

Fundamentos de Programação

Trabalho 2 - PokePedia

Grupos de até 3 pessoas
Entrega: até 23:59:59 de 21/06/2018

PokePedia

O objetivo deste trabalho é programar uma versão bem simples de uma enciclopédia baseada no jogo Pokémon!

A ideia da PokePedia é armazenar entradas de dados sobre Pokémons para eventuais consultas, além de fazer simulações simples de batalhas entre times para tentar prever qual time vencerá.

Funcionalidades

- Ao abrir a PokePedia, deve-se carregar os dados de todos os Pokémons cadastrados anteriormente
 - os dados cadastrados devem ser mantidos de forma persistente, ou seja, utilizando arquivos
- Cadastro de novos Pokémons
 - informações a serem armazenadas: número, nome, tipo (apenas um tipo), espécie, altura, peso, e geração em que foi criado
 - * obs: os dados não precisam estar corretos, apenas precisam estar preenchidos de alguma forma
- Remoção de um Pokémon
 - remove permanentemente os dados de um Pokémon cadastrado
- Buscar Pokémons a partir de número, nome ou tipo
 - a busca por número retorna o Pokémon com aquele número
 - a busca por nome retorna todos os Pokémons para os quais a string buscada seja uma parte do nome
 - a busca por tipo retorna todos os Pokémons daquele tipo
- Deve permitir que o usuário monte dois times de 6 Pokémons cada para se enfrentarem em uma batalha
 - cada time deve ser montado separadamente (não deve solicitar os dois times de uma vez)
- Deve simular uma batalha simples entre os dois times selecionados, utilizando apenas os tipos dos Pokémons para efetuar o combate

- para esta simulação, utiliza-se um Pokémon por vez, seguindo a ordem em que o time foi montado
- na simulação, compara-se os tipos dos dois Pokémon, e ganha o Pokémon cujo tipo é efetivo contra o outro
- o Pokémon mais fraco é eliminado, sendo substituído pelo próximo Pokémon do time
 - * ex: água ganha de fogo, terra ganha de elétrico, ...
- se nenhum dos Pokémon é forte contra o outro, os dois empatam e são eliminados
- a batalha chega ao final quando acabam os Pokémon de uma das equipes
- se uma das equipes ainda tem Pokémon e a outra não, esta equipe é declarada vitoriosa
- se as duas equipes ficam sem Pokémon ao mesmo tempo (quando há um empate na última luta), é declarado um empate

Pontos que serão avaliados

(a) (1.0 pontos) Utilização de arquivos

- (0.5) os dados sobre os Pokémon devem ser mantidos mesmo após fecharmos o programa, portanto ao abrir o programa deve-se carregar todos os dados anteriormente inseridos no sistema
 - para isto, armazene os dados em um arquivo
 - dica: você pode criar um vetor de muitas posições (ex: 1000, 10000) e copiar os dados do arquivo para o vetor ao abrir o programa
 - * ao fazer alguma operação nos dados de Pokémon (cadastro, alteração, remoção), copie os dados do vetor para o arquivo!
- (0.5) se o arquivo ainda não existe (primeira vez que se abre o programa), deve-se criar um novo arquivo vazio
 - para isto, basta abrir o arquivo em um modo de escrita e fechá-lo logo em seguida

(b) (0.5 pontos) Menu inicial

- (0.5) o menu deve listar as opções de funcionalidades, cada qual com um número diferente, e solicita um número ao usuário para acessar alguma das funcionalidades
 - sugestão: inserir também uma opção para sair do programa
- caso o usuário insira uma opção inválida, o programa deve mostrar uma mensagem de erro e solicitar outra opção

(c) (0.5 pontos) Dados sobre cada Pokémon

- (0.5) cada Pokémon deve ter número, nome, tipo (apenas um tipo), espécie, altura, peso, e geração em que foi criado
 - o número deve ser um inteiro positivo, e dois Pokémon não podem ter o mesmo número
 - o tipo de um Pokémon deve ser um tipo válido (por exemplo, não podemos ter um Pokémon do tipo "presidente")
 - você deve utilizar pelo menos três tipos válidos (ex: fogo, água e planta)

(d) (1.0 pontos) Cadastro de um novo Pokémon

- (0.5) para o cadastro de um novo Pokémon, deve-se solicitar ao usuário cada uma das informações sobre o Pokémon
- (0.5) o programa deve tratar os seguintes casos:

- se o usuário inserir um número já utilizado, zero, ou negativo, o programa deve mostrar uma mensagem de erro e solicitar o número novamente
 - obs: se o usuário inserir um tipo inválido (que não seja aceito no sistema), o programa deve mostrar uma mensagem de erro e solicitar o tipo novamente
- (e) (1.0 pontos) Remoção de um Pokémon cadastrado
- (0.5) para a remoção de um Pokémon, deve-se solicitar o número do Pokémon a ser alterado, e deve-se confirmar que deseja removê-lo
 - (0.5) o programa deve tratar os seguintes casos:
 - se o usuário desistiu de remover, pode digitar o número 0 para sair da parte de remoção de um Pokémon
 - se o usuário inserir um número negativo, o programa deve mostrar uma mensagem de erro e solicitar o número novamente
- (f) (0.5 pontos) Listar todos os Pokémons cadastrados no sistema
- você pode optar por mostrar todos os dados de cada Pokémon cadastrado, ou mostrar só parte dos dados (ex: mostrar só número, nome e tipo)
- (g) (2.5 pontos) Buscar por Pokémons no sistema
- (0.5) o programa deve permitir uma busca por nome
 - a busca deve retornar todos os Pokémons para os quais o valor da busca ocorra no nome
 - * ex: buscando por "char", podemos encontrar "charmander", "charmeleon" e "charizard"
 - dica: função "strstr"!
 - (0.5) o programa deve permitir uma busca por tipo
 - neste caso, deve-se listar todos os Pokémons do tipo correspondente
 - (0.5) o programa deve permitir uma busca por número
 - neste caso, deve-se listar todas as informações do Pokémon encontrado
 - (0.5) você deve evitar problemas com maiúsculas e minúsculas na busca
 - dica: utilize apenas letras minúsculas
 - * neste caso, antes de buscar, transforme todas as letras da palavra a ser buscada, para minúsculas
 - (0.5) caso a busca não encontre nenhum Pokémon, deve-se imprimir uma mensagem informando sobre isso
- (h) (1.0 pontos) Montar dois times de 6 Pokémons cada
- (0.5) o programa deve permitir que o usuário monte dois times de 6 Pokémons cada para simular uma batalha
 - para montar o time, o programa deve solicitar ao usuário o número de cada Pokémon
 - obs: monte um time por vez
 - * exemplo: ao entrar nesta opção, o programa pergunta qual time você deseja montar: 1 ou 2
 - obs: os times podem repetir Pokémons
 - (0.5) obs: ao selecionar esta opção, se não houver nenhum Pokémon cadastrado, o programa deve apresentar uma mensagem de erro e voltar ao menu inicial
- (i) (2.0 pontos) O programa deve simular uma batalha entre os dois times montados
- (0.5) a batalha deve seguir a ideia apresentada na parte de "Funcionalidades" deste PDF

- (0.5) o programa deve indicar quais Pokémon se enfrentaram, e qual o resultado de cada duelo
 - ex: charmander vence bulbasaur
 - ex: charmander perde para squirtle
 - ex: pikachu empata com bulbasaur
 - sugestão: Pokémons do time 1 aparecem à esquerda e Pokémons do time 2 aparecem à direita
- (0.5) ao final da batalha, deve-se identificar o time vencedor, ou se houve empate
- (0.5) o programa só pode efetuar a simulação de batalha se houverem dois times selecionados
 - ao selecionar esta opção, o programa deve verificar se há dois times montados, e se não houver, mostra uma mensagem de erro e volta ao menu principal

Comentários adicionais

- a nota será dada avaliando o cumprimento ou não de cada subtópico, tratando cada um individualmente como feito ou não feito
 - assim como foi feito no jogo da velha, ou seja, sem casos intermediários
- apesar de recomendado, não é necessário limpar a tela a cada nova interação com o programa
 - se você desejar, pode utilizar o comando `system("@cls||clear")` da biblioteca "stdlib.h"
- lembrando: o prazo é até meia-noite de quinta-feira, 21/06/2018
 - apesar disto, quem tiver terminado o trabalho antes pode apresentar no laboratório nas aulas do dia 20/06 e 21/06
 - * a aula de 21/06 não é em laboratório, então vou tentar conseguir um outro laboratório para ficarmos neste dia, e confirmo se conseguir
 - * todos os trabalhos, tanto os que forem apresentados no laboratório quanto os que não, devem ser enviados via e-mail para ffrdamasceno@gmail.com
- tente seguir uma ideia intuitiva quando for escolher quais tipos são fortes contra quais
 - por exemplo, uma pessoa que nunca jogou Pokémon na vida responderia intuitivamente que água ganha de fogo
 - dica: utilize a tabela a seguir, que indica quais tipos são fortes contra quais no jogo

| Tipo\Forte contra | Normal | Fogo | Água | Elétrico | Planta | Gelo | Lutador | Veneno | Terra | Voador | Psíquico | Inseto | Pedra | Fantasma | Dragão | Noturno | Metálico | Fada |
|-------------------|--------|------|------|----------|--------|------|---------|--------|-------|--------|----------|--------|-------|----------|--------|---------|----------|------|
| Normal | | | | | | | | | | | | | | | | | | |
| Fogo | | | | | | | | | | | | | | | | | | |
| Água | | | | | | | | | | | | | | | | | | |
| Elétrico | | | | | | | | | | | | | | | | | | |
| Planta | | | | | | | | | | | | | | | | | | |
| Gelo | | | | | | | | | | | | | | | | | | |
| Lutador | | | | | | | | | | | | | | | | | | |
| Veneno | | | | | | | | | | | | | | | | | | |
| Terra | | | | | | | | | | | | | | | | | | |
| Voador | | | | | | | | | | | | | | | | | | |
| Psíquico | | | | | | | | | | | | | | | | | | |
| Inseto | | | | | | | | | | | | | | | | | | |
| Pedra | | | | | | | | | | | | | | | | | | |
| Fantasma | | | | | | | | | | | | | | | | | | |
| Dragão | | | | | | | | | | | | | | | | | | |
| Noturno | | | | | | | | | | | | | | | | | | |
| Metálico | | | | | | | | | | | | | | | | | | |
| Fada | | | | | | | | | | | | | | | | | | |