Desafio de Programação – PUCRS DELL IT Academy

Lucas Sartori Piatnicki

Centro de Pesquisa em Engenharia de Software – CePES

Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

Av. Ipiranga, 6681 – Prédio 32 – sala 110  
CEP: 90619-900 – Porto Alegre - RS – Brasil

lucas.piatnicki@edu.pucrs.br

# 1. Problema

# Guia do Vendedor Interespacial. Vide arquivo outerspace.pdf para mais informações.

# 2. *Brainstorming* inicial

Entradas (E):

snob representa I

krok representa V

squid representa X

leij representa L

snob snob Prata valem 34 créditos

snob krok Ouro valem 57800 créditos

squid squid Iron valem 3910 créditos

quanto vale squid leij snob snob ?

quantos créditos são snob krok Silver ?

quantos créditos são snob krok Gold ?

quantos créditos são snob krok Iron ?

quanto vale wood could woodchuck mood ?

Saídas (S):

squid leij snob snob vale 42

snob krok Silver custa 68 créditos

snob krok Gold custa 57800 créditos

snob krok Iron custa 782 créditos

Nem ideia do que isto significa!

Etapa 1: as linhas E1-7 são *premissas*; as linhas E8-12 são *hipóteses* e as linhas S1-5 são *provas*.

Etapa 2: a linha S1 é o output da linha E8; nesta última, as strings que se encontram entre as duas primeiras (**quanto vale**) e a última (**?**) compõem a expressão **squid leij snob snob**, da qual é necessário extrair a representação das linhas 3, 4, 1 e 1, nesta ordem, e então aglutinar os valores, para obter o número em romanos.

Etapa 3: para realizar o output S1, é preciso copiar a expressão **squid leij snob snob** da linha E8, adicionar a string **vale** à expressão, e em seguida adicionar o número resultante do método de conversão de romanos para decimais, utilizando o número romano **XLII** (que é o resultado da Etapa 2).

# 3. Desenvolvimento

Vide arquivo **README.md**.

# 4. Solução apresentada

A tentativa de solução desenvolvida por mim (em Java) foi fruto de esforço pessoal, sem caráter colaborativo. A implementação foi realizada por etapas (não necessariamente as mesmas que as expostas na seção 2 acima), em incrementos de sofisticação. Partindo da realização da leitura e da interpretação da linha E1, passei para a leitura de todas as linhas (E1-12), detendo-me na linha de input E8 e sua respectiva conversão para o output S1, seguindo a lógica de conversão. O código é composto de 5 classes, sendo uma a App. Por motivo de falta de tempo hábil, as saídas S2-5 não puderam ser verificadas. Os arquivos de entrada denominam-se **single.txt** e **multiple.txt** (dependendo do método), e o arquivo de saída gerado denomina-se **output.txt**. Para mais detalhes, vide o corpo do código e o repositório do projeto no *GitHub*, cujo endereço web é <https://github.com/lucaspiat/outerspace>.

Instruções para execução: extrair a pasta **outerspace** no disco local, executar o *Visual Studio Code*, escolher a opção **Open Folder...**, e selecionar a pasta **outerspace**. Abrir o terminal nesta pasta e executar com o comando ./mvnw spring-boot:run. A pasta que contém as classes do programa e a classe **App** encontram-se na sub-pasta **\src\main\java\com\pucrs\outerspace\**. Se for executar fora do *Spring Boot*, remover a linha 11 da classe **OuterspaceApplication.java**:

SpringApplication.run(OuterspaceApplication.class, args);

e então compilar a App e executar com o Java 11 ou superior.