**Curso Superior de Tecnologia em Sistemas para Internet (3º período)**

**COSSI-1014 — Orientação a Objetos**

Unidade temática #1 — A linguagem de programação Java

Sintaxe Java

Roteiro de aula

Prof. Me. Jonas Pontes

=======================================================

1. **Introdução à sintaxe Java 🡪 Data: 15/02**

Objetivo: Apresentar a estrutura básica de um programa Java e os conceitos fundamentais.

Estrutura Básica:

* Declaração de classes e métodos.
* Método main como ponto de entrada.
* I/O em Java

Variáveis e Tipos de Dados:

* Declaração e inicialização de variáveis.
* Tipos primitivos: int, double, char, boolean.
* Tabela de tipos, tamanho do dado (em bits) e intervalo numérico

Exemplo Prático:

* Escrever um programa simples que declare variáveis de diferentes tipos.

Exercícios:

* Criar um programa que calcula e mostre o resultado das quatro operações básicas entre duas variáveis declaradas e inicializada.

=======================================================

1. **Entrada e saída em Java**

Objetivo: Ensinar como interagir com o usuário, receber dados e exibir informações.

Scanner Class:

* Importar e utilizar a classe Scanner para obter entrada do usuário.

System.out.println e System.out.print:

* Mostrar como exibir informações no console.

Exemplos Práticos:

* Criar um programa que solicita dois números ao usuário, realiza a soma e exibe o resultado.

Exercícios:

* Escrever que calcule e mostre o nome de uma pessoa e sua idade em 31 de dezembro de um determinado ano. Para isso, leia o nome da pessoa, seu ano de nascimento e o ano para qual deseje calcular a idade.
* Desenvolver um programa que pede a base e altura de um triângulo e calcula a área.
* Problema Divisão do Tesouro (OBI): https://olimpiada.ic.unicamp.br/pratique/pj/2020/f1/tesouro/.
* Problema Piso da Escola (OBI): https://olimpiada.ic.unicamp.br/pratique/p2/2018/f1/piso/.

=======================================================

1. **Operadores e estruturas de decisão em Java 🡪 16/02**

Objetivo: Explorar os operadores aritméticos e lógicos disponíveis em Java, como sua aplicação em estruturas de decisão.

Operadores Aritméticos:

* +, -, \*, /, %.

Operadores Relacionais:

* ==, !=, <, >, <=, >=.

Operadores Lógicos:

* &&, ||, !.

Estruturas de decisão:

* if, else-if, else
* Switch-Case

Exemplos Práticos:

* Criar um programa que realiza operações aritméticas e lógicas simples.
* Criar um programa que classifica a nota do aluno em A, B, C, D ou E.

Exercícios:

* Implementar um programa que verifica se um número é par ou ímpar.
* Desenvolver um programa que verifica se um ano é bissexto.
* Problema Acelerador de Partículas (OBI): https://olimpiada.ic.unicamp.br/pratique/p2/2020/f1/acelerador/

=======================================================

1. **Estruturas de Repetição em Java – revisão em 22/02**

Objetivo: Explorar as estruturas de repetição (for, while) e seu uso em Java.

Loop for:

* Sintaxe e aplicação em soma de números

Loop while:

* Execução enquanto uma condição é verdadeira. Soma dos números com base nas entradas, mediante uma condição.
* Aprimorar a calculadora. Ela deve permitir ao usuário fazer múltiplas operações.

Exemplos Práticos:

* Criar um programa que calcula a soma e média dos números de 1 a 10 usando um loop.

Exercícios:

* Implementar um programa que verifica se um número é primo.
* Implemente um programa que calcule o MDC e o MMC entre dois números inteiros.
* Escreva um programa em Java que leia uma quantidade indefinida de valores reais, que representam as massas dos bois de uma fazenda, e apresente o boi mais pesado, mais leve e a média dos valores lidos. 0 ou número negativo são as condições de parada.
* Exercícios de fixação do capítulo 3. Variáveis Primitivas e Controle de Fluxo da apostila Java e Orientação a Objetos. Link: https://www.alura.com.br/apostila-java-orientacao-objetos/variaveis-primitivas-e-controle-de-fluxo.

1. **Arranjos em Java 🡪 22/02**

Objetivo: Aprender sobre vetores, matrizes, listas e mapas em Java.

Vetores e matrizes:

* Declaração;
* Inserção e modificação de item do arranjo;
* Tamanho;
* String e método, length e split.

Listas:

* Criação e aplicação de ArrayList;
* Acesso e modificação de itens;
* Métodos remove, clear, size, add;
* For-each em listas;
* Ordenação.

Mapas:

* Criação e aplicação de HashMap;
* Métodos put, get, remove, clear, size;
* Iteração por valor e chave

Exemplos Práticos:

* Criar um programa que leia o nome completo de uma pessoa e mostre o primeiro e o último nome;
* Ler um vetor e uma matriz e encontrar as quantidades de elementos tem na matriz de cada elemento do vetor;
* Definir uma lista que guarde um conjunto de frutas. Depois, mostre as frutas que foram armazenadas. Fazer inclusão, exclusão e atualização.
* Criar um mapa com o nome dos alunos da turma com suas respetivas idades. Mostrar esse mapa e fazer operações básicas.

Exercícios:

* Crie um programa que solicita ao usuário uma frase e conta o número de palavras na frase.
* Desenvolva um programa que cria uma lista de números inteiros, permite a adição de novos números, remove um número específico e, ao final, exibe a lista ordenada.
* Implemente um programa que seja um dicionário de termos relacionados à informática, armazenando palavras e suas respectivas definições em um HashMap. Permita a busca e remoção de palavras.