

Prova 1

Você foi contratada(o) para implementar o código de um aplicativo para o jogo Pokemon criado e comercializado pela Nintendo. Você deve criar representações para os Pokémons, uma pokeagenda e os treinadores. Nesse aplicativo você pode capturar pokémons, verificar o poder de ataque deles e comparar seu poder de ataque com o de outros treinadores pokémons. Use a classe de [testes](#) para guiar sua implementação, verificando se o teste após a implementação de cada passo descrito no roteiro.



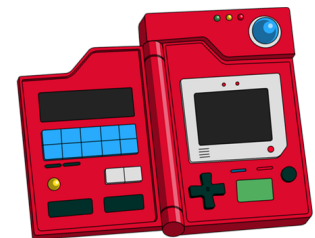
Passo 1: Escreva o código para representar um Pokémon no nosso sistema. Cada Pokémon tem um **nome**, um **nível** (inteiro), **ataque base** (inteiro) e um **tipo** (string), todos definidos ao criar o Pokémon. Podemos **consultar**, bem como **alterar** essas informações em um Pokémon, **exceto pelo nível**, que só pode aumentar de um em um quando ele **ganha um nível**. Dois Pokémons são **iguais** se eles possuem o mesmo nome e nível. Além disso, um Pokémon deve **informar seu poder de ataque** que é a multiplicação entre seu nível e o ataque base. A representação em String de um Pokémon é:

```
<nome> (<tipo>). lvl. <nível>; power: <poder_ataque>
```

Por exemplo:

```
Squirtle (Agua). lvl 15; power: 120  
Pikachu (Eletrico). lvl. 30; power: 300
```

Passo 2: Para organizar seus Pokémons capturados, cada treinador tem uma **Pokeagenda** que armazena e manipula a **coleção de Pokémons** capturados de um treinador. Para cada treinador, a ordem na qual os pokémons foram adicionados importa. Além disso, um treinador pode capturar pokémons repetidos (exemplo, dois Rattatas de nível 5).



Crie métodos para **adicionar** Pokémons na pokeagenda (como um treinador sempre captura um pokémon, não precisa retornar booleanos ou Exceptions nessa funcionalidade), **consultar** se um Pokémon de um nome específico pertence à pokeagenda, indicar **quantos pokémons** foram

capturados, e para retornar qual **o Pokémon mais forte que foi capturado**, ou seja, aquele com o maior poder de ataque (em caso de empate, retorne o último encontrado).

Passo 3: Uma das características mais legais de ser um treinador Pokémon é se amostrar com sua coleção de Pokémons. ;)

Para isso, crie funcionalidades na sua Pokeagenda como **retornar uma lista de pokemons de um tipo específico** que estão cadastrados na pokeagenda. Além disso, como a pokeagenda possui pokémons de diferentes poderes de ataque, crie um método para retornar o **poder de ataque total** da sua coleção de pokémons. Esse poder de ataque total corresponde à soma do poder de ataque de cada Pokémon cadastrado.

Duas pokeagendas **são iguais** se elas possuem a mesma quantidade de pokémons cadastrados e o mesmo poder total. Além disso, faça com que sua pokeagenda possa retornar o seu cadastro de pokémons seguindo o seguinte formato de String.

```
<total_pokemons> pokemons capturados:  
1- <nome_pokemon1> (tipo1). lvl. <nivel_1>; power: <poder_ataque1>  
2- <nome_pokemon2> (tipo2). lvl. <nivel_2>; power: <poder_ataque2>  
3- ...
```

Por exemplo:

```
6 pokemons capturados:  
1 - Squirtle (Agua). lvl 15; power: 120  
2 - Pikachu (Eletrico). lvl. 30; power: 300  
3 - Vileplume (Grama). lvl. 45; power: 675  
4 - Articuno (Gelo). lvl. 50; power: 1000  
5 - Pikachu (Eletrico). lvl. 15; power: 150  
6 - Starmie (Psiquico). lvl. 50; power: 750
```



Passo 4: Chegou o momento de criar o(a) **treinador(a) Pokémon** do nosso aplicativo. Cada treinador(a) Pokémon possui um **nome**, e uma **pokeagenda que inicia vazia**. Um(a) treinador(a) pode capturar pokemons para adiciona-los à sua pokeagenda.

Além disso, o(a) treinador(a) pode usar **todas as funcionalidades de sua pokeagenda** para consultar pokemons. Ou seja, posso verificar em um treinador: quantos pokemons capturou, se capturou um Pokémon de nome específico, que pokémons capturou de acordo com um tipo específico, etc. Além disso, cada treinador pode se apresentar por meio da seguinte string:

```
Ola, sou <nome_treinador>!  
Meu poder total eh <poder_ataque_total>, vai encarar?!"  
<total_pokemons> pokemons capturados:  
1- <nome_pokemon1> (tipo1). lvl. <nivel_1>; power: <poder_ataque1>  
2- <nome_pokemon2> (tipo2). lvl. <nivel_2>; power: <poder_ataque2>  
3- ...
```

Por exemplo:

```
Ola, sou Neto.  
Meu poder total eh 2995, vai encarar?!  
6 pokemons capturados:  
1 - Squirtle (Agua). lvl 15; power: 120  
2 - Pikachu (Eletrico). lvl. 30; power: 300  
3 - Vileplume (Grama). lvl. 45; power: 675  
4 - Articuno (Gelo). lvl. 50; power: 1000  
5 - Pikachu (Eletrico). lvl. 15; power: 150  
6 - Starmie (Psiquico). lvl. 50; power: 750
```

Além disso, dois treinadores pokemons são **iguais** se possuírem o mesmo nome e duas pokeagendas iguais (veja o Passo 3).

Passo 5: Cada treinador deve ser capaz de comparar sua força com a de outro treinador. A força de cada treinador é o poder de ataque total de todos os seus Pokémons capturados. De acordo com a comparação, faça com que o Treinador retorne uma String para cada caso:

Se eu sou mais forte:	"Aew, ganhei! Sou mais forte que <outro_treinador>."
Se somos iguais:	"Eu e <outro_treinador> estamos empatados."
Se sou mais fraco:	"Droga, perdi! <outro_treinador> eh mais forte!"

OBS: Substitua <outro_treinador> apenas pelo **nome do outro treinador**.

Como pontuação extra para a prova implemente as seguintes funcionalidades:

Extra (0,3): Tratamento com Exception para os dados e tipos de pokémons. Não é permitido criar Pokemons com: (i) Nome vazio, ou nulo; (ii) Nivel menor que 1; (iii) Poder de ataque menor que 0 (ainda existe o Magikarp); (iv) Tipos diferentes de Agua, Fogo, Eletrico, Grama, Gelo, Dragao, Fantasma e Psiquico.

Extra (0,7): Mochila e Itens do treinador pokemon. Aumente seu programa para que o seu Treinador possua uma mochila de Item. Um Item possui apenas um **nome** e um **preço** que podem ser consultados. A Mochila pode adicionar itens, e também remover e verificar se ela possui itens a partir do nome do item. Modifique também o Treinador para incluir as funcionalidades de adicionar e remover itens na sua mochila. Por fim, permita que o treinador possa consultar **quanto ele ou ela pode lucrar vendendo os itens da mochila**, ao retornar a soma dos preços de todos os itens presentes na mochila.



Entrega da prova: Instruções disponíveis no [Canvas](#).

Link para o arquivo de testes: <https://goo.gl/FYnzbi>

Nomeie seu projeto usando o seguinte padrão:

- **Nome_Sobrenome_Turma**; Ex: Mateus_Mangueira_03

Importante:

- Seu código será corrigido baseado no design voltado para modularização e reuso de funcionalidades dos objetos; Também vamos considerar (porém com menor peso), os resultados dos testes fornecidos.
- Utilize a padronização de nomes em Java para variáveis, classes, métodos e pacotes;
- Implemente focando em modularização e reuso de seu código;
- Divida bem as responsabilidades de suas classes.
- Para facilitar a criação de Strings com ‘quebra de linha’ use:
 - `String FIM_DE_LINHA = System.lineSeparator();`

Boa prova e boa implementação!