Candidato: Igor Pereira Dourado

Matrícula PUCRS: 19204004

Linguagem de programação para desenvolvimento: Java

EXPLICAÇÕES

Autoavaliação: Gostei de praticar fazendo esse código, consegui explorar e demonstrar bastante conceitos que aprendi em POO, técnicas de programas, projeto de algoritmo etc. Como pontos fortes no meu desenvolvimento eu destacaria principalmente isso, também que de forma rápida eu consegui idealizar uma ou mais soluções para cada tipo de funcionalidade exigida no dicionário e consegui fazer com que desse certo e de maneira eficiente no código, apesar de extenso (que posso destacar como um ponto negativo ou positivo) o código está bem detalhado; consegui usar bastante bem as Collections e bibliotecas do Java, entendendo a parte em que cada uma deveria entrar e também como pensar e usá-las de maneira eficiente. Como pontos que tive certa dificuldade está ligado a parte de pensar nas exceções que poderiam ser geradas por erro do usuário e em criar uma persistência para cada parte em questão. Também na hora da testagem, a meu ver ficaram testes um pouco óbvios, porém que de fato testa as funcionalidades dos métodos sem necessariamente ter casos de “pleonasmo” nos testes, mesmo em número reduzido.

Em primeiro lugar criei meu projeto com o Maven, para criar de forma correta todas as dependências e a área de testes, logo após fiz o escopo do código e minhas ideias em papel e aí então comecei a colocá-las em código, como mostrarei no decorrer desse documento.

Observações:

\*O código está em grande parte comentado, então optei por mostrar aqui mais meu pensamento para cada questão e alguns comentários, o código direto da questão, dos métodos que interferem diretamente neles, de um exemplo de execução e dos testes correspondentes a questão. Então no código há comentários mais específicos sobre cada parte do código e por qual motivo usei cada biblioteca ou método em cada parte do código, há métodos gerais que não interferem diretamente na questão e alguns outros testes desses métodos.

\*Creio que consegui tratar qualquer exceção que pode vim a acontecer no programa por parte do usuário (com Try\_Catch e While) e informei a mensagem adequada de como prosseguir e como não cometer o erro novamente.

Leitura do CSV:

Texto

Descrição gerada automaticamente

Utilizei a biblioteca de java.io que normalmente uso para ler arquivo e consegui ler o arquivo CSV sem precisar usar nenhuma biblioteca mais complexa, durante o código eu expliquei em comentários o motivo de eu ter escolhido cada biblioteca e Collections específicas em cada parte, por exemplo, na List de “bolsistas” eu escolhi utilizar um ArrayList ao invés de LinkedList pois as operações de Get e Add são de complexidades O(1), ao contrário da LinkedList, ou seja, são mais eficientes. O mesmo ocorreu no map de “bolsistaZero”, em que escolhi HashMap ao invés de TreeMap por razões semelhantes. Por sinal, como comentei no código, eu criei esse dicionário para associar um determinado ano a uma lista de bolsistas no qual o bolsista tem esse ano como ano de referência e uma lista de bolsistas referentes associados a cada ano. Foi minha estratégia pensada para a questão 1 e 3 que usam do atributo de “anos” e fazer uso de poucas operações (e de baixa complexidade de tempo) para deixar o código mais simples, eficiente e rápido nas operações.

Questão 1- Explicação:

Tela de computador

Descrição gerada automaticamente

Minha estratégia na questão 1 foi me utilizar do Map que tem como “key” os anos e como “value” um ArrayList com os bolsistas que tem como ano aquele ano digitado pelo usuário e então pegar pela “key” o ano digitado pelo o usuário e verificar se ele consta no dicionário, se constar, peguei o último valor da Lista referente aquele ano por conta da seguinte lógica: como está ordenado por ano do mais recente até o ano mais antigo, interpretei que o primeiro bolsista cadastrado no programa foi o último a aparecer no CSV, logo ele seria o bolsista zero do GERAL e segui essa lógica para cada ano, pegando o último bolsista de cada ano no CSV original e considerando como o bolsista Zero.

Exemplo de execução e resultado:

Interface gráfica do usuário, Texto, Site

Descrição gerada automaticamente

Testes unitários para a questão 1:

Texto

Descrição gerada automaticamente

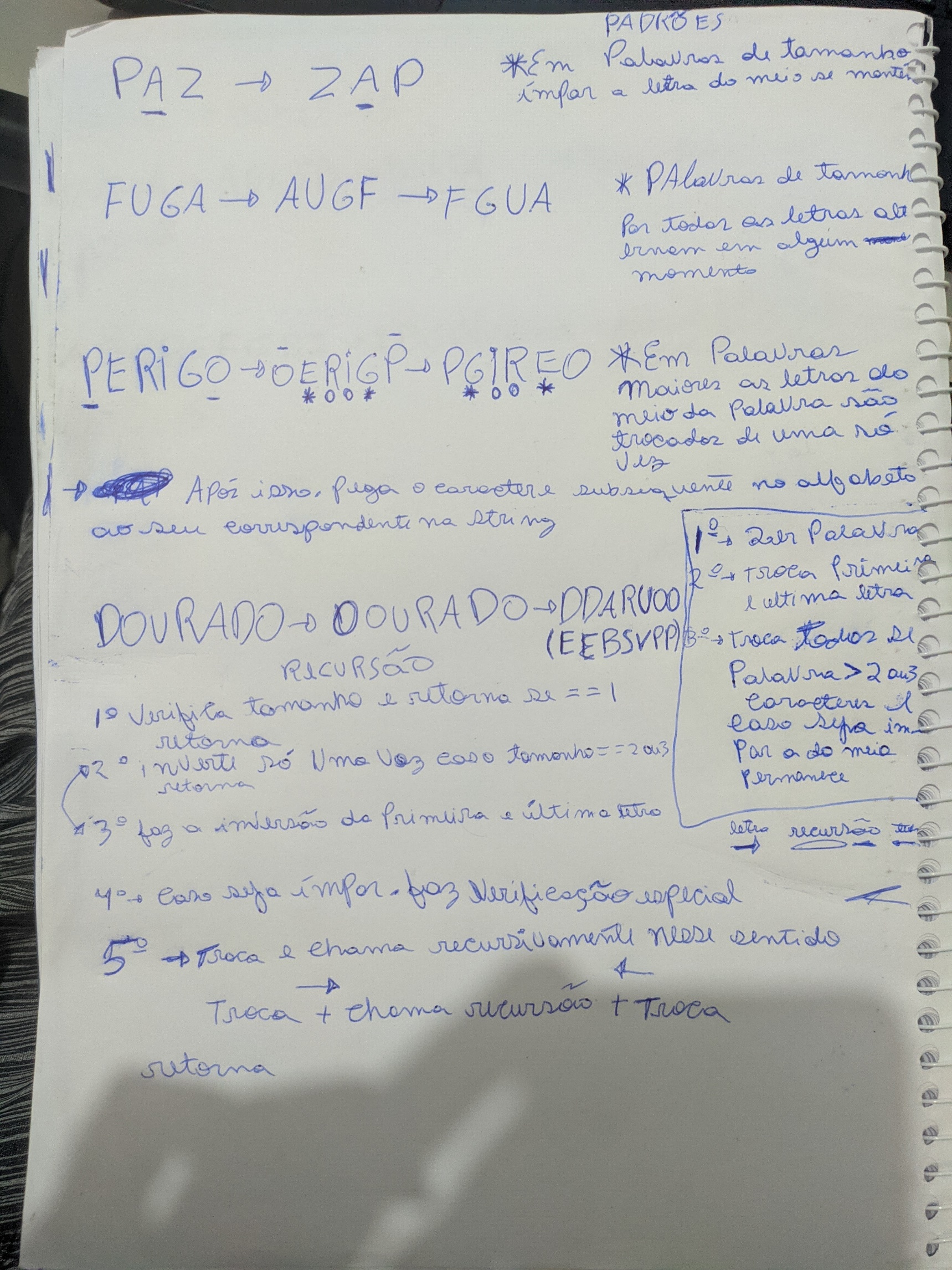
Questão 2- Explicação:

O usuário pesquisa por um nome e como pode ter mais correspondências para uma mesma pesquisa de nome, criei uma lista de possíveis candidatos e exibi na tela contendo apenas os 2 últimos sobrenomes de cada bolsista, visto que sigilo dos bolsistas é uma questão importante na questão. Então, se houver, o programa listará as opções e o usuário deve escolher o que contenha os 2 últimos sobrenomes do nome que procura, sem saber o nome completo de outros, e só então ele entrará na parte da codificação.

Tela de computador

Descrição gerada automaticamente

Pra primeira parte da codificação eu pensei de duas formas: uma utilizando o método reverse, visto que a palavra será revertida em um segundo ponto ou utilizar recursão, visto que ocorre seguidas troca, mas achei interessante explorar a recursão nesse ponto e pensei da seguinte forma:



Tela de computador

Descrição gerada automaticamente

Logo após apliquei a máscara para “adicionar 1” a cada caractere da palavra primeiramente codificada e ficou seguinte forma:

Texto

Descrição gerada automaticamente

E então foi codificado o nome do usuário e mantido em sigilo, enquanto suas informações foram exibidas.

Execução e testes unitários da questão 2 e seus métodos:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Questão 3- Explicação:

Aqui novamente me utilizei do Map (dicionário) criado lá em cima, pesquiso o ano como key do Map e, se constar, é retornado uma lista com todos os bolsistas daquele ano. Então aí eu somo todos eles daquele ano e armazeno em uma variável e no final eu divido pelo total de bolsistas que constam naquele ano, resultando na média de valores de bolsa do ano pesquisado, que depois eu formatei para ficar no padrão de moeda brasileira ou mostra a mensagem que não consta bolsistas naquele ano pesquisado.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Execução e testes unitários da questão 3:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Questão 4 – Explicação:

Para essa questão eu pensei desde a leitura do CSV (como aconteceu com as questões anteriores também) e criei a cópia do ArrayList de bolsistas para que a ordem original não seja alterada, e então eu ordenei por valor de pagamento da bolsa esse ArrayList copiado em ordem decrescente com um comparador que implementei na classe de Bolsista e já ordenei na leitura mesmo. Após essa List ser ordenada, no método da questão, dependendo da opção escolhida pelo o usuário para a exibição eu peguei os 3 últimos (mais baixos) ou os 3 primeiros (mais altos) e printei suas informações, dependendo da escolha do usuário, é possível printar ambos.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Tela de computador

Descrição gerada automaticamente

Execução e testes unitários da questão 4:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Questão 5- Demonstração saindo do programa:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Referências que usei em partes de dúvida durante o código:

how to know if a string is contained in another java:

https://stackoverflow.com/questions/2275004/in-java-how-do-i-check-if-a-string-contains-a-substring-ignoring-case

imprimir acento na tela:

https://itqna.net/questions/11564/allow-special-characters-java-keyboard-input

como fazer para ignorar caracteres especiais no java:

https://stackoverflow.com/questions/31716511/how-to-ignore-special-characters-and-spaces-in-string

How to swap String characters in Java?

https://stackoverflow.com/questions/956199/how-to-swap-string-characters-in-java

How do I increment a variable to the next or previous letter in the alphabet?

https://stackoverflow.com/questions/2899301/how-do-i-increment-a-variable-to-the-next-or-previous-letter-in-the-alphabet

Comparator

https://beginnersbook.com/2013/12/java-arraylist-of-object-sort-example-comparable-and-comparator/

Testes unitários:

https://www.youtube.com/watch?v=60yrTfVdFwo&ab\_channel=software\_technology