**IMPORTANTE**: O aluno deverá informar ao professor quando fará os dois acessos permitidos a ele ao Open LMS, pela Internet;

1. Poderá haver consulta a qualquer material do próprio aluno, seja ele físico (livros, artigos, material de aula etc.) ou virtual (livros, artigos, material de aula, exercícios, resoluções etc.), desde que esse material esteja previamente armazenado em seu computador ou em qualquer dispositivo de armazenamento externo (*pendrive*, *hd* externo etc.).

**IMPORTANTE**: Não será permitido o acesso pela Internet a pastas compartilhadas (*Google* *Drive*, *OneDrive* etc.), nem a repositórios virtuais (*GitHub*, *GitLab*, *BitBucket* etc.), mesmo sendo de posse e administração do próprio aluno;

1. O aluno deverá responder às questões da prova no próprio arquivo da prova (.*docx*) devidamente identificado (RA e Nome completo do Aluno) e quando for o caso, gerando os códigos necessários, somente na linguagem de programação JAVA e utilizando sempre e somente o paradigma da Programação Orientada à Objetos visto nas aulas;
2. Para realizar a entrega da prova na plataforma Open LMS, o aluno deverá gerar e entregar um único arquivo compactado (*.rar* ou .*zip*), tendo seu RA e seu NOME completo como nome desse arquivo. Nesse arquivo compactado o aluno deverá fornecer, obrigatoriamente, os seus dados acadêmicos preenchidos na prova, bem como a resolução das questões nela solicitadas (.*docx*), além dos arquivos e códigos gerados em suas resoluções, uma pasta para cada questão, contendo as suas classes e demais arquivos que possibilitem sua posterior execução pelo professor, durante a resolução.

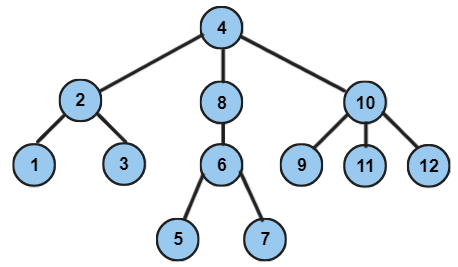
**IMPORTANTE**: Na correção, para executar o código gerado pelo aluno em sua prova, o professor seguirá exatamente as instruções fornecidas pelos alunos e contidas nas resoluções das provas! Caso não obtenha sucesso, a questão será considerada errada.

1. Não poderá haver troca de informações, nem de materiais, sejam físicos ou virtuais, entre os alunos durante a prova;
2. Não é permitido ao estudante se ausentar da sala antes da entrega da prova;
3. Celulares e outros equipamentos eletrônicos (exceto o notebook do aluno, se assim optar) devem permanecer desligados enquanto o estudante estiver na sala;
4. O tempo limite para realização da prova é de **90** minutos;
5. Mantenha sobre a carteira apenas um documento com foto, caneta, lápis e borracha;
6. O entendimento das questões faz parte da avaliação;
7. O tempo mínimo de permanência na sala é de **30** minutos;
8. O estudante que chegar atrasado em até **30** minutos do início da prova poderá fazê-la.

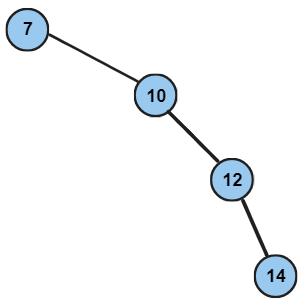
**Questões de 1 à 3**: (*1 ponto cada*)

Dadas as figuras 1, 2 e 3 abaixo, verifique se cada afirmação de 1 à 9 é *Verdadeira* ou *Falsa* e depois responda as questões, de 1 à 3:

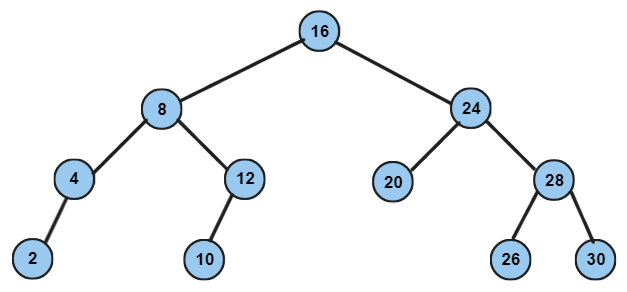
FIGURAS:



*Figura 3*



*Figura 2*



*Figura 1*

AFIRMAÇÕES:

A1 – Dada a representação da **Árvore Binária** da figura 1, onde o nó 16 é sua raiz, quando percorrida através do algoritmo **PRÉ-ORDEM**, o resultado encontrado é: ***2, 4, 10, 12, 8, 20, 26, 30, 28, 24, 16***

A2 – Dada a representação da **Árvore Binária** da figura 1, onde o nó 16 é sua raiz, quando percorrida através do algoritmo **IN-ORDEM**, o resultado encontrado é: ***2, 4, 8, 10, 12, 16, 20, 24, 26, 28, 30***

A3 – Dada a representação da **Árvore Binária** da figura 1, onde o nó 16 é sua raiz, quando percorrida através do algoritmo **PÓS-ORDEM**, o resultado encontrado é: ***16, 8, 4, 2, 12, 10, 24, 20, 28, 26, 30***

A4 – Dada a representação da **Árvore Binária** da figura 2, onde o nó 7 é sua raiz, quando percorrida através do algoritmo **PRÉ-ORDEM**, o resultado encontrado é: ***14, 12, 10, 7***

A5 – Dada a representação da **Árvore Binária** da figura 2, onde o nó 7 é sua raiz, quando percorrida através do algoritmo **IN-ORDEM**, o resultado encontrado é: ***7, 10, 12, 14***

A6 – Dada a representação da **Árvore Binária** da figura 2, onde o nó 7 é sua raiz, quando percorrida através do algoritmo **PÓS-ORDEM**, o resultado encontrado é: ***14, 12, 10, 7***

A7 – Dada a representação da **Árvore** da figura 3, pode-se afirmar que o **nó 7** foi o **último** a ser inserido na estrutura, pois é o nó com maior profundidade.

A8 – Dada a representação da **Árvore** da figura 3, pode-se afirmar que o **nó 12** foi o **último** a ser inserido na estrutura, pois é o nó que fica mais à direita.

A9 – Dada a representação da **Árvore** da figura 3, pode-se afirmar que o **nó 3** foi inserido **posteriormente** ao **nó 1**, pois o nó 3 está à direita do nó 1.

QUESTÕES:

Q1 – Assinale a única alternativa correta:

a) As afirmações A1, A2 e A3 são verdadeiras;

b) Apenas a afirmação A2 é verdadeira, dentre A1, A2 e A3;

c) As afirmações A1, A2 e A3 são falsas;

d) Apenas a afirmação A1 é falsa, dentre A1, A2 e A3;

e) N.D.A.

Q2 – Assinale a única alternativa correta:

a) As afirmações A4, A5 e A6 são verdadeiras;

b) Apenas a afirmação A5 é verdadeira, dentre A4, A5 e A6;

c) As afirmações A4, A5 e A6 são falsas;

d) Apenas a afirmação A4 é falsa, dentre A4, A5 e A6;

e) N.D.A.

Q3 – Assinale a única alternativa correta:

a) As afirmações A7, A8 e A9 são verdadeiras;

b) Apenas a afirmação A9 é verdadeira, dentre A7, A8 e A9;

c) As afirmações A7, A8 e A9 são falsas;

d) Apenas a afirmação A7 é falsa, dentre A7, A8 e A9;

e) N.D.A.

**Questão 4:** (4 *pontos*)

Gerado por qualquer *IDE*, fornecer o código fonte, em *Java*, das classes necessárias (incluindo a de execução), implementadas sob o paradigma da *Programação Orientada a Objetos*, além do programa ***.jar*** que as executa diretamente, objetivando resolver o problema descrito a seguir.

ATENÇÃO:

1. Se o programa *.jar* fornecido não executar automaticamente a aplicação desenvolvida, a resolução da questão será invalidada (0 ponto);
2. As classes soluções desta questão (arquivos *.java*) deverão ser compactadas em um único arquivo (.*zip* ou .*rar*), em conjunto com o respectivo arquivo *.jar* funcional, além deste arquivo *.docx* da prova, contendo todas as respostas para as outras questões e a devida identificação do aluno e entregue em resposta à tarefa do *OPEN LMS* da prova.

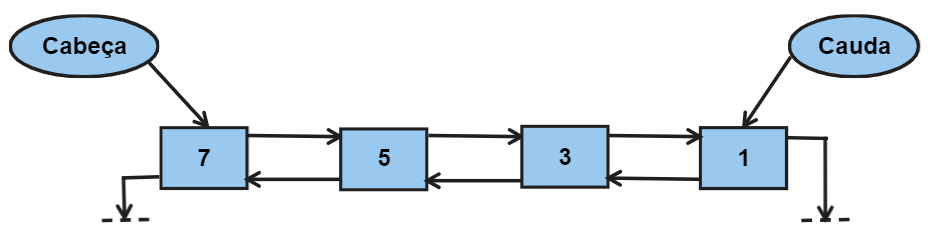
Problema:

Utilizando-se apenas os códigos vistos e praticados em aula e os códigos desenvolvidos pelo próprio aluno para esta disciplina, através de uma aplicação implementada durante a prova, deseja-se criar uma **Estrutura de Dados – ED** dinâmica e linear, baseada em **Listas Duplamente Ligadas**, com cabeça e cauda, capaz de armazenar um número inteiro por nó.

Essa ED deverá nascer **vazia** e, a cada solicitação do usuário (via teclado), a ela deverá ser inserido um número inteiro, entre 1 e 100, randômico, um valor por nó, um após o outro, sendo que a ED deverá se manter ordenada de forma **decrescente** (com relação a esses números inteiros) a cada **inserção** de um **novo** **nó**.

Após cada inserção e a respectiva ordenação, a ED deverá ter o conteúdo de seus nós **impressos** em tela, em sequência, sempre da cabeça para a cauda, ou seja, ordenados decrescentemente.

Casos de Teste:



1. Ao se iniciar o programa, a ED deverá estar vazia e apresentar: “Lista Vazia!”;
2. Ao inserir o nó 5, a ED deverá mostrar tamanho 1 e apresentar seu nó 5;
3. Ao inserir o nó 1, a ED deverá mostrar tamanho 2 e apresentar seus nós: 5, 1;
4. Ao inserir o nó 3, a ED deverá mostrar tamanho 3 e apresentar seus nós: 5, 3, 1;
5. Ao inserir o nó 7, a ED deverá mostrar tamanho 4 e apresentar seus nós: 7, 5, 3, 1.

**Questão 5:** (3 *pontos*)

Gerado por qualquer *IDE*, fornecer o código fonte, em *Java*, das classes necessárias (incluindo a de execução), implementadas sob o paradigma da *Programação Orientada a Objetos*, além do programa ***.jar*** que as executa diretamente, objetivando resolver o problema descrito a seguir.

ATENÇÃO:

1. Se o programa .jar fornecido não executar automaticamente a aplicação desenvolvida, a resolução da questão será invalidada (0 ponto);
2. As classes soluções desta questão (arquivos *.java*), deverão ser compactadas em um único arquivo (.*zip* ou .*rar*), em conjunto com o respectivo arquivo *.jar* funcional, além deste arquivo *.docx* da prova, contendo todas as respostas para as outras questões e a devida identificação do aluno e entregue em resposta à tarefa do *OPEN LMS* da prova.

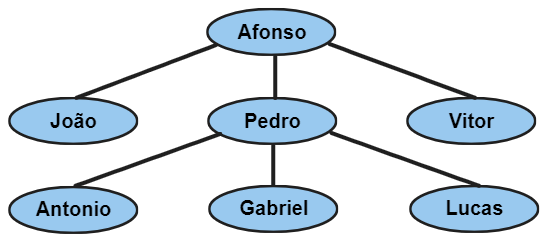
Problema:

Utilizando-se apenas os códigos vistos e praticados em aula e os códigos desenvolvidos pelo próprio aluno para esta disciplina, através de uma aplicação implementada durante a prova, deseja-se criar uma **Estrutura de Dados – ED** dinâmica e linear, baseada em **Árvores**, capaz de armazenar uma ***String*** por nó, representando seu nome.

Essa ED deverá nascer **vazia** e, a cada solicitação do usuário (via teclado), a ela serão informados o **nome do novo nó filho** em questão e o **nome do pai desse nó filho**, um após o outro, sendo que a ED deverá inserir esse novo nó filho abaixo do seu nó pai, no local destinado a ele, seguindo as respectivas regras para a inserção de um **novo** **nó**.

Após cada inserção e sua a respectiva alocação, a ED deverá ter o conteúdo de todos os seus nós **impressos** em tela, em sequência, através de um percurso em **PRÉ-ORDEM**.

Casos de Teste:



1. Ao se iniciar o programa, a ED deverá mostrar tamanho 0 e apresentar: “Árvore vazia!”;
2. Ao inserir o nó “Afonso”, com null de pai (sem pai), a ED deverá mostrar tamanho 1 e apresentar: “Afonso”;
3. Ao inserir o nó “João”, com nó “Afonso” de pai, a ED deverá mostrar tamanho 2 e apresentar: “Afonso, João”;
4. Ao inserir o nó “Pedro”, com nó “Afonso” de pai, a ED deverá mostrar tamanho 3 e apresentar: “Afonso, João, Pedro”;
5. Ao inserir o nó “Antonio”, com nó “Pedro” de pai, a ED deverá mostrar tamanho 4 e apresentar: “Afonso, João, Pedro, Antonio”;
6. Ao inserir o nó “Vitor”, com nó “Afonso” de pai, a ED deverá mostrar tamanho 5 e apresentar: “Afonso, João, Pedro, Antonio, Vitor”;
7. Ao inserir o nó “Gabriel”, com nó “Pedro” de pai, a ED deverá mostrar tamanho 6 e apresentar: “Afonso, João, Pedro, Antonio, Gabriel, Vitor”;
8. Ao inserir o nó “Lucas”, com nó “Pedro” de pai, a ED deverá mostrar tamanho 7 e apresentar: “Afonso, João, Pedro, Antonio, Gabriel, Lucas, Vitor”.