Resumo: Entrada e Saída de Dados em Java

# 1. O que é Entrada e Saída de Dados em Java?

Entrada e saída de dados (E/S) é a forma de comunicação entre o usuário e o programa.  
- Entrada: o programa recebe dados do usuário (como nome, idade, etc).  
- Saída: o programa exibe resultados ou grava em arquivos.  
  
Essas operações são essenciais para criar programas interativos e úteis.

# 2. Entrada de Dados

Ferramentas principais:  
1. Scanner (mais usado para iniciantes)  
2. BufferedReader + InputStreamReader (mais eficiente e usado com arquivos também)  
  
Scanner:  
Classe prática para ler dados do teclado. Permite ler:  
- Strings (nextLine())  
- Números inteiros (nextInt())  
- Números decimais (nextDouble())  
  
Exemplo:  
Scanner scanner = new Scanner(System.in);  
System.out.print("Digite seu nome: ");  
String nome = scanner.nextLine();  
System.out.print("Digite sua idade: ");  
int idade = scanner.nextInt();  
System.out.println("Olá, " + nome + ". Você tem " + idade + " anos.");  
scanner.close();  
  
BufferedReader + InputStreamReader:  
Mais indicado para leitura eficiente de texto (ideal em arquivos grandes).  
  
Exemplo:  
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));  
System.out.print("Digite seu nome: ");  
String nome = reader.readLine();  
System.out.print("Digite sua idade: ");  
int idade = Integer.parseInt(reader.readLine());

# 3. Saída de Dados

System.out:  
Forma mais comum de exibir dados no console:  
- print() → escreve na mesma linha  
- println() → escreve e pula linha  
  
Exemplo:  
System.out.println("Hello, World!");  
  
PrintWriter:  
Permite formatação mais avançada na saída de dados. Pode imprimir no console ou escrever em arquivos.  
  
Exemplo:  
PrintWriter writer = new PrintWriter(System.out, true);  
writer.println("Hello, World!");  
writer.printf("Nome: %s, Idade: %d%n", "João", 30);

# 4. Manipulação de Arquivos em Java

Permite armazenar dados permanentemente (em texto ou binário).  
Você pode: Criar arquivos, Escrever/ler conteúdo, Editar e salvar informações  
  
Escrever em arquivo com BufferedWriter:  
BufferedWriter writer = new BufferedWriter(new FileWriter("arquivo.txt"));  
writer.write("Olá, Mundo!");  
writer.newLine();  
writer.write("Segunda linha.");  
writer.close();  
  
Ler de arquivo com BufferedReader:  
BufferedReader reader = new BufferedReader(new FileReader("arquivo.txt"));  
String linha;  
while ((linha = reader.readLine()) != null) {  
 System.out.println(linha);  
}  
reader.close();

# Resumo Geral

Tema | Ferramenta | Função Principal  
---------------------|------------------------|------------------------------  
Entrada de Dados | Scanner | Leitura fácil do console  
 | BufferedReader | Leitura eficiente  
Saída de Dados | System.out | Mostrar mensagens no console  
 | PrintWriter | Impressão formatada  
Manipular Arquivos | BufferedReader/Writer | Ler e gravar arquivos

# Conteúdo Bônus (YouTube)

Canal: Cryswerton Silva  
Vídeo no YouTube sobre entrada e saída de dados em Java para iniciantes.