

# Leitura e Escrita em Arquivo

# Leitura de Arquivo

- Para leitura de arquivos existem diversas classes que podem ser utilizadas. Algumas inclusive lêem objetos em arquivos de forma criptografada ( com caracteres diferentes do usual ).
- Nessa aula, utilizaremos a forma mais parecida possível com o que foi aprendido em C. Para tanto serão trabalhadas as classes `FileInputStream` e `Scanner`. A classe `Scanner` pode ser utilizada inclusive para leitura via teclado.

# Leitura de Arquivo – Ex:

- **O seguinte trecho de código faz a leitura de um arquivo `arq.txt` que contem dois números inteiros em sequência e um número real em seguida.**

...

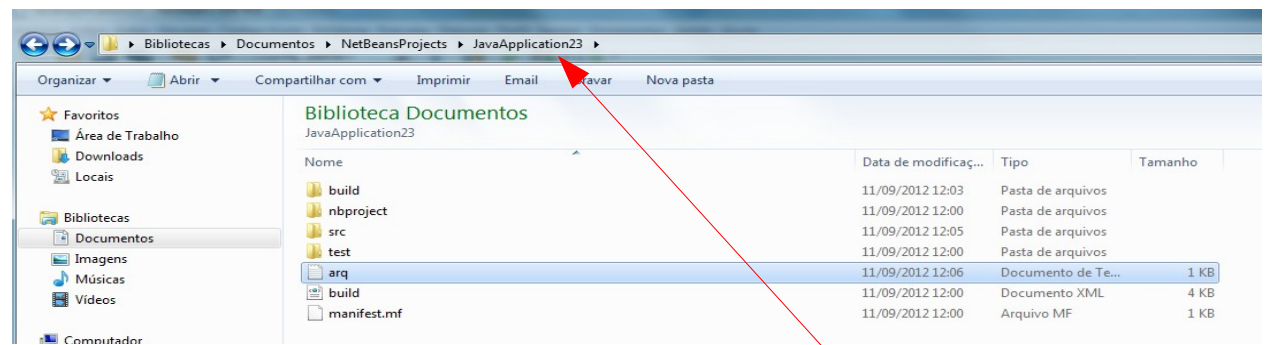
```
1 FileInputStream fil = new
  FileInputStream("c:/arq.txt");
2 Scanner scan = new Scanner(fil);
3 int num2 = scan.nextInt();
4 int num1 = scan.nextInt();
5 double num3 = scan.nextDouble();
6 System.out.print(num2+num1);
7 System.out.println(num3);
8 scan.close();
```

...

- **Análise:**

- Na linha 1 é alocado um objeto do tipo `FileInputStream`.
- Na linha 2 é feita a ligação de um objeto `Scanner` ao objeto arquivo.
- Nas linhas 3, 4, 5 é feita a leitura do arquivo.
- Nas linhas 6 e 7 são mostrados os valores lidos na tela.
- Na linha 8 o scanner é fechado.

# Exemplo



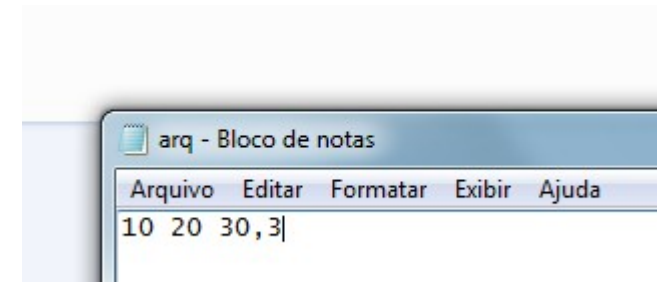
```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        try {
            FileInputStream fil = new FileInputStream("arq.txt");
            Scanner scan = new Scanner(fil);

            int num2 = scan.nextInt();
            int num1 = scan.nextInt();
            double num3 = scan.nextDouble();
            System.out.println(num2 + num1);
            System.out.println(num3);
            scan.close();
        } catch (FileNotFoundException erro) {
            System.out.println("Arquivo não encontrado");
        }
    }
}
```

- Nesse exemplo, arq.txt se encontra na raiz do projeto.
- Números reais devem ser separados por , no arquivo.



# Escrita em Arquivo

- Assim como para leitura de arquivos existem diversas classes que podem ser utilizadas na escrita de arquivos. Algumas inclusive escrevem objetos em arquivos de forma criptografada ( com caracteres diferentes do usual ).
- Nessa aula, novamente, utilizaremos a forma mais parecida possível com o que foi aprendido em C. Para tanto será utilizada a classe Formatter.

# Escrita em Arquivo – Ex:

- **O seguinte trecho de código faz a escrita no arquivo arq.txt**

...

```
1 Formatter formater = new Formatter("c:/arq.txt");
```

```
2 int i = 10;
```

```
3 formater.format("Valor = %d\n", i);
```

```
4 formater.format("Texto no final de tudo");
```

```
5 formater.close();
```

...

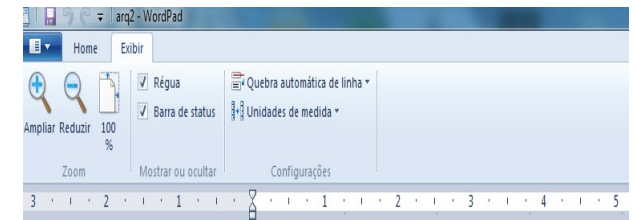
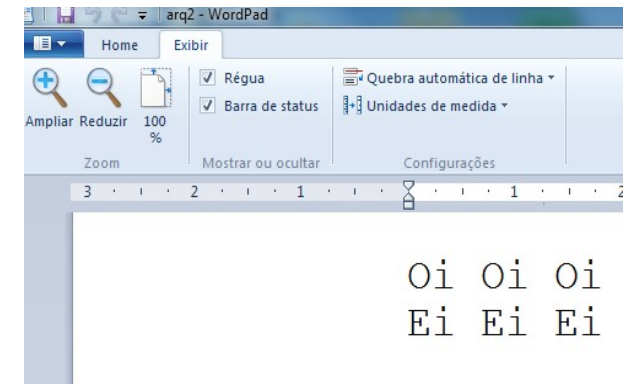
- Análise:

- Na linha 1 o objeto da classe `Formatter` que conhece o arquivo `c:/arq.txt` é criado.
- Na linha 3 é impresso o valor de `i` no arquivo.
- Na linha 4 é impressa uma mensagem no arquivo.
- Na linha 5 o “formatador” é fechado e é terminada a escrita no arquivo.

# Acrescentando valores no final do arquivo

```
package javaapplication23;
import java.util.Scanner;
import java.util.Formatter;
import java.io.*;
public class UsaArquivo {
    public static void main(String[] args) throws FileNotFoundException, IOException
    {
        FileInputStream fil = new FileInputStream("arq2.txt");
        Scanner scan = new Scanner(fil);
        // Lendo o que havia no arquivo e colocando em s
        String s = "";
        while (scan.hasNext())
        {
            s = s + scan.nextLine() + "\n";
        }
        scan.close();

        Formatter formater = new Formatter("arq2.txt");
        formater.format(s);
        formater.format("Texto no final de tudo");
        formater.close();
    }
}
```



# Dúvidas?





# Exercício

- Faça um programa que leia notas em um arquivo e imprima na tela:
  - A maior nota
  - A menor nota
  - A média das notas