

Trabalho Prático

Individual ou em Dupla
Data Entrega: 09/12/2022

Objetivo: Aplicar conhecimentos adquiridos em sala de aula em um enunciado mais elaborado e inserido no contexto mercadológico.

Pokemon Go

Pokemon Go foi lançado em 6 de julho e teve um dos maiores crescimentos e engajamentos da história dos jogos digitais. Estima-se que tenha sido baixado aproximadamente 500 milhões de vezes apenas nos EUA. Estima-se que o jogo arrecade em média US\$ 2 milhões ao dia.

Evoluindo seus Pokémons

Para evoluir seus Pokemons, um jogador precisa acumular Candies. Quando um Pokemon evolui, ele se torna mais forte. Cada Pokemon precisa de uma quantidade diferente de Candies para evoluir. Apesar do jogo considerar diferentes tipos de Candy para cada família de Pokemon, neste TI consideraremos apenas a quantidade de Candies necessários, sem nos preocuparmos com o tipo do Candy.

Definição do problema

Você deseja criar um sistema que gerencie a lista de Pokemons de um usuário. Considere a tabela de evoluções de Pokemons disponível em <http://www.pokemongoevolution.com/>, armazenada em um arquivo texto com o seguinte formato:

Pokemon de Origem	Número de Candies necessários	Pokemon evoluído
-------------------	-------------------------------	------------------

Parte do arquivo texto pode ser visto a seguir:

```
Abra;25;Kadabra  
Aerodactyl;0;Não evolui  
Arbok;0;Não evolui  
Arcanine;0;Não evolui  
Articuno;0;Não evolui  
Bellsprout;25;Weepinbell  
Bulbasaur;25;Ivysaur  
Caterpie;12;Metapod
```

Chansey;0;Não evolui Charmander;25;Charmeleon Charmeleon;100;Charizard Clefable;0;Não evolui Clefairy;50;Clefable Cloyster;0;Não evolui Cubone;50;Marowak Dewgong;0;Não evolui Diglett;50;Dugtrio Ditto;0;Não evolui

Cada linha armazena uma regra de evolução. Os dados são separados por ponto-e-vírgula (;). O primeiro campo contém o nome do Pokemon original. O segundo campo contém o número de Candies necessário para evoluir o Pokemon. O terceiro campo contém o nome do Pokemon evoluído.

Alguns Pokemons não possuem forma evoluída. Neste caso, o 2º campo terá o valor de candies igual a 0 (zero) e o nome do Pokemon evoluído será “Não Evolui”.

Agora considere o inventário da mochila de Pokemons de um jogador no seguinte formato:

Nome do Pokemon	Quantidade disponível	Número de candies disponível para a evolução
-----------------	-----------------------	--

Por exemplo, o arquivo:

Bellsprout;3;200 Bulbasaur;4;90 Caterpie;70;324 Charmeleon;1;233 Doduo;2;80 Horsea;4;130

Esse arquivo indica que o usuário possui, por exemplo, 2 Doduos e 80 candies compatíveis para evolução de Doduos. Após cada evolução, os candies remanescentes devem ser transportados para o Pokemon evoluído. Por exemplo:

A tabela de evolução de um Doduo é:

Doduo	50	Dodrio
-------	----	--------

Se a mochila original possui:

Doduo	2	80
-------	---	----

A mochila resultante deve possuir:

Doduo	1	0
Dodrio	1	30

Ou seja: A mochila anterior irá produzir a seguinte mochila resultante:

Weepinbell;2;0 Victreebel;1;25 Bulbasaur;1; 0 Ivysaur;3;15 Caterpie;70;0 Metapod;27;0 Charizard;1;133 Doduo;1;0 Dodrio;1;30 Horsea;2;0 Seadra;2;30
--

Faça:

- 1- Crie a estrutura de dados necessária para armazenar a tabela de evolução de Pokemons disponível no arquivo **evolucao.txt**
- 2- Crie no bloco de notas um arquivo chamado **mochila.txt** com a composição de uma mochila de um usuário de Pokemon Go, com os campos: nome, quantidade, candies. Os campos devem estar numa mesma linha e serem separados por ponto-e-vírgula.
- 3- Crie a estrutura de dados necessária para armazenar a mochila com as informações carregadas do arquivo criado.
- 4- Crie os seguintes subprogramas:
 - a. Função que lê os arquivos e os armazena na memória;
 - b. Procedimento que evolui todos os Pokemons possíveis e produza uma mochila resultante com os Pokemons evoluídos, a quantidade de Pokemons de cada tipo e a quantidade de Candies restante;
 - c. Procedimento que armazene a mochila evoluída em um arquivo texto com o mesmo formato da mochila original;
 - d. Procedimento que exibe menu com as opções relacionadas com os subprogramas anteriores. As opções são: carregar tabela de evolução; carregar mochila; evoluir mochila; exibir mochila evoluída no console e gravar mochila evoluída em arquivo.
- 5- No programa principal, faça as declarações, leituras e exibição do menu até que o usuário solicite o término do programa. A cada opção escolhida no menu, os resultados correspondentes devem ser exibidos.

OBS.: Deve-se separar o código de interface da lógica do processamento do programa. Defina métodos auxiliares que julgar necessário para modularizar adequadamente seu programa.

- 6- O código também deverá estar **legível e documentado**:
 - i. Comentários - os comentários devem acrescentar alguma informação útil (não fazer comentário do tipo “repetição”, “condição”, “se $x > 0$ ”).
 - a. Incluir comentário antes do início do código contendo: objetivo do programa, nome do programador, data de escrita e da última atualização;
 - b. Incluir comentário no início de cada método, explicando o que faz e como deve ser usado
 - ii. Indentação - utilize indentação para mostrar a estrutura lógica do programa.
 - iii. Nomes de variáveis e métodos - escolha nomes representativos.
 - iv. Espaços e linhas em branco – utilizar espaços e linhas em branco para melhorar a legibilidade, por exemplo, depois de comentários.
 - v. Simplicidade da solução – sempre que possível, optar pela forma mais simples de resolver um problema.

- 7- Um vídeo postado no YouTube (não listado) com a apresentação do programa funcionando e a explicação das principais partes do código, com duração máxima de 10 minutos (não serão aceitos vídeos que extrapolem esta duração);
- 8- Os trabalhos deverão ser postados no Canvas – código completo com documentação e link para o vídeo no YouTube.

Entrega

Deverão ser entregues pelo Canvas:

1. Valor total do trabalho: 20,0 pontos.
 - i. Corretude do programa – 10,0
 - ii. Legibilidade do programa – 4,0
 - iii. Documentação e Vídeo – 6,0
2. O trabalho pode ser feito individual ou dupla.

Observações: não deixe para a última hora. Não serão aceitos trabalhos enviados por e-mail ou por chat.

Observação: Trabalhos copiados receberão nota zero, incluindo conjuntos de trabalhos considerados muito semelhantes.