|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **PREENCHA OS CAMPOS EM AMARELO** | **PROVA P3** | |
|  |
| Disc.: | **ECM251 – lINGUAGEM DE PROGRAMAÇÃO i** | |
| Curso: | **ENGENHARIA DA COMPUTAÇÃO** | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Aluno: | | **Pedro Wilian Palumbo Bevilacqua** | | | | | | | | | Test Code:  **MCJQP** |
| Curso: | **ECM** | | | Série | **3ª** | Período: | **Matutino** |  |  | |
| São Caetano do Sul, 23 de setembro de 2025. | | | | | | | | | RA: | **23.01307-9** |
| Assinatura: | | | Pedro Wilian Palumbo Bevilacqua | | | | | | 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 CANVAS 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 CANVAS 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 de sua posse (*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 CANVAS 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, *Smartphones*, *Smartwatches* e outros equipamentos eletrônicos 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*)

Verifique se cada afirmação, de A1 à A9, é *Verdadeira* ou *Falsa* e depois responda as questões, de Q1 à Q3:

AFIRMAÇÕES:

A1 – A classe *JComboBox*, utilizada para criar listas *drop-down*, gera eventos do tipo *ItemEvent*, assim como *JCheckBox* e *JRadioButton*.

A2 – O componente *JPasswordField* herda diretamente da classe *JComponent* e não possui métodos adicionais em relação ao *JTextField*.

A3 – Para internacionalizar uma aplicação *Java* é necessário criar *arquivos.properties* para cada idioma, seguindo o padrão <*nomearquivo*>\_<*linguagem*>\_<*país*>.*properties*.

A4 – A classe *ResourceBundle* é utilizada para carregar os arquivos de internacionalização e o método *getString*() recupera os valores associados às chaves.

A5 – O acesso a arquivos de texto em *Java* é limitado apenas à leitura; a linguagem não fornece mecanismos para escrita em arquivos.

A6 – Na criptografia de chave simétrica, a mesma chave é utilizada para cifrar e decifrar as mensagens, tornando o processo geralmente mais rápido do que a criptografia assimétrica.

A7 – Um dos principais desafios da criptografia por chave simétrica é o gerenciamento seguro e a distribuição da chave secreta entre as partes envolvidas.

A8 – O *Smoke* *Test* é um tipo de teste que verifica de forma abrangente e exaustiva todas as camadas da aplicação antes de ser entregue ao usuário final.

A9 – O Teste de Integração é focado em verificar a interação entre diferentes módulos ou componentes da aplicação, garantindo que funcionem corretamente em conjunto.

QUESTÕES:

Q1 – Assinale a única alternativa correta:

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

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

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

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

e) N.D.A.

Q2 – Assinale a única alternativa correta:

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

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

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

d) Apenas a afirmação A5 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 falsas;

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

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

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

e) N.D.A.

**Questão 4:** (*7 pontos totais*)

Gerado por qualquer *IDE*, sem o uso de Inteligência Artificial, utilizando-se como base apenas os códigos e técnicas vistos e praticados durante as aulas da disciplina, desenvolver uma aplicação e 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*, de geração obrigatória**, que as executa diretamente, objetivando resolver o problema descrito a seguir.

ATENÇÃO: As classes desta questão (arquivos *.java*) deverão ser compactadas em um único arquivo (.*zip* ou .*rar*), em conjunto com o respectivo arquivo *.jar* funcional, com os arquivos de trabalho da aplicação (arquivos textos *.txt*, arquivo de chave criptográfica e arquivos cifrados e decifrados gerados), além deste arquivo *.docx* da prova, contendo todas as respostas para as outras questões, com a devida identificação do aluno e entregue em resposta à tarefa do *CANVAS LMS* da prova.

**Problema**:

Parte I (1 ponto) - *Geração da Interface Gráfica do Usuário (GUI) e dos feedbacks de operação:*

A aplicação deverá ter uma *GUI* orientada a menus (sem utilização de *JOptionPane*);

Todos os *feedbacks* (retorno de cada opção escolhida e/ou processada) e informações necessárias ao usuário deverão ser apresentados na área de mensagens dessa *GUI*.

Todos os comandos e execuções da aplicação deverão ser disparados através e somente de cliques do *mouse* sobre itens de menus nessa *GUI*, obrigatoriamente do tipo:

1. “***Mensagem***”, com subitens: “***Ler arquivo de mensagem***”;

“***Gerar e mostrar CS***”;

“***Salvar arquivo de mensagem com CS***”;

1. “***Chaves***”, com subitens: “***Gerar e salvar chave pública***”;

“***Gerar e salvar chave privada***”;

1. “***Criptografia****”*, com subitens: “***Cifrar e salvar a mensagem com CS cifrada****”*;

“***Decifrar e salvar a mensagem com CS decifrada***”;

1. “***Geral***”, com subitens: “***Mostrar a mensagem com CS decifrada***”;

“***Sair***”.

Parte II (1+1+1 pontos) – *Subitens de “****Mensagem****”*:

Ao clicar sobre o item “***Mensagem***”, seguido pelo subitem “***Ler arquivo de mensagem***”, o aplicativo deverá ler o arquivo denominado “***msg\_original.txt***”, armazenado na mesma pasta da aplicação, contendo qualquer texto digitado e salvo previamente por um editor de texto;

Ao clicar sobre o item “***Mensagem***”, seguido pelo subitem “***Gerar e mostrar CS***”, o aplicativo deverá calcular o valor do *CheckSum* da mensagem lida, pelo método ***Soma e Complemento de 2***, e mostrar o resultado do cálculo na *GUI*;

Ao clicar sobre o item “***Mensagem***”, seguido pelo subitem “***Salvar arquivo de mensagem com CS***”, o aplicativo deverá salvar, na mesma pasta da mensagem lida, um arquivo texto, denominado “***msg\_com\_cs.txt***”, contendo a mensagem lida seguida do valor do *CheckSum* calculado anteriormente.

Parte III (0,5+0,5 ponto) – *Subitens de “****Chaves****”*:

Ao clicar sobre o item “***Chaves***”, seguido pelo subitem “***Gerar e salvar chave pública***”, o aplicativo deverá gerar e salvar em arquivo, denominado ***chave.publica***, uma chave pública, através do **Algoritmo de criptografia RSA** de chaves assimétricas, na mesma pasta da aplicação;

Ao clicar sobre o item “***Chaves***”, seguido pelo subitem “***Gerar e salvar chave privada***”, o aplicativo deverá gerar e salvar em arquivo, denominado ***chave.privada***, uma chave privada, através do **Algoritmo de criptografia RSA** de chaves assimétricas, na mesma pasta da aplicação.

Parte IV (0,5+0,5 ponto) – *Subitens de “****Criptografia****”*:

Ao clicar sobre o item “***Criptografia***”, seguido pelo subitem “***Cifrar e salvar a mensagem com CS cifrada***”, o aplicativo deverá cifrar, com o **Algoritmo de criptografia RSA** de chaves assimétricas e com sua chave pública gerada e salva anteriormente no arquivo ***chave.publica***, a mensagem contida no arquivo “***msg\_com\_cs.txt***“, gerado anteriormente, salvando o conteúdo cifrado resultante no arquivo “***msg\_cifrada.txt***”, na mesma pasta da aplicação;

Ao clicar sobre o item “***Criptografia***”, seguido pelo subitem “***Decifrar e salvar a mensagem com CS decifrada***”, o aplicativo deverá decifrar, com o **Algoritmo de criptografia RSA** de chaves assimétricas e com sua chave privada gerada e salva anteriormente no arquivo ***chave.privada***, a mensagem contida no arquivo “***msg\_cifrada.txt***“, gerado anteriormente, salvando o conteúdo decifrado resultante no arquivo “***msg\_decifrada.txt***”, na mesma pasta da aplicação.

Parte V (0,5+0,5 ponto) – *Subitens de “****Geral****”*:

Ao clicar sobre o item “***Geral***”, seguido pelo subitem “***Mostrar a mensagem com CS decifrada***”, o aplicativo deverá apresentar na *GUI*, de uma só vez, o conteúdo lido do arquivo gerado anteriormente “***msg\_decifrada.txt***”;

Ao clicar sobre o item “***Geral***”, seguido pelo subitem “***Sair***”, o aplicativo deverá fechar a *GUI* e encerrar o seu processamento.