

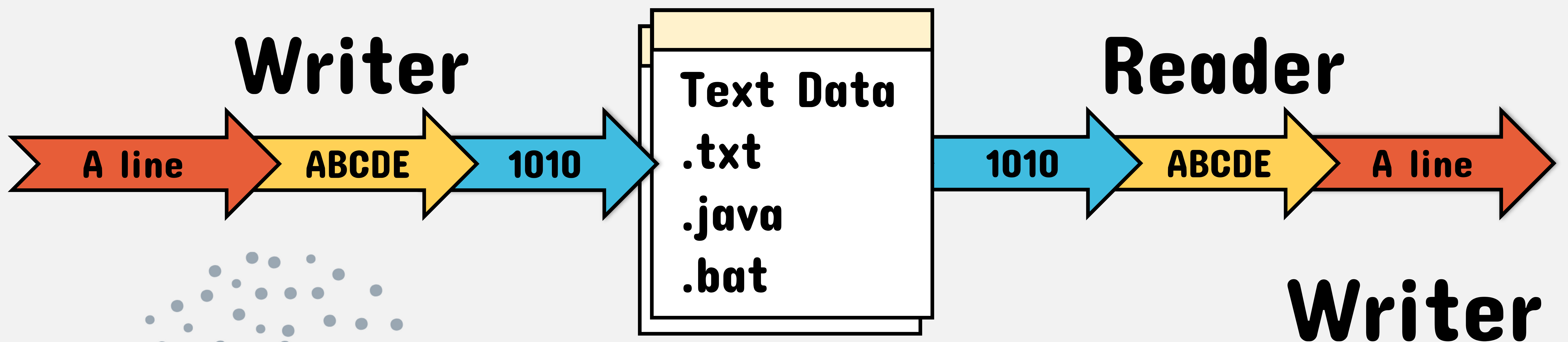
TEXT I/O

Overview

Text I/O enables a program to **read** and **write** text. It is possible to redirect the input or output to files or to other input and output streams. It includes **Reader** and **Writer** classes.

Why Important?

It is highly important for a computer program to **interact with users**. Text I/O helps a Java program to be interactive with user by being able to request input from the user, process it, and displays the result.



Writer class is an abstract class for writing to character streams.
Methods: **close()**, **flush()**, **write(String str)**

BufferedWriter

Writes text to a character-output stream. Override method from writer class.

Method: void **newLine()**

PrintWriter

Prints formatted representations of objects to a text-output stream. Override method from writer class.

Methods:

- boolean **checkError()**
- void **clearError()**
- void **print(String str)**
- void **println(String str)**
- void **printf(String format, Object.... args)**

OutputStreamWriter

Bridge from character streams to byte streams. Override method from writer class.

Method: String **getEncoding()**

FileWriter

Writing character files. Override method from OutputStreamWriter class.

Reader

Reader class is an abstract class for reading to character streams.
Methods: **close()**, **read()**, **reset()**, **skip(long n)**

BufferedReader

Reads text from a character-input stream. Override method from reader class.

Method: void **readLine()**

InputStreamReader

InputStreamReader converts bytes to characters. Override method from reader class.

Method: String **getEncoding()**

FileReader

Reading character files. Override method from InputStreamReader class.

Code Examples

Reader Examples

```
// Read input from the keyboard
public static void main(String[] args) throws
IOException{
    BufferedReader input = new BufferedReader(new
    InputStreamReader(System.in));
    String str = input.readLine();
    input.close();
}

// Read input from a file
public static void main(String[] args) throws
IOException{
    BufferedReader input = new BufferedReader(new
    FileReader("file/example.txt"));
    String str = input.readLine();
    input.close();
}
```

Writer Example

```
// Write output to a file
public static void main(String[] args) throws
IOException{
    PrintWriter output = new PrintWriter(new
    BufferedWriter(new
    FileWriter("file/example.txt")));
    output.print("Hello");
    output.println(" World!");
    output.close();
}
// Don't forget to import java.io !!!
```

Sources & References

- <https://math.hws.edu/javanotes/source/chapter2/TextIO.java>
- <https://www.codeproject.com/Articles/2853/Java-Basics-Input-and-Output>
- <https://www.coursehero.com/file/p6ubsdb/7-What-is-the-difference-between-text-IO-and-binary-IO-In-text-mode-a-file-is/>
- <https://docs.oracle.com/javase/7/docs/api/java/io/Writer.html>
- <https://docs.oracle.com/javase/7/docs/api/java/io/Reader.html>
- <https://www.slideshare.net/sunilos/java-input-output-and-file-handling>
- <http://web.cse.ohio-state.edu/software/2231/web-sw2/extras/slides/32.Java-Input-Output.pdf>
- <https://intellipaat.com/tutorial/java-tutorial/java-file-io/>
- <https://www.programcreek.com/wp-content/uploads/2012/05/java-io-class-hierarchy-diagram.gif>
- <http://www.falkhausen.de/java-8/java.io/Writer.html>
- http://ocw.ui.ac.id/pluginfile.php/1483/mod_resource/content/0/10-fop2-text-io_rev5.pdf
- Icons/ornaments from Ms. Powerpoint and canva.com