

1. Crie um projeto novo chamado `teste-io`. E, nele, crie um programa (simplesmente uma classe com um `main`) que leia da entrada padrão. Para isso, você vai precisar de um `BufferedReader` que leia do `System.in`, conforme programa abaixo:

```
public class TestaEntrada {
    public static void main(String[] args) throws
        IOException {
        InputStream is = System.in;
        InputStreamReader isr = new InputStreamReader(is);
        BufferedReader br = new BufferedReader(isr);
        String s = br.readLine();

        while (s != null) {
            System.out.println(s);
            s = br.readLine();
        }
    }
}
```

TesteEntrada.java

2. Com base no programa anterior, faça um programa para ler as linhas de um arquivo. Para isso, você vai precisar de um `FileInputStream`.
3. Seu programa lê todas as linhas desse arquivo. Repare na utilização do polimorfismo. Como ambos são `InputStream`, isso faz com que eles se encaixem no `InputStreamReader`. Além da `FileInputStream`, que outras classes poderiam tomar seu lugar? Olhe na documentação!
4. Utilize a classe `Scanner` do Java 5 para ler de um arquivo e colocar na tela. O código vai ficar incrivelmente pequeno.

```
public class EntradaDeUmArquivo {
    public static void main(String[] args) throws
        IOException {
        InputStream is = new FileInputStream("arquivo.txt");
        Scanner entrada = new Scanner(is);

        System.out.println("Digite sua mensagem:");
        while (entrada.hasNextLine()) {
            System.out.println(entrada.nextLine());
        }
        entrada.close();
    }
}
```

```

    }
}

```

EntradaDeUmArquivo.java

Depois troque a variável `is` para que ela se refira ao `System.in`. Agora você está lendo do teclado!

5. Altere seu programa para que ele leia do arquivo e, em vez de jogar na tela, jogue em um outro arquivo. Você vai precisar, além do código anterior para ler de um arquivo, do código para escrever em um arquivo. Para isso, você pode usar o `BufferedWriter` ou o `PrintStream`. Este último é de mais fácil manipulação.

Se for usar o `BufferedWriter`, fazemos assim para abri-lo:

```

OutputStream os = new FileOutputStream("saida.txt");
OutputStreamWriter osw = new OutputStreamWriter(os);
BufferedWriter bw = new BufferedWriter(osw);

```

Escrita.java

Dentro do loop de leitura do teclado, você deve usar `bw.write(x)`, onde `x` é a linha que você leu. Use `bw.newLine()` para pular de linha. Não se esqueça de, no término do loop, dar um `bw.close()`. Você pode seguir o modelo:

```

while (entrada.hasNextLine()) {
    String linha = entrada.nextLine();
    bw.write(linha);
    bw.newLine();
}
bw.close();

```

Escrita2.java

Após rodar seu programa, dê um refresh no seu projeto (clique da direita no nome do projeto, refresh) e veja que ele criou um arquivo `saida.txt` no diretório.

6. Altere novamente o programa para ele virar um pequeno editor: lê do teclado e escreve em arquivo. Repare que a mudança a ser feita é mínima!
7. Faça um programa para escrever registros de contas bancárias escritos num arquivo (`“contas.txt”`). Os campos (número da conta, saldo e titular) são separados por `“;”`. Cada registro ocupa uma linha no arquivo. Os dados de cada conta são lidos via teclado.
8. Faça um programa para ler registros de contas bancárias escritos arquivo `“contas.txt”` (até 100 contas). Os campos (número da conta, saldo e titular) são separados por `“;”`. Cada registro ocupa uma linha no arquivo. As contas lidas devem ser armazenadas num vetor.

Divirta-se!