

 07

Mão na massa: Leitura de arquivo

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

- 1) No Eclipse, crie um novo projeto, chamado **java-io**. Dentro deste projeto, crie a classe **TesteLeitura**, no pacote **br.com.alura.java.io.teste**, com o método **main**:

```
public class TesteLeitura {  
  
    public static void main(String[] args) {  
  
    }  
  
}
```

- 2) Baixe [aqui \(https://s3.amazonaws.com/caelum-online-public/857-java-io/01/lorem.zip\)](https://s3.amazonaws.com/caelum-online-public/857-java-io/01/lorem.zip) o zip com o arquivo **lorem.txt** e copie-o para a **raiz** do projeto **java-io**.

- 3) Para estabelecer um fluxo de entrada com arquivo, instancie um **FileInputStream**. Utilize o construtor que recebe o caminho do arquivo como parâmetro:

```
public class TesteLeitura {  
  
    public static void main(String[] args) {  
  
        FileInputStream fis = new FileInputStream("lorem.txt");  
    }  
  
}
```

- 4) Trate a exceção, adicionando um **throws**:

```
public class TesteLeitura {  
  
    public static void main(String[] args) throws FileNotFoundException {  
  
        FileInputStream fis = new FileInputStream("lorem.txt");  
    }  
  
}
```

- 5) Como o **FileInputStream** somente lê os bytes, instancie a classe **InputStreamReader** para transformar esses bytes em caracteres. Ela recebe um **InputStream** como parâmetro no construtor:

```
public class TesteLeitura {  
  
    public static void main(String[] args) throws FileNotFoundException {
```

```

        FileInputStream fis = new FileInputStream("lorem.txt");
        InputStreamReader isr = new InputStreamReader(fis);
    }

}

```

- 6) Em seguida, instancie a classe `BufferedReader` para poder ler a linha inteira do arquivo. Ela recebe um `Reader` como parâmetro no construtor:

```

public class TesteLeitura {

    public static void main(String[] args) throws FileNotFoundException {

        FileInputStream fis = new FileInputStream("lorem.txt");
        InputStreamReader isr = new InputStreamReader(fis);
        BufferedReader br = new BufferedReader(isr);
    }

}

```

- 7) Através do método `readLine`, do `BufferedReader`, leia a linha do arquivo e guarde-a em uma variável. Você precisará tratar uma exceção novamente, então, no `throws`, troque a exceção `FileNotFoundException` pela exceção `IOException`:

```

public class TesteLeitura {

    public static void main(String[] args) throws IOException {

        FileInputStream fis = new FileInputStream("lorem.txt");
        InputStreamReader isr = new InputStreamReader(fis);
        BufferedReader br = new BufferedReader(isr);

        String linha = br.readLine();
    }

}

```

- 8) O método `readLine` devolve `null` quando não existe mais nenhum conteúdo a ser lido. Então, enquanto ainda houver conteúdo, ou seja, enquanto a linha não for nula, imprima-a e continue lendo o arquivo:

```

public class TesteLeitura {

    public static void main(String[] args) throws IOException {

        FileInputStream fis = new FileInputStream("lorem.txt");
        InputStreamReader isr = new InputStreamReader(fis);
        BufferedReader br = new BufferedReader(isr);

        String linha = br.readLine();

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

}

```

```

        br.close();
    }

}

```

9) Feche o `BufferedReader`, através do método `close`:

```

public class TesteLeitura {

    public static void main(String[] args) throws IOException {

        FileInputStream fis = new FileInputStream("lorem.txt");
        InputStreamReader isr = new InputStreamReader(fis);
        BufferedReader br = new BufferedReader(isr);

        String linha = br.readLine();

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

        br.close();
    }

}

```

10) Por fim, saiba que o `FileInputStream` é um `InputStream` e o `InputStreamReader` é um `Reader`. Então, você pode utilizar esses tipos mais genéricos:

```

public class TesteLeitura {

    public static void main(String[] args) throws IOException {

        InputStream fis = new FileInputStream("lorem.txt");
        Reader isr = new InputStreamReader(fis);
        BufferedReader br = new BufferedReader(isr);

        String linha = br.readLine();

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

        br.close();
    }

}

```

O `BufferedReader` também é um `Reader`, mas esse tipo não possui o método `readLine`.

