

LP1A3 – Exercícios

Aula 8 – Coleções de Objetos Genéricas - Classes e Métodos

Instruções para entrega das listas de exercícios:

Meio de Entrega: As resoluções das listas de exercícios devem ser entregues exclusivamente por meio do ambiente Moodle (<http://eadcampus.spo.ifsp.edu.br>).

Forma de Entrega: Para exercícios com uma única classe, deve ser entregue o arquivo da classe (extensão JAVA) referente ao exercício. Por exemplo: Exercício3.java. Para exercícios com mais de uma classe, cada exercício deve ter uma pasta, na qual serão colocados os arquivos JAVA referentes ao exercício. Por exemplo: Para o Exercício 4, deve existir uma pasta “Exercicio4” contendo todos os arquivos JAVA deste exercício. **Entregue apenas os arquivos JAVA.** Todos os arquivos da lista devem ser compactados em um único arquivo (extensão RAR ou ZIP), cujo nome deverá conter a aula, o nome e um sobrenome do aluno. Por exemplo: Aula2_JoaoSilva.zip.

Prazo de Entrega: O prazo de entrega está definido na própria página de exercícios do Moodle, lembrando que o sistema bloqueia o envio de arquivos após a data e horário indicados.

Obs.: A resolução deste(s) exercício(s) deve ser feita de forma INDIVIDUAL. Listas de exercícios com uma ou mais respostas idênticas serão desconsideradas integralmente para efeitos de nota de participação.

1. Faça um programa para controlar uma lista de produtos utilizando a classe **ArrayList**. Enquanto o usuário não digitar a opção 0 (zero), o programa deve apresentar as opções a seguir. De acordo com o número da opção informado, o programa deverá efetuar a operação, solicitando as informações necessárias ao usuário (quando necessário). Os métodos referentes a cada operação devem ser implementados em uma classe chamada **Lista**.
 - 1) **Inserir produto** – Solicitar ao usuário o nome do produto.
 - 2) **Inserir produto em posição específica** – Solicitar ao usuário a posição e o nome do produto.
 - 3) **Consultar produto (pelo nome)** – Solicitar ao usuário o nome do produto.
 - 4) **Consultar produto (pela posição)** – Solicitar ao usuário a posição do produto.
 - 5) **Substituir produto** – Solicitar ao usuário a posição do produto a ser substituído e o nome do novo produto.
 - 6) **Remover produto (pelo nome)** – Solicitar ao usuário o nome do produto.
 - 7) **Remover produto (pela posição)** – Solicitar ao usuário a posição do produto.
2. **(Atividade assíncrona referente ao dia 12/09/2020)** Faça um programa para controlar uma fila de pessoas utilizando a classe **LinkedList**. Enquanto o usuário não digitar a opção 0 (zero), o programa deve apresentar as opções a seguir. De acordo com o número da opção informado, o programa deverá efetuar a operação, solicitando as informações necessárias ao usuário (quando necessário). Não se esqueça que em uma fila, o 1º elemento a entrar é o 1º a sair. Os métodos referentes a cada operação devem ser implementados em uma classe chamada **Fila**.
 - 1) **Inserir pessoa** – Solicitar ao usuário o nome da pessoa.
 - 2) **Remover pessoa**
 - 3) **Imprimir fila**
3. **(Atividade assíncrona referente ao dia 12/09/2020)** Faça um programa para controlar uma pilha de livros utilizando a classe **LinkedList**. Enquanto o usuário não digitar a opção 0 (zero), o programa deve apresentar as opções a seguir. De acordo com o número da opção informado, o programa deverá efetuar a operação, solicitando as informações necessárias ao usuário (quando necessário). Não se esqueça que em uma pilha, o último elemento a

entrar é o 1º a sair. Os métodos referentes a cada operação devem ser implementados em uma classe chamada **Pilha**.

- 1) **Inserir livro** – Solicitar ao usuário o título do livro.
- 2) **Remover livro**
- 3) **Imprimir pilha**

4. **(Atividade assíncrona referente ao dia 12/09/2020)** Faça um programa para manipular um conjunto de produtos utilizando a classe **TreeSet**. Enquanto o usuário não digitar a opção 0 (zero), o programa deve apresentar as opções a seguir. De acordo com o número da opção informado, o programa deverá efetuar a operação, solicitando as informações necessárias ao usuário (quando necessário). Os métodos referentes a cada operação devem ser implementados em uma classe chamada **Conjunto**.

- 1) **Inserir produtos no 1º conjunto** – Solicitar ao usuário a quantidade de produtos e, em seguida, os nomes dos produtos.
- 2) **Inserir produtos no 2º conjunto** – Solicitar ao usuário a quantidade de produtos e, em seguida, os nomes dos produtos.
- 3) **Consultar todos produtos dos dois conjuntos (união).**
- 4) **Consultar apenas produtos em comum nos dois conjuntos (intersecção).**
- 5) **Consultar produtos do 1º conjunto que não existem no 2º conjunto (diferença).**
- 6) **Consultar se o 1º conjunto está contido no 2º conjunto (verificação de subconjunto).**