



# Principet e Avancuara të Gjuheve të Programimit

Lamir Shkurti, Can. PhD.  
lamirshkurti@gmail.com

# JAVA I/O STREAMS

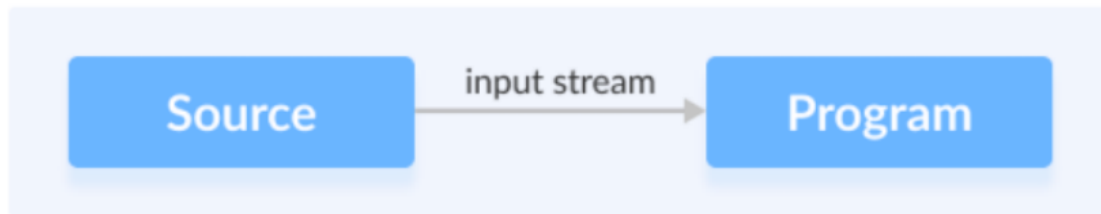


Në Java, streams janë sekuenca e të dhënave që lexohen nga burimi dhe shkruhen në destinacion.

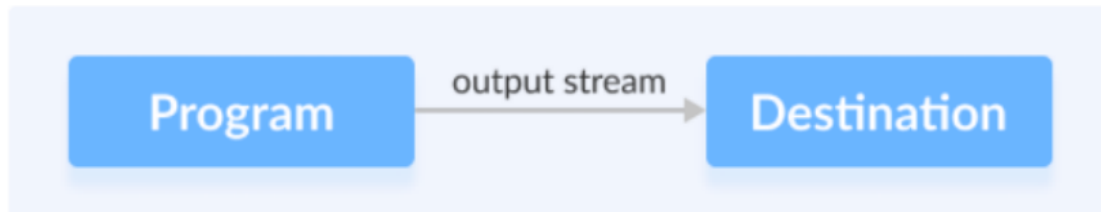
- një **input stream** përdoret për të lexuar të dhëna nga burimi dhe
- një **output stream** përdoret për të shkruar të dhëna në destinacion.

# JAVA I/O STREAMS

Reading data from source



Writing data to destination



# JAVA I/O STREAMS – LLOJET E STREAMS

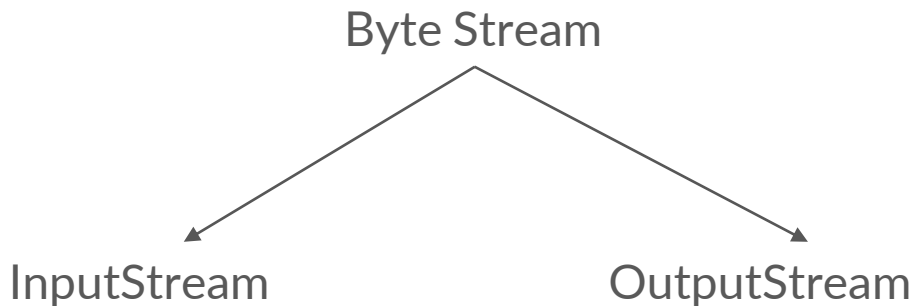


Në varësi të të dhënave që mban një stream, ajo mund të klasifikohet në:

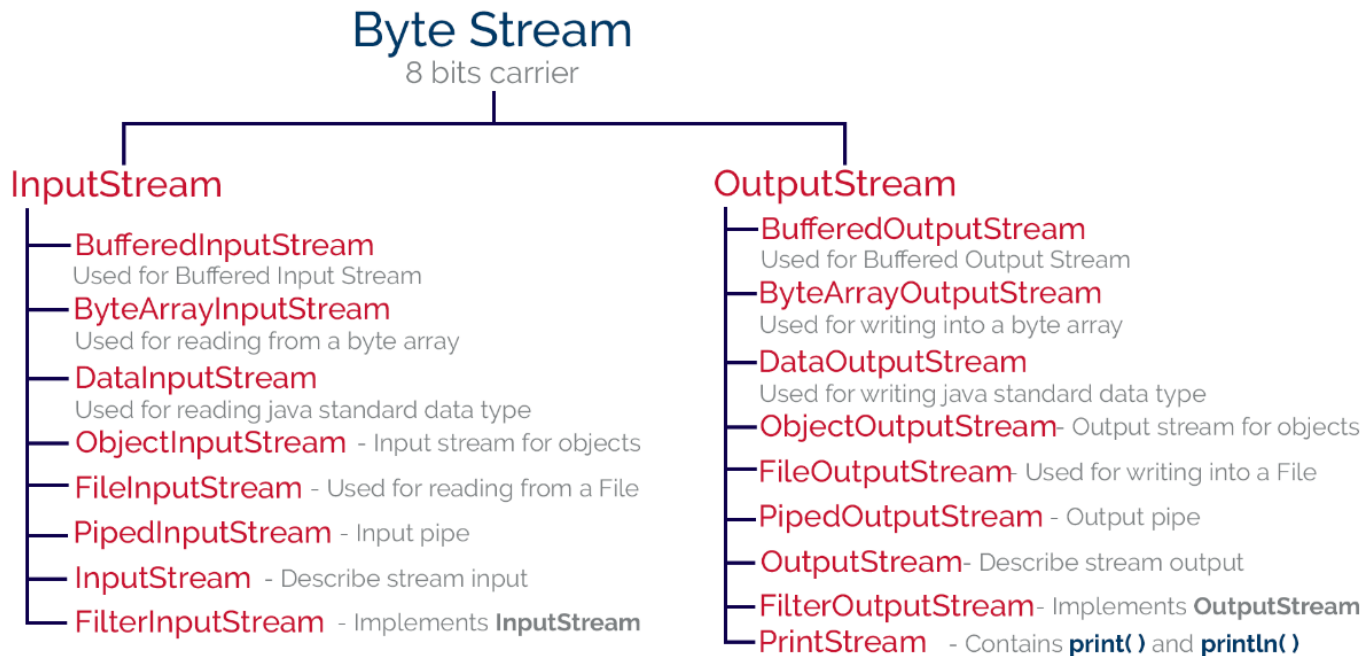
- **Byte Stream**
- **Character Stream**

## JAVA I/O STREAMS – BYTE STREAM

- Byte stream përdoret për të lexuar dhe shkruar një bajt të vetëm (8 bit) të të dhënave.
- Të gjitha klasat e byte stream rrjedhin nga klasat abstrakte të quajtura `InputStream` dhe `OutputStream`.



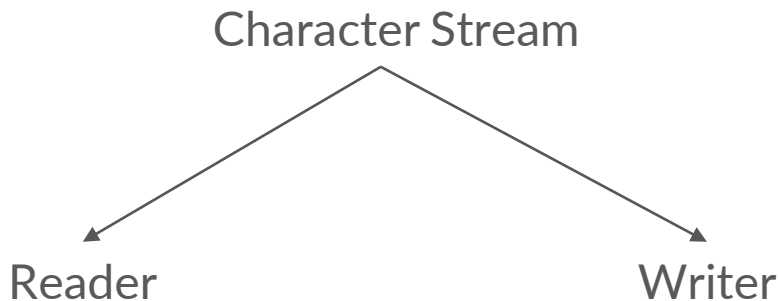
# JAVA I/O STREAMS – BYTE STREAM



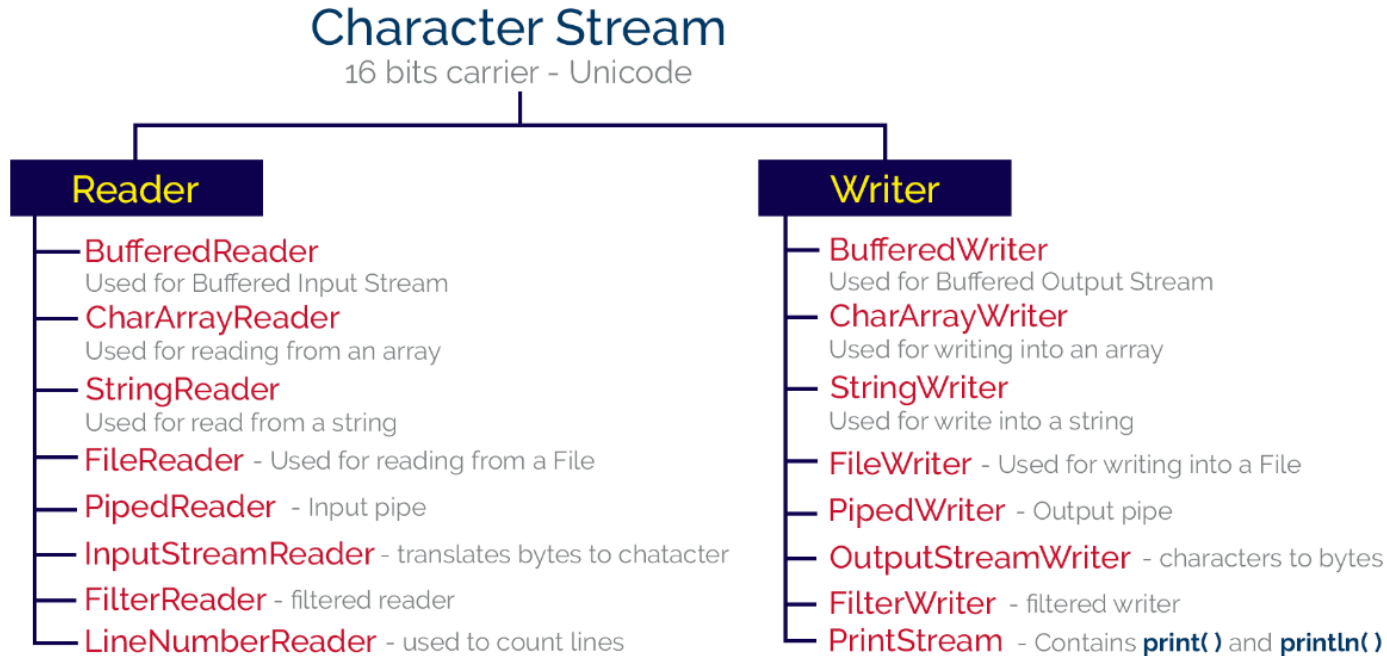
Dy prej metodave më të rëndësishme: `read()` dhe `write()`

# JAVA I/O STREAMS – CHARACTER STREAM

- Character Stream përdoret për të lexuar dhe shkruar një karakter të vetëm të të dhënave.
- Të gjitha klasat e character stream rrjedhin nga klasat abstrakte Reader dhe Writer.



# JAVA I/O STREAMS – CHARACTER STREAM





# JAVA I/O STREAMS

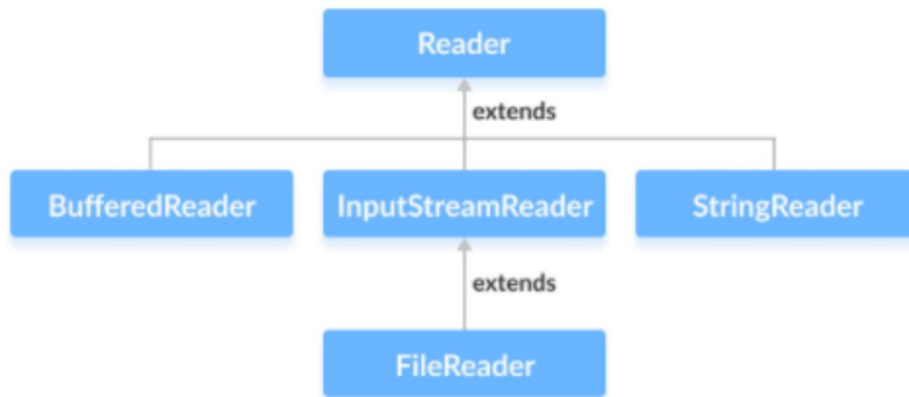


- **Fokusi** jonë do të jetë në stream të bazuar në **karaktere** (character stream)
  - **Reader**
  - **Writer**

# READER

- Klasa Reader e paketës java.io është një superklasë abstrakte që përfaqëson një stream të karaktereve.
- Meqenëse Reader është një klasë abstrakte, nuk është e dobishme në vetvete. Sidoqoftë, nënklasat e saj mund të përdoren për të lexuar të dhëna.

- Nënklasat e klasës Reader



# METODAT E KLASËS `Reader`



- `ready()` - checks if the reader is ready to be read
- `read(char[] array)`- reads the characters from the stream and stores in the specified array
- `read(char[] array, int start, int length)` -reads the number of characters equal to length from the stream and stores in the specified array starting from the start
- `mark()` - marks the position in the stream up to which data has been read
- `reset()` - returns the control to the point in the stream where the mark is set
- `skip()` - discards the specified number of characters from the stream

# FILE READER - SHEMBULL



- Supozoni se kemi një file me emrin input.txt me përmbajtjen vijuese.

This is a line of text inside the file

.

- Të lexojmë këtë file duke përdorur FileReader (një nënklasë e Reader).

# FILE READER - SHEMBULL

```
import java.io.Reader;
import java.io.FileReader;

public class Main {
    public static void main(String[] args) {

        // Creates an array of character
        char[] array = new char[100];

        try {
            // Creates a reader using the FileReader
            Reader input = new FileReader("input.txt");

            // Checks if reader is ready
            System.out.println("Is there data in the stream? " + input.ready());

            // Reads characters
            input.read(array);
            System.out.println("Data in the stream:");
            System.out.println(array);

            // Closes the reader
            input.close();
        }

        catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

# BufferedReader



- **BufferedReader** mban një buffer të brendshëm prej 8192 karakteresh.
- Gjatë leximit përmes **BufferedReader**, një pjesë e karaktereve lexohet nga disku dhe ruhet në bufferin e brendshëm dhe pastaj nga bufferi i brendshëm lexohen individualisht. Prandaj, komunikimi me disk zvogëlohet. Kjo është arsyeja pse leximi i karaktereve është më i shpejtë duke përdorur **BufferedReader**.

# BufferedReader



- Në mënyrë që të krijojmë një `BufferedReader`, duhet së pari të importojmë: `java.io.BufferedReader`

```
// Creates a FileReader
```

```
FileReader file = new FileReader(String file);
```

```
// Creates a BufferedReader
```

```
BufferedReader buffer = new BufferedReader(file);
```

# Metodat e klasës **BufferedReader**



- **read() Method**
  - **read()**- reads a single character from the internal buffer of the reader
  - **read(char[] array)** - reads the characters from the reader and stores in the specified array
  - **read(char[] array, int start, int length)**- reads the number of characters equal to length from the reader and stores in the specified array starting from the position start



## BufferedReader - Shembull



- Supozoni se kemi një file me emrin input.txt me përmbajtjen vijuese.

This is a line of text inside the file

.

- Të lexojmë këtë file duke përdorur BufferedReader

# BufferedReader - Shembull

```
import java.io.FileReader;
import java.io.BufferedReader;

class Main2 {
    public static void main(String[] args) {
        // Creates an array of character
        char[] array = new char[100];

        try {
            // Creates a FileReader
            FileReader file = new FileReader("input.txt");

            // Creates a BufferedReader
            BufferedReader input = new BufferedReader(file);

            // Reads characters
            input.read(array);
            System.out.println("Data in the file: ");
            System.out.println(array);

            // Closes the reader
            input.close();
        }

        catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

# Writer

- Klasa **Writer** e paketës **java.io** është një klasë abstrakte që përfaqëson një stream të karaktereve.
- Meqenëse **Writer** është një klasë abstrakte, nuk është e dobishme në vetvete. Sidoqoftë, nënklasat e saj mund të përdoren për të shkruar të dhëna.

- Nënklasat e Writer



# Metodat e klasës Writer



## Metodat e klasës Writer:

- **write(char[] array)** - writes the characters from the specified array to the output stream
- **write(String data)** - writes the specified string to the writer
- **append(char c)** - inserts the specified character to the current writer
- **flush()** - forces to write all the data present in the writer to the corresponding destination
- **close()** - closes the writer

# Writer - Shembull

```
import java.io.FileWriter;
import java.io.Writer;

public class Main3 {

    public static void main(String args[]) {

        String data = "This is the data in the output file";

        try {
            // Creates a Writer using FileWriter
            Writer output = new FileWriter("output.txt");

            // Writes string to the file
            output.write(data);

            // Closes the writer
            output.close();
        }

        catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

## Principet e Avancuara të Gjuhëve të Programimit

