

07

Mão na massa: Scanner

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

Conhecendo o Scanner

1) Baixe [aqui](https://caelum-online-public.s3.amazonaws.com/857-java-io/04/contas.csv) (<https://caelum-online-public.s3.amazonaws.com/857-java-io/04/contas.csv>) o arquivo **contas.csv** e copie-o para a **raiz** do projeto **java-io**.

2) No projeto **java-io**, crie a classe **TesteLeitura2**, no pacote **br.com.alura.java.io.teste**, com o método **main**:

```
public class TesteLeitura2 {  
  
    public static void main(String[] args) {  
  
    }  
  
}
```

3) Instancie um **Scanner**. Utilize o construtor que recebe um **File** como parâmetro:

```
public class TesteLeitura2 {  
  
    public static void main(String[] args) {  
  
        Scanner scanner = new Scanner(new File("contas.csv"));  
    }  
  
}
```

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

```
public class TesteLeitura2 {  
  
    public static void main(String[] args) throws Exception {  
  
        Scanner scanner = new Scanner(new File("contas.csv"));  
    }  
  
}
```

5) Através do método **nextLine**, do **Scanner**, leia a linha do arquivo, guarde-a em uma variável e imprima-a:

```
public class TesteLeitura2 {  
  
    public static void main(String[] args) throws Exception {
```

```

Scanner scanner = new Scanner(new File("contas.csv"));

String linha = scanner.nextLine();
System.out.println(linha);
}

}

```

6) O `Scanner` possui o método `hasNextLine`, que devolve `false` quando não existe mais nenhum conteúdo a ser lido. Então, enquanto ainda houver conteúdo, ou seja, enquanto o método `hasNextLine` retornar `true`, leia a linha e imprima-a:

```

public class TesteLeitura2 {

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

        Scanner scanner = new Scanner(new File("contas.csv"));

        while (scanner.hasNextLine()) {
            String linha = scanner.nextLine();
            System.out.println(linha);
        }
    }
}

```

7) Em seguida, feche o `Scanner`:

```

public class TesteLeitura2 {

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

        Scanner scanner = new Scanner(new File("contas.csv"));

        while (scanner.hasNextLine()) {
            String linha = scanner.nextLine();
            System.out.println(linha);
        }

        scanner.close();
    }
}

```

Separando os valores

8) Para separar os valores da linha, crie um novo `Scanner`. Ele receberá como entrada a própria linha:

```

public class TesteLeitura2 {

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

        Scanner scanner = new Scanner(new File("contas.csv"));

```

```

while (scanner.hasNextLine()) {
    String linha = scanner.nextLine();
    System.out.println(linha);

    Scanner linhaScanner = new Scanner(linha);
}

scanner.close();
}
}

```

9) Separe os valores, utilizando um **delimitador**, através do método `useDelimiter`. No caso, o delimitador será a vírgula;

```

public class TesteLeitura2 {

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

        Scanner scanner = new Scanner(new File("contas.csv"));

        while (scanner.hasNextLine()) {
            String linha = scanner.nextLine();
            System.out.println(linha);

            Scanner linhaScanner = new Scanner(linha);
            linhaScanner.useDelimiter(",");
        }

        scanner.close();
    }
}

```

10) Guarde os valores da linha, da conta, em variáveis específicas. Com o método `next` você consegue acessar o próximo valor:

```

public class TesteLeitura2 {

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

        Scanner scanner = new Scanner(new File("contas.csv"));

        while (scanner.hasNextLine()) {
            String linha = scanner.nextLine();
            System.out.println(linha);

            Scanner linhaScanner = new Scanner(linha);
            linhaScanner.useDelimiter(",");

            String tipoConta = linhaScanner.next();
            String agencia = linhaScanner.next();
            String numero = linhaScanner.next();
            String titular = linhaScanner.next();
        }
    }
}

```

```

        String saldo = linhaScanner.next();
    }

    scanner.close();
}

}

```

11) Como `agencia` e `numero` são **números inteiros**, então utilize o método `nextInt` (não esqueça de modificar o tipo da variável também). A mesma coisa com o `saldo`, que é um `double`, então utilize o método `nextDouble`:

```

public class TesteLeitura2 {

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

        Scanner scanner = new Scanner(new File("contas.csv"));

        while (scanner.hasNextLine()) {
            String linha = scanner.nextLine();
            System.out.println(linha);

            Scanner linhaScanner = new Scanner(linha);
            linhaScanner.useDelimiter(",");

            String tipoConta = linhaScanner.next();
            int agencia = linhaScanner.nextInt();
            int numero = linhaScanner.nextInt();
            String titular = linhaScanner.next();
            double saldo = linhaScanner.nextDouble();

        }

        scanner.close();
    }
}

```

12) Em seguida, imprima os valores e feche o `Scanner`. Além disso, para o `Scanner` saber os números com casas decimais utilizam o ponto como delimitador, faça ele seguir o padrão americano, através do método `useLocale`:

```

public class TesteLeitura2 {

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

        Scanner scanner = new Scanner(new File("contas.csv"));

        while (scanner.hasNextLine()) {
            String linha = scanner.nextLine();
            System.out.println(linha);

            Scanner linhaScanner = new Scanner(linha);
            linhaScanner.useLocale(Locale.US);
            linhaScanner.useDelimiter(",");

            String tipoConta = linhaScanner.next();
            int agencia = linhaScanner.nextInt();

```

```

    ... código - scanner.close();
    int numero = linhaScanner.nextInt();
    String titular = linhaScanner.next();
    double saldo = linhaScanner.nextDouble();

    System.out.println(valor1 + valor2 + valor3 + valor4 + valor5);

    linhaScanner.close();
}

scanner.close();
}
}

```

13) Agora, remova a impressão da linha e a impressão dos valores da conta. Além disso, formate os valores da conta, através do `String.format`, para seguir o modelo: "CC - 22-33, Nico Steppat: 210,1". Guarde essa string formatada em uma variável e imprima-a:

```

public class TesteLeitura2 {

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

        Scanner scanner = new Scanner(new File("contas.csv"));

        while (scanner.hasNextLine()) {
            String linha = scanner.nextLine();

            Scanner linhaScanner = new Scanner(linha);
            linhaScanner.useLocale(Locale.US);
            linhaScanner.useDelimiter(",");

            String tipoConta = linhaScanner.next();
            int agencia = linhaScanner.nextInt();
            int numero = linhaScanner.nextInt();
            String titular = linhaScanner.next();
            double saldo = linhaScanner.nextDouble();

            String valorFormatado = String.format("%s - %d-%d, %s: %f", tipoConta, agencia, numero,
                System.out.println(valorFormatado);

            linhaScanner.close();
        }

        scanner.close();
    }
}

```

14) Formate mais ainda a string, para que a agência tenha 4 caracteres no total, mas se tiver menos, que o restante seja preenchido com zeros. Faça a mesma coisa para o número da conta, mas para que tenha 8 caracteres no total, e se tiver menos, que o restante seja preenchido com zeros. Para o nome do titular, que ele tenha 20 caracteres e para o saldo da conta, que ele possua 8 caracteres, sendo os 3 últimos o ponto e duas casas decimais:

```

public class TesteLeitura2 {

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

        Scanner scanner = new Scanner(new File("contas.csv"));

        while (scanner.hasNextLine()) {
            String linha = scanner.nextLine();

            Scanner linhaScanner = new Scanner(linha);
            linhaScanner.useLocale(Locale.US);
            linhaScanner.useDelimiter(",");

            String tipoConta = linhaScanner.next();
            int agencia = linhaScanner.nextInt();
            int numero = linhaScanner.nextInt();
            String titular = linhaScanner.next();
            double saldo = linhaScanner.nextDouble();

            String valorFormatado = String.format("%s - %04d-%08d, %20s: %08.2f", tipoConta,
                agencia, numero, titular, saldo);

            System.out.println(valorFormatado);

            linhaScanner.close();
        }

        scanner.close();
    }
}

```

15) Por fim, para imprimir o saldo com uma vírgula como separador entre o valor inteiro e o valor decimal, passe um `Locale` do Brasil como primeiro parâmetro. Como não há um existente para o Brasil, crie-o. Você também pode fazer o `format` diretamente no `System.out`, mas deve lembrar de pular uma linha ao final:

```

public class TesteLeitura2 {

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

        Scanner scanner = new Scanner(new File("contas.csv"));

        while (scanner.hasNextLine()) {
            String linha = scanner.nextLine();

            Scanner linhaScanner = new Scanner(linha);
            linhaScanner.useLocale(Locale.US);
            linhaScanner.useDelimiter(",");

            String tipoConta = linhaScanner.next();
            int agencia = linhaScanner.nextInt();
            int numero = linhaScanner.nextInt();
            String titular = linhaScanner.next();
            double saldo = linhaScanner.nextDouble();

            System.out.format(new Locale("pt", "BR"), "%s - %04d-%08d, %20s: %08.2f%n",

```

```
    tipoConta, agencia, numero, titular, saldo);  
  
    linhaScanner.close();  
}  
  
scanner.close();  
}  
  
}
```