

Programação Orientada a Objetos
4º Período – 2020 – 1º Semestre
Prof. Me. Marcos R Moraes
[maromo]
marcos.moraes6@fatec.sp.gov.br

OBJETIVO

O objetivo principal do trabalho é propiciar aos alunos um embasamento prático dos conceitos teóricos adquiridos através dos conteúdos programáticos ministrados neste semestre em torno de uma atividade de desenvolvimento de software em linguagem Java no Paradigma de Orientação a Objetos.

Como a disciplina ocorreu de forma remota, alguns alunos ficaram impossibilitados, às vezes, de frequentar a aula online. Deverão recorrer às gravações feitas das aulas, para sanar dúvidas dos pontos apresentados durante a *live* pelo professor.

METODOLOGIA DO TRABALHO

O trabalho deve ser desenvolvido em equipe e caberá a cada equipe a responsabilidade de desenvolver o trabalho em sua totalidade. As equipes deverão ser compostas de no **máximo 5 (cinco) alunos**.

O trabalho final deverá ser entregue até o dia 07 de julho, impreterivelmente, não serão aceitos trabalhos fora deste prazo, haja visto os prazos que o professor tem para realizar acertos e fechamento da disciplina junto à Instituição. O grupo deverá escolher **dois** entre os três problemas apresentados pelo professor, **ou um deles**, caso o grupo queira apresentar uma **proposta alternativa**, sendo obrigatório, o desenvolvimento de dois programas.

Projeto deve utilizar conceitos de programação orientada a objetos como: classes, objetos, métodos, atributos, atributos estáticos, coleções, gerenciamento de exceções, interface gráfica com o usuário. Ao menos 80% destes conceitos devem estar presentes no projeto.

ROTEIRO DO TRABALHO

Desenvolver em linguagem Java.

Criar um roteiro de apresentação (powerpoint, por exemplo), com objetivos, tecnologias usadas, apresentação das funcionalidades e uma avaliação crítica do trabalho.

Gravar a apresentação dada no roteiro em uma plataforma streaming, no Teams você pode gravar como reunião e compartilhar o link do Stream

(<https://web.microsoftstream.com/studio/videos>) - fica protegido dentro da Instituição. Coloque o professor como um dos proprietários.

O código deve ser enviado na plataforma **Github** e enviado o link para o professor junto com a **apresentação em formato ppt ou pptx**.

Nota 1: A apresentação deve conter uma avaliação crítica que é parte importante do projeto, pois é o momento em que o grupo realiza uma análise do trabalho, destacando sua coerência, originalidade, exposição da ideia central e argumentos.

Nota 2: O vídeo não pode ultrapassar 15 minutos.

Abaixo os problemas solicitados pelo professor para a solução do grupo de alunos.

Caso 1

A Fatec Arthur de Azevedo, por determinação do Centro Paula Souza, e principalmente motivado pela pandemia do Covid-19, precisará realizar o vestibular de inverno online. Por essa razão, sua equipe foi contratada para criar um programa que capture as respostas de todos os candidatos, e dado um gabarito oficial classifique cada candidato com sua pontuação de acerto e em ordem de pontuação da maior para o menor.

Sabe-se que você receberá para a classificação um arquivo chamado **respostas.txt** com a seguinte informação:

```
12134 bcdadceeaabdeaeaabcdadceeeadeaeaabcdaceeeeababceabceda
12131 cbabcadeacbaeabdeaeaabcdaeaaabdaabdeaeaabcedaaabdeae
12233 abdeaeaabcdaeaaabdaabdeaeadeaeaabcdaeaaabdaabdeaeaaa
13311 cbabcadeacbaeabdeaeaabcdaeaaabdaabdeaeaabcedaaabdeae
14411 eabdeaeaabcdaeaaabdaaeaaabcdaeaaaeaeaabcdaeaaabdaabde
```

Será dado um arquivo chamado **gabarito.txt**, com a seguinte informação
bcdadceeeadeaeaabcdaceeeeababceabdeaeaabcdaeaaabdbabced

Resposta deverá ser no seguinte formato
CCCCC XX

onde CCCCC representa o número do candidato e XX representa o número de questões certas. A classificação deverá ser gravada em um arquivo chamado **resultado.txt**

Nota

O problema deve ser resolvido totalmente orientado a Objetos, e você deverá pesquisar sobre as classes em Java, File, InoutStream, OutPutStream, RamdomAccessFile, Reader, Writer, documentadas no final deste arquivo.

Caso 2

Desenvolver um programa que simule um carrinho de supermercado. A equipe usará como base duas estruturas: a produto e a carrinho.

Classe : Produto

Campos membros: codigo, descricao, precounit

Classe: Item

Campos membros: numitem, Produto, qtd, precototalItem

Classe: Carrinho

Campos membros: dataCompra, nomeCliente, precototal, List<Item>

Fazer um programa principal que simule a abertura de um carrinho de compras e que nele pode-se adicionar quantos itens forem necessários de produtos, e ao final deve imprimir na tela um resumo da compra feita com o preço total por item e o total da compra (do carrinho). Estes dados devem ser gravado em um arquivo chamado compra.txt.

Não é necessário rigidez em relação aos atributos, você está livre para acrescentar o que achar pertinente.

Nota

O problema deve ser resolvido totalmente orientado a Objetos, e você deverá pesquisar sobre as classes em Java, File, InoutStream, OutPutStream, RandomAccessFile, Reader, Writer, documentadas no final deste arquivo.

Caso 3

Projeto Conta Bancária

Dadas duas classes:

```
Classe Cliente{  
    int idCliente;  
    String nomeCliente;  
}
```

```
Classe Conta {  
    long  numeroConta;  
    Cliente cliente;  
    double saldo;  
}
```

Crie um programa que permita gravar as informações de um cliente e de várias contas que esse possua. Uma vez aberta a conta, faça um programa que simule, operações de depósito, saque, e exibição de saldo para cada conta do cliente.

Não é necessário rigidez em relação aos atributos, você está livre para acrescentar o que achar pertinente.

Notas:

A) você deve considerar como regra para sacar que só é feito o saque se o cliente possuir saldo, e para depositar a regra é que o valor seja positivo.

B) Todo o histórico de transacoes devem ser gravados em um arquivo **resultado.txt**.

C) O problema deve ser resolvido totalmente orientado a Objetos, e você deverá pesquisar sobre as classes em Java, File, InoutStream, OutPutStream, RamdomAccessFile, Reader, Writer, documentadas a seguir.

Classes para tratar Streams

Object

- o File
- o InputStream
 - o FileInputStream
 - o FilterInputStream
 - o DataInputStream
 - o OutputStream
 - o FileOutputStream
 - o FilterOutputStream
 - o DataOutputStream
- o RandomAccessFile
- o Reader
 - o BufferedReader
 - o InputStreamReader
 - o FileReader
- o Writer
 - o OutputStreamWriter
 - o FileWriter
 - o PrintWriter

Classes para entrada ou saída baseada em bytes

Classes para entrada ou saída baseada em caracteres