

## Lista de Exercícios 02

Data limite de entrega: 05/06/2022

Atenção: Os códigos deverão ser enviados via GitHub. Criar um repositório em seu GitHub com o nome LCP2022 (caso ainda não exista) e criar uma pasta Lista02, os arquivos .java deverão estar dentro desta pasta. Conforme fizemos com a Lista 01, ao terminar, enviar um email com o título LCP2022 e com os seguintes dados: Nome Completo, RA e o link do seu GitHub para [lucas.guerreiro@unesp.br](mailto:lucas.guerreiro@unesp.br).

- 1) Escreva um programa que receba um nome completo em qualquer tipo de capitalização e devolva o nome completo com apenas a primeira letra de cada nome maiúscula e as demais minúsculas. Além disso, deve-se tratar para que as preposições "da", "de" e "do" fiquem minúsculas. Por exemplo, se a entrada for: "JOÃO CARLOS DA SILVA", a saída será: "João Carlos da Silva". O nome da classe deverá ser *Exercicio01.java*.
- 2) Escreva um programa que irá receber duas strings e irá gerar uma String de saída que será uma mescla entre elas. Por exemplo, se a primeira String for "ABcD" e a segunda string for "DEfg", a saída deverá ser "ADBecFDg". Caso uma String seja maior que a outra, o restante dela deverá ser incluso no final da String final. O nome da classe deverá ser *Exercicio02.java*.
- 3) Palíndromos são palavras que são idênticas quando lidas da esquerda para a direita ou da direita para a esquerda. Escrever um programa para ler uma String e retornar se esta é um palíndromo ou não. Exemplo de palíndromo: *SUBINOONIBUS*. O nome da classe deverá ser *Exercicio03.java*.
- 4) Escreva um programa para jogar "Jogo da Velha". Neste caso, o programa irá fazer uso de uma matriz que irá controlar o jogo e esta deverá ser exibida a cada jogada. Os jogadores jogam em ordem: Jogador 1, Jogador 2, Jogador 1, e assim sucessivamente, a partir de perguntas no console. O Jogador 1 será atribuído com o marcador **X** e o Jogador 2 com o marcador **O**, sendo espaços vazios do jogo demarcados como underscore (\_). Com isso, a cada jogada, o jogador assinala a linha e coluna que deseja jogar e é exibido o jogo atual com as posições marcadas de acordo com todas as jogadas até o momento.

Por exemplo, se na primeira jogada o jogador indicar a posição de linha 0 e de coluna 0, o retorno no console será:

```
X  _  _  
_  _  _  
_  _  _
```

Se na sequência a jogada do jogador 2 for na linha 1 e coluna 1, o retorno no console será:

```
x _ _  
_ o _  
_ _ _
```

Por fim, identificar o vencedor do jogo, caso exista um (uma sequência de 3 símbolos na horizontal, vertical ou diagonal). O nome da classe do programa principal deve ser *JogoVelha.java*.

5) Escreva um programa que irá receber do usuário um número de telefone e irá formatá-lo de acordo com as condições: o número recebido será no formato (XX) XXXXX-XXXX ou (XX) XXXX-XXXX, se for no primeiro formato devemos exibir mensagem de que este é um número de telefone celular e se for no segundo formato exibir que é um telefone residencial. Além disso, devemos exibir uma mensagem: “O DDD é **XX** e o telefone é **XXXXXXXX**” através do método Split (sendo o número do telefone exibido em 8 ou 9 caracteres de acordo com a entrada do usuário). O nome da classe deve ser *Telefones.java*.

6) Utilizando expressões regulares, criar um validador de RG, sendo este formato: PARTE1.PARTE2.PARTE3-DÍGITOVERIFICADOR. Com as seguintes regras:

- PARTE1 é composta por um ou dois dígitos;
- PARTE2 e PARTE3 possuem exatamente 3 dígitos.
- DÍGITOVERIFICADOR será um dígito ou “X”.

Com isso, deverá ser exibido pelo programa se o RG é válido ou não. O programa deverá se chamar *Verifica.java*.

7) Utilizando o conceito de *Collections* (*List*, *LinkedList*, *ArrayList* etc), fazer um programa que irá receber os dados de alunos (nome, RA, idade, curso) e armazenar estes dados em uma lista de objetos de alunos. O programa principal irá então adicionar alunos a lista, e também remover alunos da lista a partir de um RA, caso o RA não exista, apenas exibir mensagem de RA não existente. Alimentar a lista pelo método principal e realizar operações de adição e remoção de alunos conforme a descrição acima. O programa principal deverá se chamar *Exercicio07.java*.