

Entrada e Saída em Java

1

Java.io

- Java.io é a biblioteca que contém todas as classes que manipulam as entradas e saídas dos programas escritos em Java;
- Ela utiliza **fluxos de entrada e de saída** para realizar as operações de **leitura** e **escrita**, sejam elas em um **arquivo**, uma **conexão** remota via sockets ou na **entrada e saída padrão** de um programa.

2

2

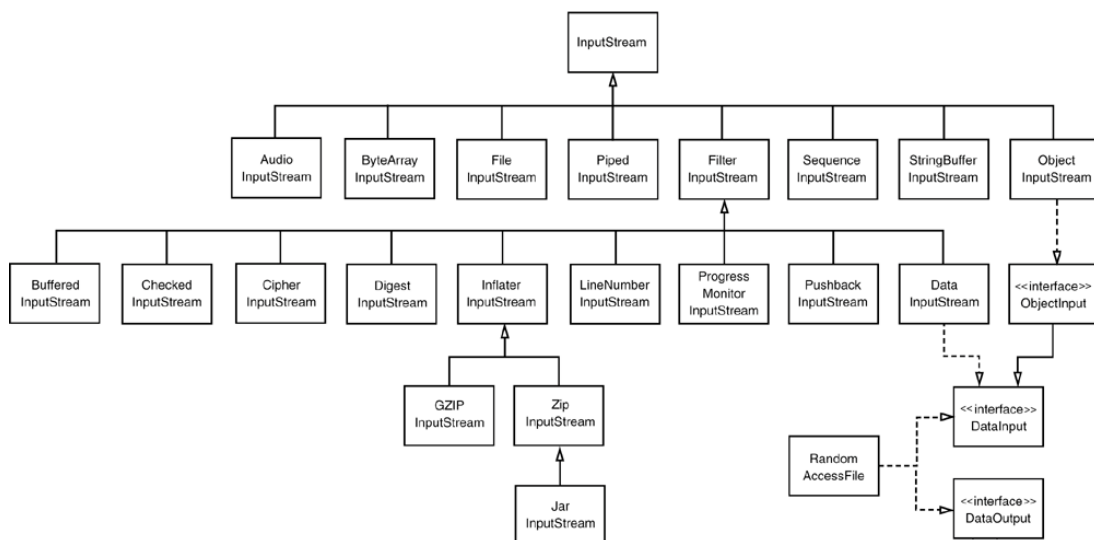
InputStream e OutputStream

- As classes **InputStream** e **OutputStream** representam respectivamente o fluxo de **entrada** e de **saída** de dados em bytes.
 - **InputStream**: Lê um fluxo de bytes;
 - **OutputStream**: Escreve um fluxo de bytes.
- A vantagem dessa abstração é que pode-se ler e escrever em qualquer tipo de entrada e saída utilizando-se os mesmos métodos.

3

3

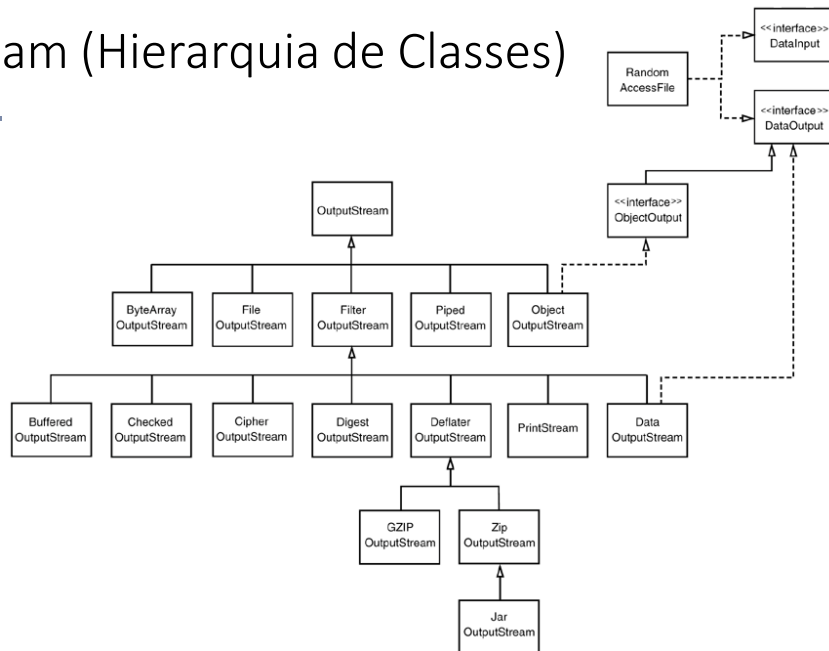
InputStream (Hierarquia de Classes)



4

4

OutputStream (Hierarquia de Classes)



5

5

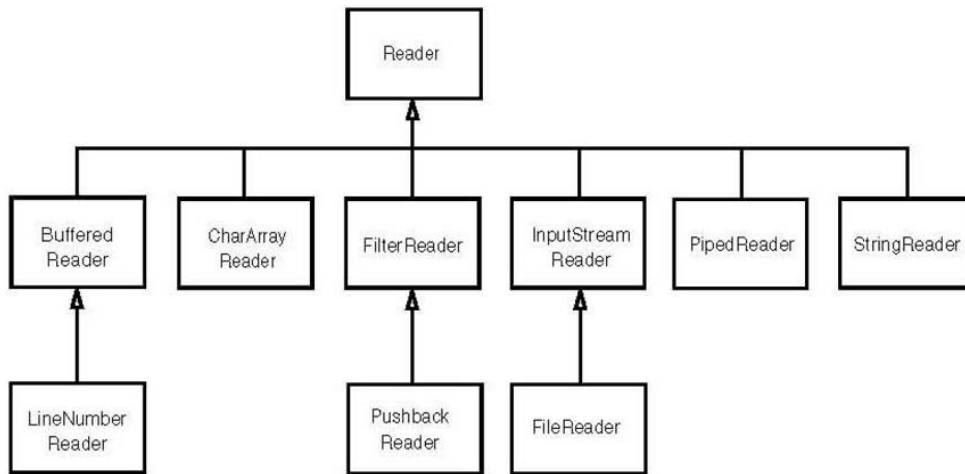
Reader e Writer

- As classes **Reader** e **Writer** definem respectivamente o fluxo de **entrada** e de **saída** de dados em caracteres.
 - **Reader**: Lê um fluxo de caracteres;
 - **Writer**: Escreve um fluxo de caracteres.
- A grande vantagem dessa abstração é que pode-se ler e escrever em qualquer tipo de entrada e saída utilizando-se os mesmos métodos.

6

6

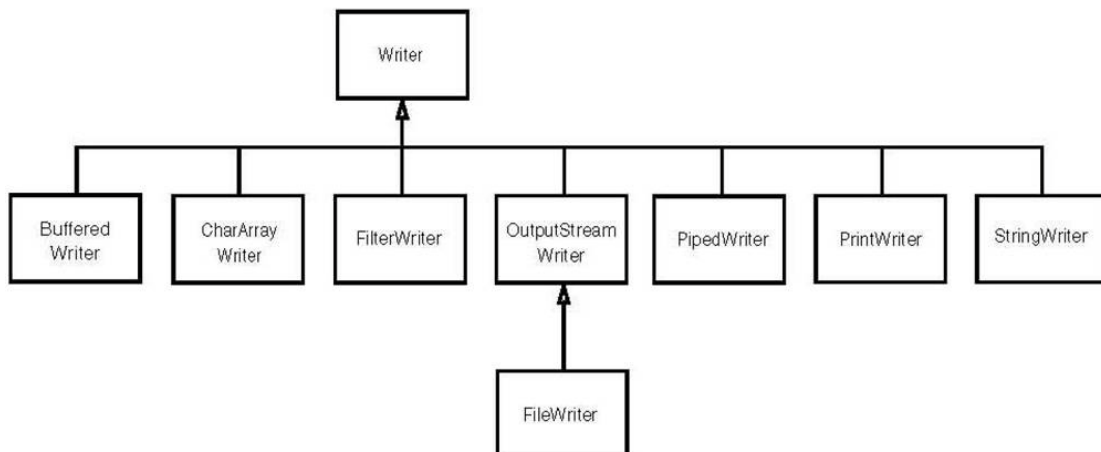
Reader (Hierarquia de Classes)



7

7

Writer (Hierarquia de Classes)



8

8

Tratamento de Exceção

- Diversos métodos do java.io lançam exceção;
- As exceções são do tipo **IOException**;
- Cada exceção do java.io deve ser obrigatoriamente **tratada** ou **declarada**;
- O tratamento da exceção é feito com o uso do bloco **try/catch**;
- Entretanto, a declaração da exceção acontece pelo uso do comando **throws**.

9

9

InputStream e Reader

- As classes **InputStream** e **Reader** têm um método chamado **read()** para ler dados da entrada;
- A classe **ObjectInputStream** tem um método chamado **readObject()** para ler objetos em formato binário da entrada de dados.

10

10

Lendo **inteiro** de arquivo **binário**

```
public class TestaEntrada {  
    public static void main(String[] args) throws IOException {  
        InputStream is = new FileInputStream("arquivo.data");  
        int b = is.read();  
        is.close();  
    }  
}
```

11

11

Lendo **caracter** de arquivo **texto**

```
public class TestaEntrada {  
    public static void main(String[] args) throws IOException {  
        Reader r = new FileReader("arquivo.txt");  
        char c = (char) r.read();  
        r.close();  
    }  
}
```

12

12

Lendo **string** de arquivo **texto**

```
public class TestaEntrada {  
    public static void main(String[] args) throws IOException {  
        Reader r = new FileReader("arquivo.txt");  
        BufferedReader br = new BufferedReader(r);  
        String s = br.readLine();  
        r.close();  
    }  
}
```

13

13

Lendo **várias strings** de arquivo **texto**

```
public class TestaEntrada {  
    public static void main(String[] args) throws IOException {  
        Reader r = new FileReader("arquivo.txt");  
        BufferedReader br = new BufferedReader(r);  
        String s = br.readLine(); // primeira linha  
        while(s != null) {  
            System.out.println(s);  
            s = br.readLine();  
        }  
        r.close();  
    }  
}
```

4

14

Lendo várias strings de arquivo **texto** usando **Scanner**

```

public class TestaEntrada {
    public static void main(String[] args) throws IOException {
        Reader r = new FileReader("arquivo.txt");
        Scanner sc = new Scanner(r);
        String s = sc.nextLine(); // primeira linha
        while(s != null) {
            System.out.println(s);
            s = sc.nextLine();
        }
        r.close();
    }
}

```

15

15

Lendo vários **objetos** de arquivo **binário**

```

public class TestaEntrada {
    public static void main(String[] args) throws IOException {
        try {
            InputStream is = new FileInputStream("arquivoBinario.data");
            ObjectInputStream ois = new ObjectInputStream(is);
            while(true) {
                Estudante e = new Estudante();
                e = (Estudante) ois.readObject();
                turma.cadastraEstudante(e);
            }
        } catch (EOFException ex) { }
    }
}

```

16

OutputStream e Writer

- As classes **OutputStream** e **Writer** têm um método chamado **write()** para escrever dados na saída;
- A classe **ObjectOutputStream** tem um método chamado **writeObject()** para escrever objetos em formato binário na saída de dados.

17

17

Escrevendo inteiro em arquivo binário

```
public class TestaEntrada {  
    public static void main(String[] args) throws IOException {  
        OutputStream os = new FileOutputStream("arquivo.data");  
        int b = 10;  
        os.write(b);  
        os.close();  
    }  
}
```

18

18

Escrevendo **string** em arquivo **texto**

```
public class TestaSaida {  
    public static void main(String[] args) throws IOException {  
        Writer w = new FileWriter("arquivo.txt");  
        BufferedWriter bw = new BufferedWriter(w);  
        bw.write("Alô mundo!");  
        bw.close();  
    }  
}
```

19

19

Escrevendo **objeto** em arquivo **binário**

```
class TestaSaida {  
    public static void main(String[] args) throws IOException {  
        OutputStream os = new FileOutputStream("arquivo.data");  
        ObjectOutputStream oos = new ObjectOutputStream(os);  
        Estudante e = new Estudante("João da Silva");  
        oos.writeObject(e);  
        oos.close();  
    }  
}
```

20