

devsuperior / aula002 Público

16 stars 7 forks

Star

Watch

Código

Problemas

Solicitações de pull

Ações

Projetos

Wiki

Segurança

Percepções

mestre

...

devsuperior ...

on 25 Jun 2020

[Ver código](#)

Como ler arquivo texto em Java (aplicação real) - Aulão #002

DevSuperior - sua carreira dev com fundamento de ensino superior



/devsuperior



/devsuperior.fb



/devsuperior.ig

Assista o vídeo:



Sumário

- [O que você vai aprender](#)
- [Pré-requisitos](#)
- [Conteúdo do arquivo CSV](#)
- [Código fonte em etapas](#)
 - [Etapa 1: Leitura simples com File e Scanner](#)
 - [Etapa 2: FileReader e BufferedReader com controle manual](#)
 - [Etapa 3: Usando o bloco 'experimental com recursos'](#)
 - [Etapa 4: lendo o arquivo CSV e instanciando uma lista de objetos](#)

O que você vai aprender

- Arquivo de Classes, Scanner, IOException
- Classes FileReader, BufferedReader
- Bloquear try-catch-finally para controle manual dos recursos
- Bloco "try with resources" para controle automatizado dos recursos
- Como ler um arquivo .CSV e transformá-lo em uma lista de objetos em memória

Pré-requisitos

- Lógica de programação
 - Variáveis, entrada, processamento, saída
 - Estrutura condicional
 - Estruturas repetitivas
- OOP básico
 - Classes, atributos, métodos, objetos
 - Construtores, encapsulamento
 - Coleções (Lista)

Conteúdo do arquivo CSV

```
Name,Price,Quantity
Notebook Gamer,2100.90,10
Smartphone X,1890.00,23
TV LED 70,3500.89,8
```

```
Name;Price;Quantity
Notebook Gamer;2100,90;10
Smartphone X;1890,00;23
TV LED 70;3500,89;8
```

Código fonte em etapas

Etapa 1: Leitura simples com File e Scanner

```
aplicativo de pacote ;

importar java . io . Arquivo ;
importar java . io . IOException ;
importar java . útil . Scanner ;

Programa de classe pública {

    public static void main ( String [] args ) {

        Arquivo arquivo = new Arquivo ( "C:\\temp\\in.txt" );
        Scanner sc = null ;
        tente {
            sc = new Scanner ( arquivo );
            while ( sc . hasNextLine () ) {
                System . fora . println ( sc . nextLine () );
            }
        }
        catch ( IOException e ) {
            System . fora . println ( "Erro: " + e . getMessage () );
        }
        finalmente {
            if ( sc != null ) {
                sc . fechar ();
            }
        }
    }
}
```

Etapa 2: FileReader e BufferedReader com controle manual

```
aplicativo de pacote ;

importar java . io . Leitor Buffer ;
importar java . io . Leitor de Arquivos ;
importar java . io . IOException ;

Programa de classe pública {

    public static void main ( String [] args ) {

        String path = "C:\\temp\\in.txt" ;
        BufferedReader br = null ;
```

```

        FileReader fr = null ;

        tente {
            fr = new FileReader ( caminho );
            br = new BufferedReader ( fr );

            String linha = br . leiaLinha ();

            while ( line != null ) {
                System . fora . println ( linha );
                linha = br . leiaLinha ();
            }
        }
        catch ( IOException e ) {
            System . fora . println ( "Erro: " + e . getMessage () );
        }
        finalmente {
            tente {
                if ( br != null )
                    br . fechar ();
                if ( fr != null )
                    fr . fechar ();
            } catch ( IOException e ) {
                e . printStackTrace ();
            }
        }
    }
}

```

Etapa 3: Usando o bloco 'experimentalizar com recursos'

```

aplicativo de pacote ;

importar java . io . Leitor Buffer ;
importar java . io . Leitor de Arquivos ;
importar java . io . IOException ;

Programa de classe pública {

    public static void main ( String [] args ) {

        String path = "C:\\temp\\in.txt" ;

        try ( BufferedReader br = new BufferedReader ( new FileReader ( ca

            String linha = br . leiaLinha ();
            while ( line != null ) {
                System . fora . println ( linha );
                linha = br . leiaLinha ();
            }
        }
    }
}

```

```

    }
    catch ( IOException e ) {
        System . fora . println ( "Erro: " + e . getMessage () );
    }
}
}

```

Etapa 4: lendo o arquivo CSV e instanciando uma lista de objetos

```

aplicativo de pacote ;

importar java . io . Leitor Buffer ;
importar java . io . Leitor de Arquivos ;
importar java . io . IOException ;
importar java . util . ListaMatriz ;
importar java . util . Lista ;

entidades importadoras . Produto ;

Programa de classe pública {

    public static void main ( String [] args ) {

        String path = "c:\\temp\\in.txt" ;

        Lista < Produto > lista = new ArrayList < Produto > ();

        try ( BufferedReader br = new BufferedReader ( new FileReader ( ca

            String linha = br . leiaLinha ();
            linha = br . leiaLinha ();
            while ( linha != null ) {

                String [] vect = linha . divisão ( "," );
                Nome da string = vect [ 0 ];
                Preço duplo = Duplo . parseDouble ( vect [ 1 ] );
                Inteiro qte = Inteiro . parseInt ( vect [ 2 ] );

                Produto prod = novo Produto ( nome , preço , qte );
                lista . adicionar ( prod );

                linha = br . leiaLinha ();
            }

            Sistema . fora . println ( "PRODUTOS:" );
            for ( Produto p : lista ) {
                System . fora . println ( p );
            }
        }
    }
}

```

```

        catch ( IOException e ) {
            System . fora . println ( "Erro: " + e . getMessage () );
        }
    }
}

```

entidades de pacote ;



```

importar java . io . Serializavel ;

```

```

public class Produto implementa Serializable {
    private static final long serialVersionUID = 1L ;

```

```

    nome da string privada ;
    preço duplo privado ;
    private Quantidade inteira ;

```

```

    Produto público () {
    }

```

```

    public Product ( String name , Double price , Integer quantidade ) {
        super () ;
        isso . nome = nome ;
        isso . preço = preço ;
        isso . quantidade = quantidade ;
    }

```

```

    public String getNome () {
        return nome ;
    }

```

```

    public void setName ( String name ) {
        this . nome = nome ;
    }

```

☰ README.md

```

        return price ;
    }

```

```

    public void setPrice ( Double price ) {
        this . preço = preço ;
    }

```

```

    public Integer getQuantity () {
        return quantidade ;
    }

```

```

    public void setQuantity ( quantidade inteira ) {
        this . quantidade = quantidade ;
    }

```

```
@Override
public String toString () {
    return "Produto [nome=" + nome + ", preço=" + preço + ", quantidade"
}
}
```

Lançamentos

Nenhum lançamento publicado

Pacotes

Nenhum pacote publicado

Contribuintes 2



acenélío Nélío Alves



devsuperior DevSuperior

línguas

● Java 100,0%