

# Aula 6

# Arquivos – Leitura e Escrita

Existem classes que fazem esse trabalho:

- `BufferedReader`, `FileReader` e `BufferedWriter` e `FileWriter`;
- Fazem a leitura do arquivo, linha a linha;
- O que define a quebra de linha?
- Após lida basta manipular a linha com os recursos da `String`;
- Algumas regras importantes:
  - Passar o caminho do arquivo corretamente;
  - Fechar o arquivo no final da leitura ou da escrita.

# Arquivos - Leitura - package IO

```
public class LeituraEscrita {  
    static final String PATH_BASICO = "./temp/";  
  
    static final String EXTENSAO = ".txt";  
  
    public static void leitor(String path) throws IOException {  
        BufferedReader buffRead = new BufferedReader(new FileReader(PATH_BASICO+path));  
        String linha = "";  
        while (true) {  
            linha = buffRead.readLine();  
            if (linha != null) {  
                System.out.println(linha);  
            }  
            else {  
                break;  
            }  
        }  
        buffRead.close();  
    }  
}
```

# Arquivos - Leitura - try/catch

//CÓDIGO OMITIDO - SLIDE ANTERIOR

```
try {  
    while (true) {  
        linha = buffRead.readLine();  
        if (linha != null) {  
            System.out.println(linha);  
        }  
        else {  
            break;  
        }  
    }  
    buffRead.close();  
} catch (FileNotFoundException e) {  
    System.out.println("Arquivo não encontrado no caminho " + PATH_BASICO + EXTENSAO);  
}
```

# Arquivos - Escrita

```
public static void escritor(String path) throws IOException {
```

```
    Scanner sc = new Scanner(System.in);  
    System.out.println("Escreva o nome do arquivo: ");  
    String nome = sc.next();
```

```
    BufferedWriter buffWrite = new BufferedWriter(new FileWriter(PATH_BASICO + nome + path));  
    String linha = "";
```

```
    System.out.println("Escreva algo: ");  
    linha = sc.nextLine();  
    buffWrite.append(linha + "\n");  
    buffWrite.close();
```

```
}
```