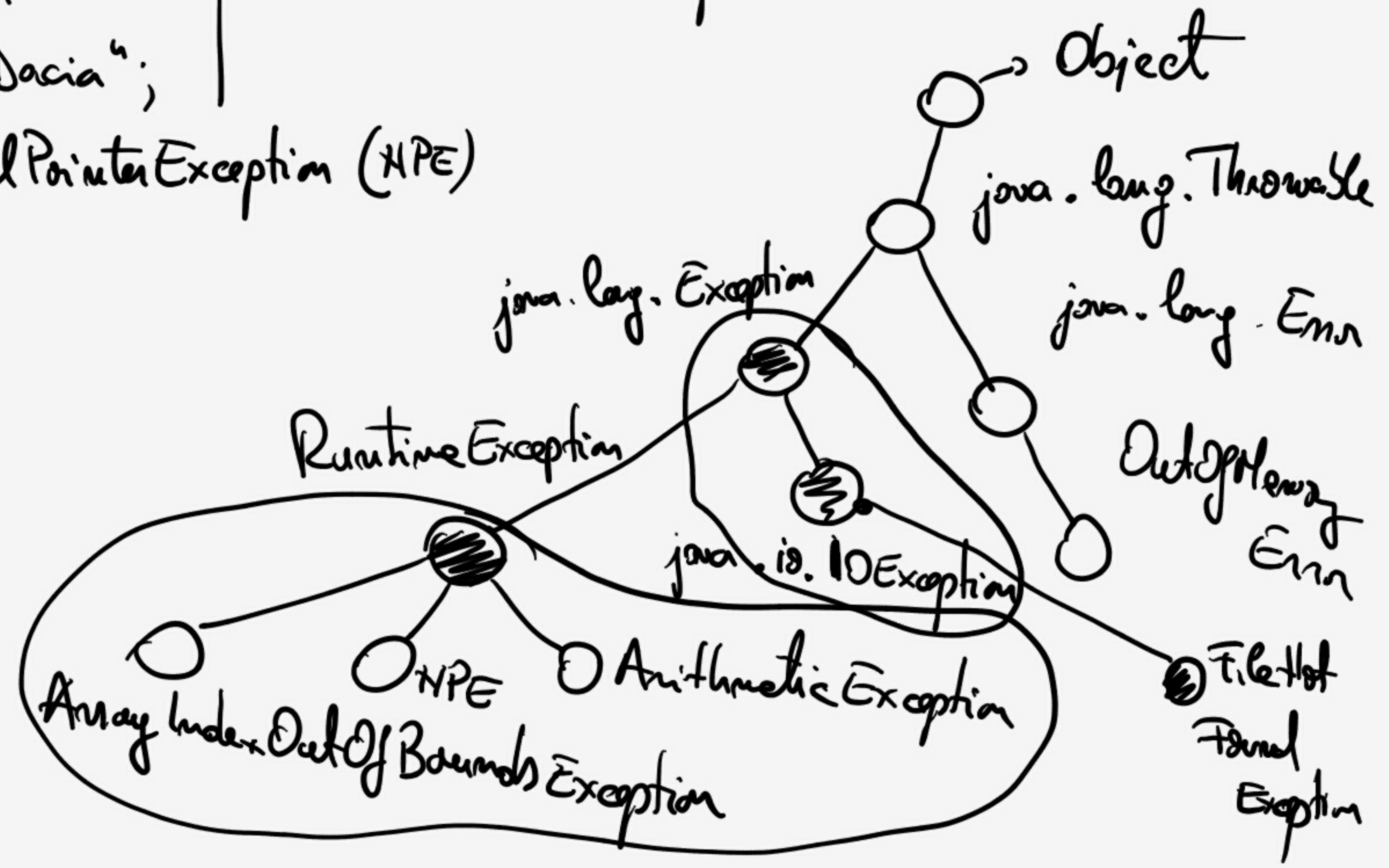


④

```
Car c; // null
c.make = "Dacia";
```

↳ Null Pointer Exception (NPE)

Exceptiile → evenimente neprevăzute care apar în timpul execuției



Tratarea Exceptiilor

①

primul (1)

al doilea (2)

```
void openFile()
```

```
try {
```

```
    FileInputStream fis;
```

```
    → fis = new FileInputStream("fisier.txt"); // throws IOException
```

```
    int c = fis.read(); // nu se execută în caz de excepție
```

```
    } catch (IOException e) {
```

```
        → System.out.printf("%s\n", e.getMessage());
```

```
    } catch (FileNotFoundException e) {
```

```
        System.out.printf("FNF\n");
```

```
    } finally { // → se execută în ambele cazuri  
        // se eliberează resurse
```

```
    }  
    int a = 4;
```

```
void init() {
```

```
    openFile();
```

```
}
```

②

void openFile () throws IOException, NullPointerException } optional

→ FileInputStream fis = new FileInputStream ("--");
int c = fis.read();

}

void init () {

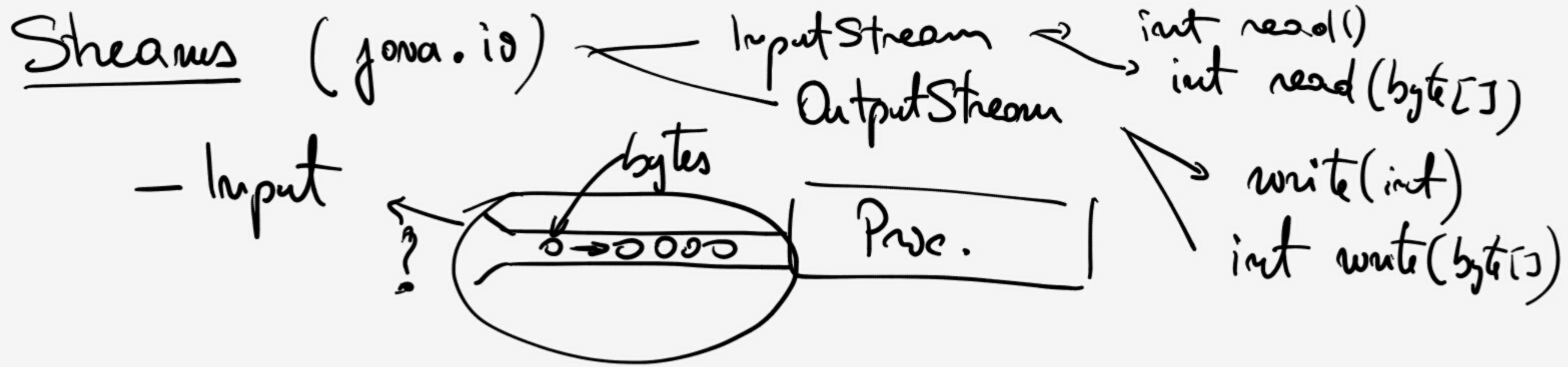
→ openFile ();
}

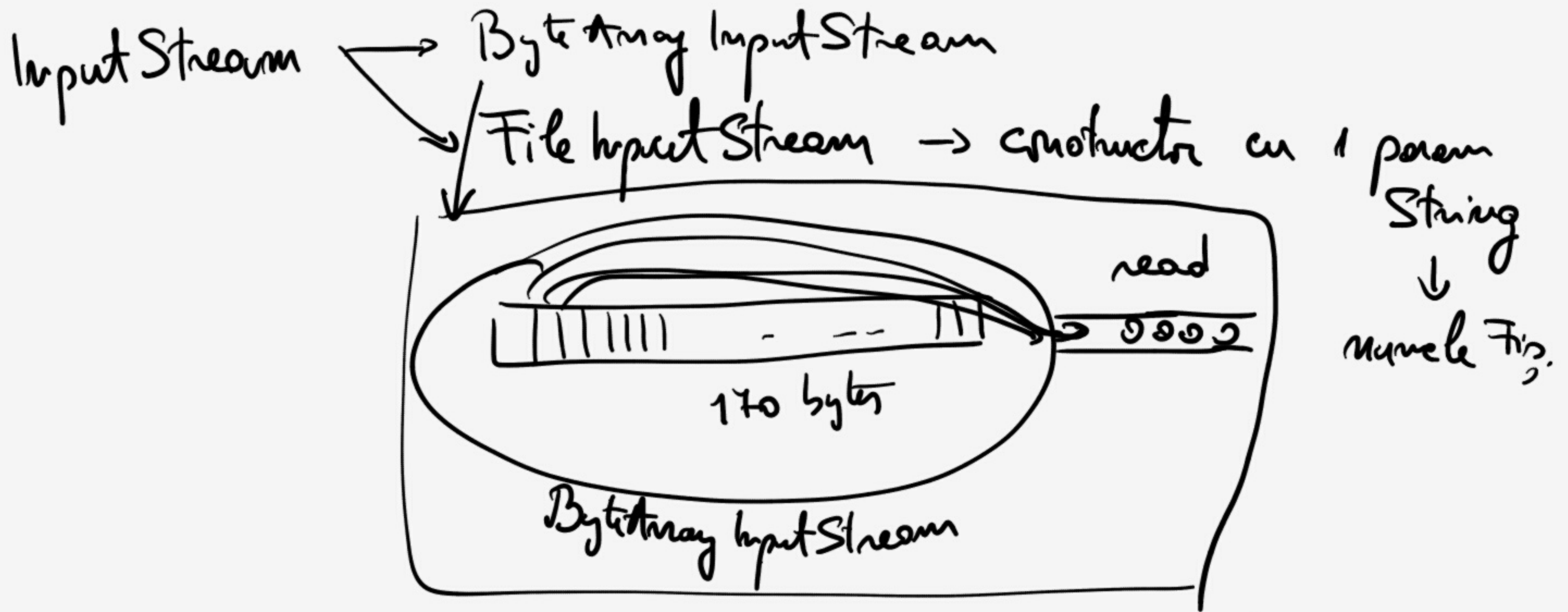
Obligatoriu pt. exceptii nehotate
Cae nu sunt de tip Error sau
RuntimeException

```

class Persona {
    private int mVarsta;
    public void setVarsta(int varsta) throws I...A...E {
        → if (varsta < 0) throw new IllegalArgumentException(
            "Varsta " + varsta + " e mai
            mica ca 0");
        mVarsta = varsta;
    }
}

```





```

FileInputStream fis = new FileInputStream("fisier.txt");
long size = (new File("fisier.txt")).length();
byte[] buffer = new byte[(int) size];
fis.read(buffer);

```

Analog pentru scriere cu write ↔ read
Out ↔ In

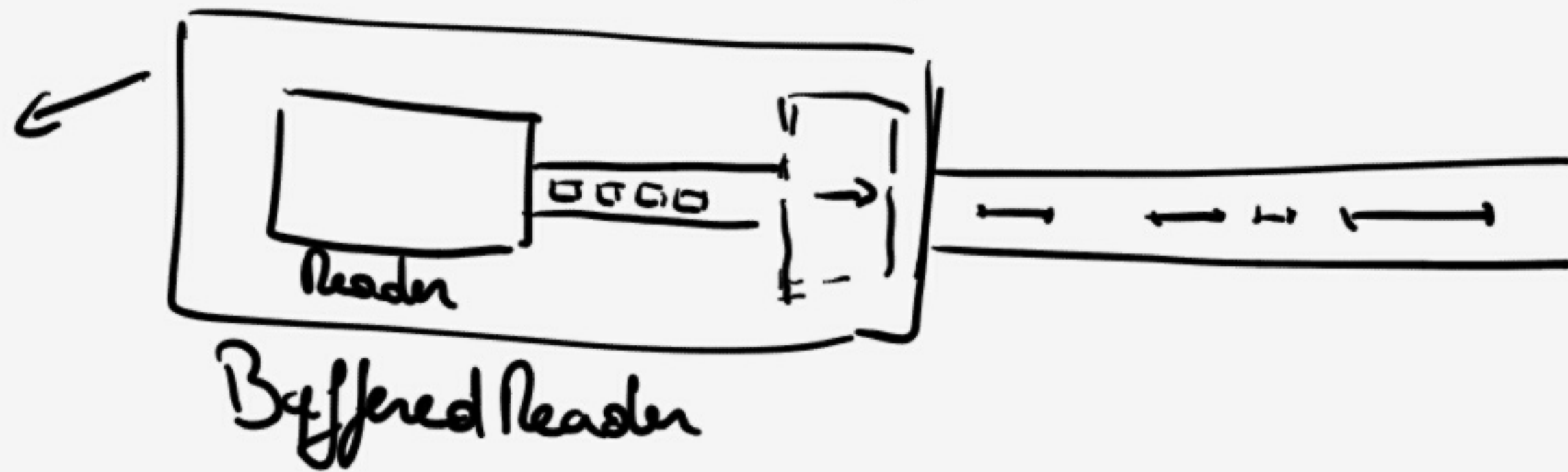
Nota: utilizati
documentatia
JAVA A7i 8

pentru fisiere de tip text

Reader → FileReader → read
Writer → FileWriter → write

BufferedReader → String readLine()

Wrapper
Class



```
try {
    Reader r = new FileReader("files.txt");
    BufferedReader bufR = new BufferedReader(r);
    String line = bufR.readLine();
}
```

String.

- split
- trim
- startsWith
- charAt

Integer.parseInt →

String s = "123";

int n = Integer.parseInt(s);

static

Un membru static al unei clase nu depinde de instanța clasei.

```
class Car {  
    String color;  
    static int carCount;  
    static void incCarCount() {  
        carCount++;  
    }  
}
```

```
Car.incCarCount();
```

