In questo esercizio si utilizza una comunicazione tramite un **canale di comunicazione** che è una **pipe**, sfruttando la classe **PipedOutputStream** e **PipedInputStream**, tra un **Thread Writer** e un **Thread Reader**.

* PipedOutputStream è utilizzata per scrivere dati sulla pipe.
* PipedInputStream è utilzzata per leggere dati sulla pipe.

Nel **main** creiamo un oggetto **PipedOutputStream** che verrà poi passato sia a **WriterThread** che a **Reader**.  
Ovviamente per il **Writer** va bene così com’è dato che deve letteralmente scriverci dentro, mentre il **Reader** deve ottenere il **riferimento** alla **pipe** creata da quest’oggetto **PipedOutputStream**, in modo da poter lavorare tramite **PipedInputStream** per leggerci.

Definiamo i **thread Reader** e **Writer** per **ereditarietà**.

**Writer** deve:

* **leggere** dati dalla **tastiera**, sfrutta un oggetto **keyboardBuf** di tipo **BufferedReader** che viene utilizzato per leggere un **flusso** di **dati in input** cioè **System.in** ( un oggetto di tipo InputStream che è associato all’input da tastiera), passando per la classe **InputStreamReader** che è utilizzato per convertire il flusso di byte in un flusso di caratteri.
  + Assegna la stringa letta in **keyboardBuf** ad s.
* inviare i dati sulla pipe, sfrutta un oggetto di tipo **DataOutputStream** che estende **OutputStream** e ha la particolarità di attuare la **conversione dei dati primitivi** in **binario** (senza doverlo fare a mano).

**Reader** deve:

* ricavare da dove **leggere** i dati, dunque passandogli il **PipedOutputStream** creato dal main, ricava da dove leggere tramite un **oggetto** di classe **DataInputStream** creato tramite un **PipedInputStream** collegato al **PipedOutputStream**.
* Leggere tramite quell’oggetto **DataInputStream**.