```
public class InputStreamReader
extends Reader
```

An InputStreamReader is a bridge from byte streams to character streams: It reads bytes and decodes them into characters using a specified `charset`. The charset that it uses may be specified by name or may be given explicitly, or the platform's default charset may be accepted.

Each invocation of one of an InputStreamReader's read() methods may cause one or more bytes to be read from the underlying byte-input stream. To enable the efficient conversion of bytes to characters, more bytes may be read ahead from the underlying stream than are necessary to satisfy the current read operation.

For top efficiency, consider wrapping an InputStreamReader within a BufferedReader. For example:

```
BufferedReader in
  = new BufferedReader(new InputStreamReader(System.in));
```

Every implementation of the Java platform is required to support the following standard charsets. Consult the release documentation for your implementation to see if any other charsets are supported. The behavior of such optional charsets may differ between implementations.

Charset	Description
`US-ASCII`	Seven-bit ASCII, a.k.a. `ISO646-US`, a.k.a. the Basic Latin block of the Unicode character set
`ISO-8859-1`	ISO Latin Alphabet No. 1, a.k.a. `ISO-LATIN-1`
`UTF-8`	Eight-bit UCS Transformation Format
`UTF-16BE`	Sixteen-bit UCS Transformation Format, big-endian byte order
`UTF-16LE`	Sixteen-bit UCS Transformation Format, little-endian byte order
`UTF-16`	Sixteen-bit UCS Transformation Format, byte order identified by an optional byte-order mark

```
public class BufferedReader
extends Reader
```

Reads text from a character-input stream, buffering characters so as to provide for the efficient reading of characters, arrays, and lines.

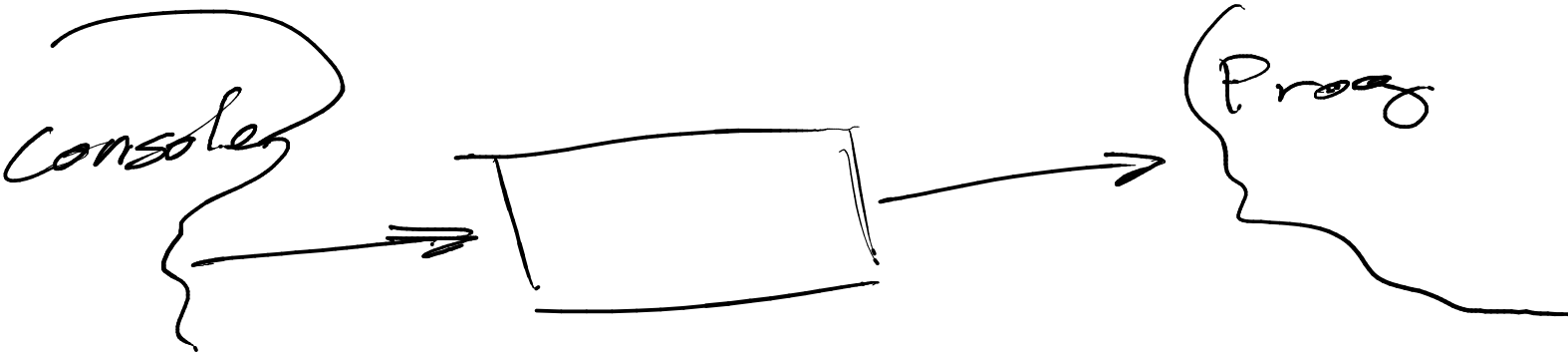
The buffer size may be specified, or the default size may be used. The default is large enough for most purposes.

In general, each read request made of a Reader causes a corresponding read request to be made of the underlying character or byte stream. It is therefore advisable to wrap a BufferedReader around any Reader whose read() operations may be costly, such as FileReaders and InputStreamReaders. For example,

```
BufferedReader in
  = new BufferedReader(new FileReader("foo.in"));
```

will buffer the input from the specified file. Without buffering, each invocation of read() or readLine() could cause bytes to be read from the file, converted into characters, and then returned, which can be very inefficient.

Programs that use DataInputStreams for textual input can be localized by replacing each DataInputStream with an appropriate BufferedReader.



Aufgabe 3.3

Die Webseite `http://checkip.dyndns.com` zeigt die zugreifende IP-Adresse an.

Schreiben Sie ein Java-Programm, das die aktuelle IP-Adresse Ihres Computers anzeigt. Sie sollten dazu in einem ersten Schritt die erste Zeile der Seite auslesen und sich anschließend überlegen, mit welchen Methoden zur Zeichenkettenmanipulation die IP-Adresse extrahiert werden kann.

Hinweis: Die Methode `indexOf` der Klasse String ermöglicht die Suche von Textelementen.