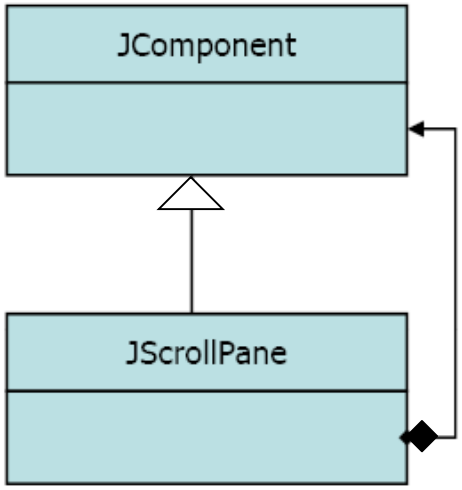
Decorator Examples

- JScrollPane
- Collection Decorators
- Java IO (Streams)

Decorator Patterns: Java API Samples

- **JScrollPane**



  - Constructor:

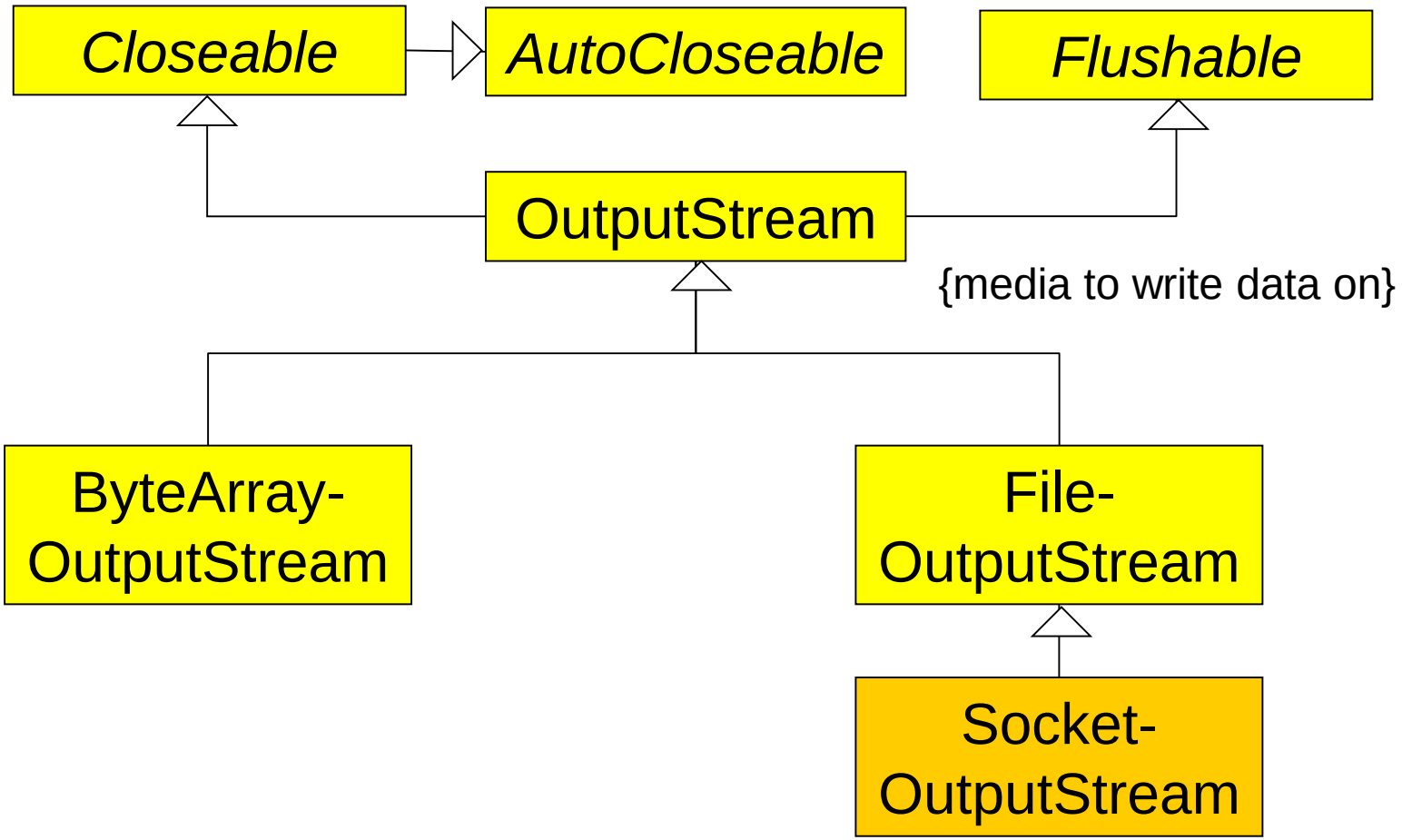
```
JScrollPane(Component view)
```

Decorator Patterns: Java API Samples

- **Collections**
  - Collections.unmodifiableCollection/Set/SortedSet/List/Map/SortedMap
  - Collections.synchronizedCollection/Set/SortedSet/List/Map/SortedMap

```
class SynchronizedList implements List {
   private final List list;
   public SynchronizedList(List list) { this.list = list; }

   public synchronized Object get(int index) {
      return list.get(index);
   }
   public synchronized boolean contains(Object x) {
      return list.contains(x);
   }
   ...
```

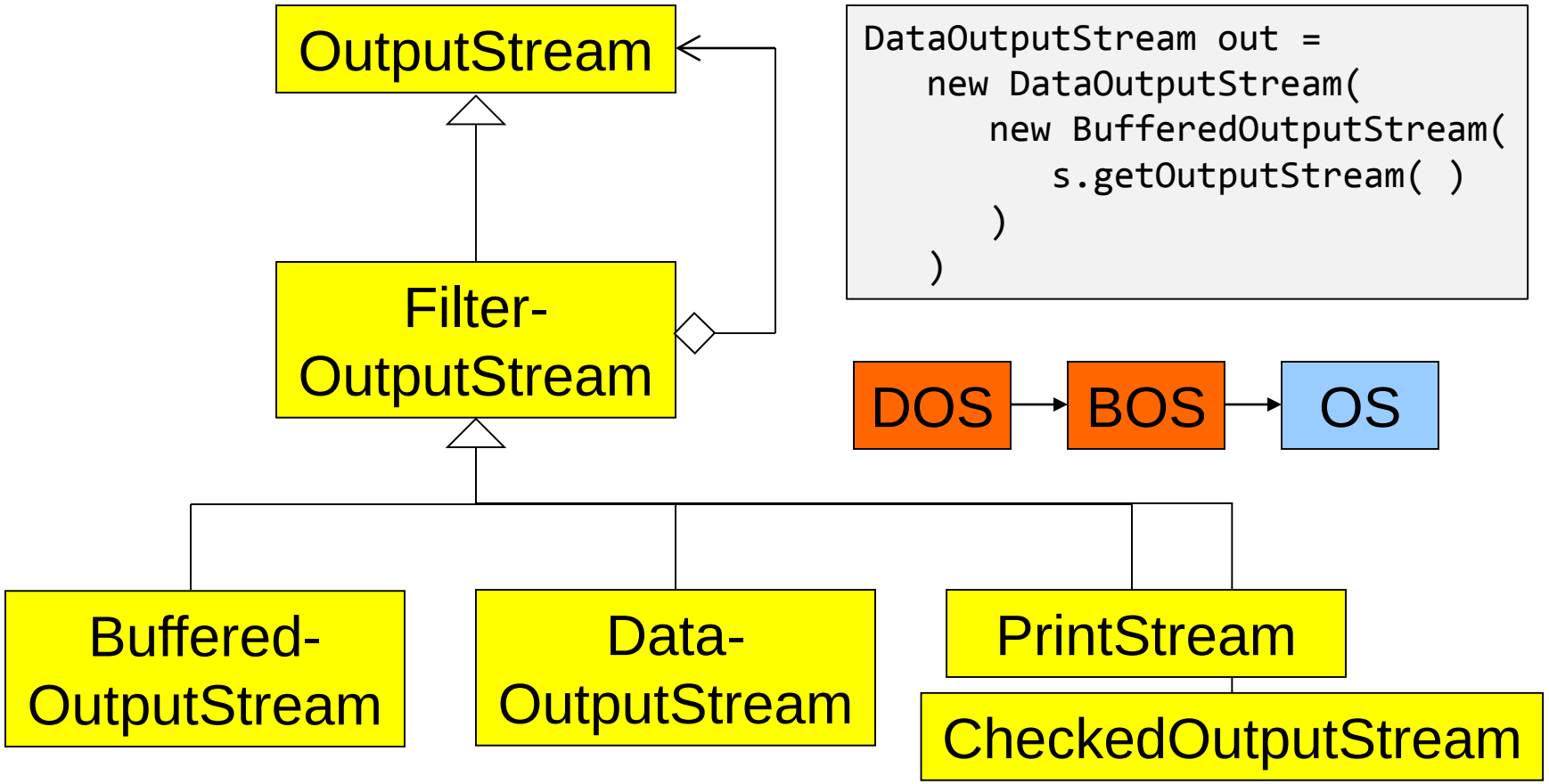
OutputStream



OutputStream

```
abstract class OutputStream implements Closeable, Flushable {
   public abstract void write(int b)      // writes a byte (0..255)
                  throws IOException;     // (byte is -128..127)

   public void write(byte[] data)
                  throws IOException;
   public void write(byte[] data, int offset, int length)
                  throws IOException;

   public void flush()                    // in this class empty
                  throws IOException;
   public void close()
                  throws IOException
}
```

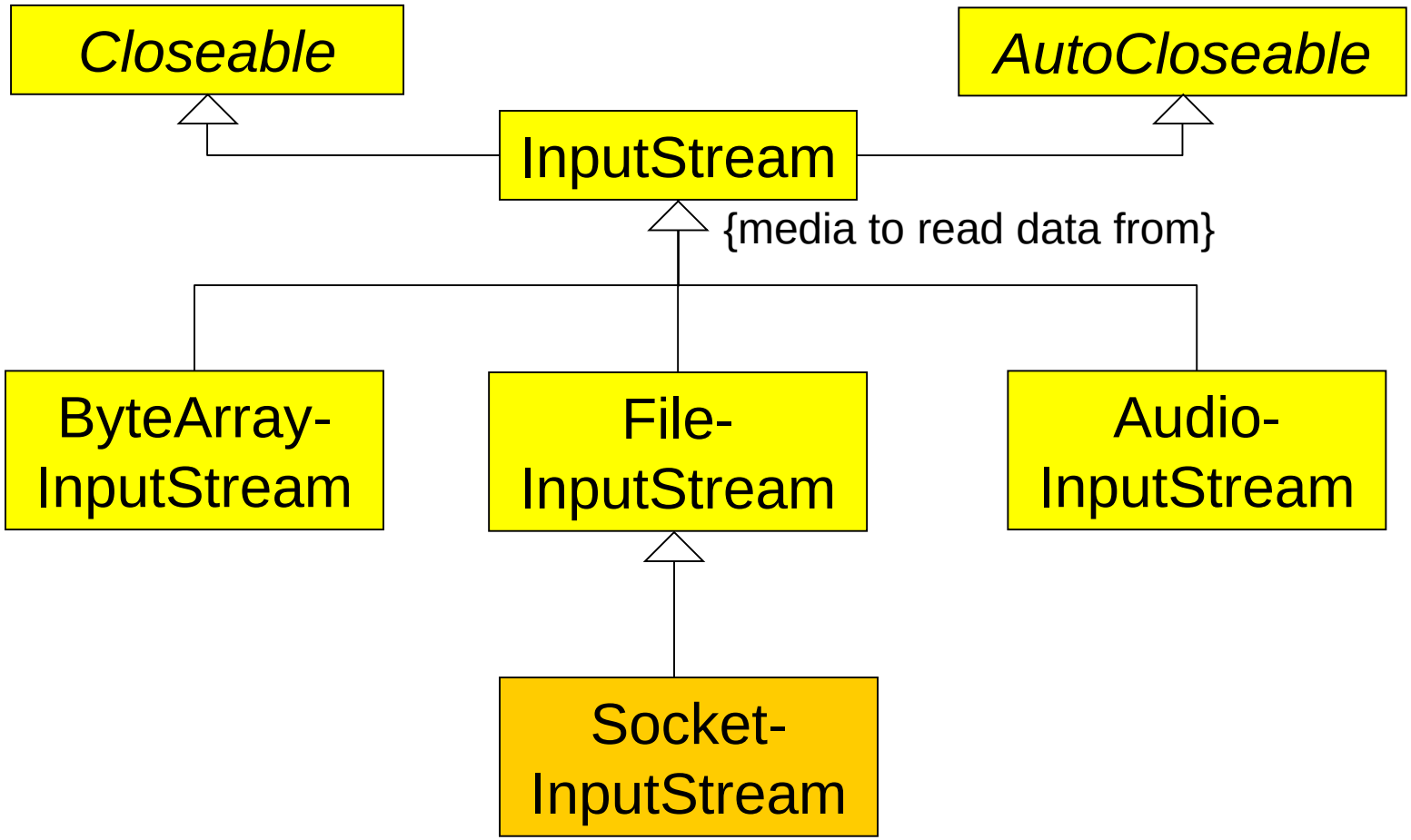
FilterOutputStreams



DataOutputStream

```
abstract class DataOutputStream extends FilterOutputStream {
   public final void writeBoolean(boolean b)   throws IOException;
   public final void writeByte(int b)          throws IOException;
   public final void writeShort(int s)         throws IOException;
   public final void writeChar(int c)          throws IOException;
   public final void writeInt(int i)           throws IOException;
   public final void writeLong(long l)         throws IOException;
   public final void writeFloat(float f)       throws IOException;
   public final void writeDouble(double d)     throws IOException;
   public final void writeChars(String s)      throws IOException;
                  // two-byte unicode characters, length not encoded
   public final void writeBytes(String s)      throws IOException;
                  // one-byte Latin-1 characters, length not encoded
   public final void writeUTF(String s)        throws IOException;
                  // writes string using UTF-8 encoding
}
```

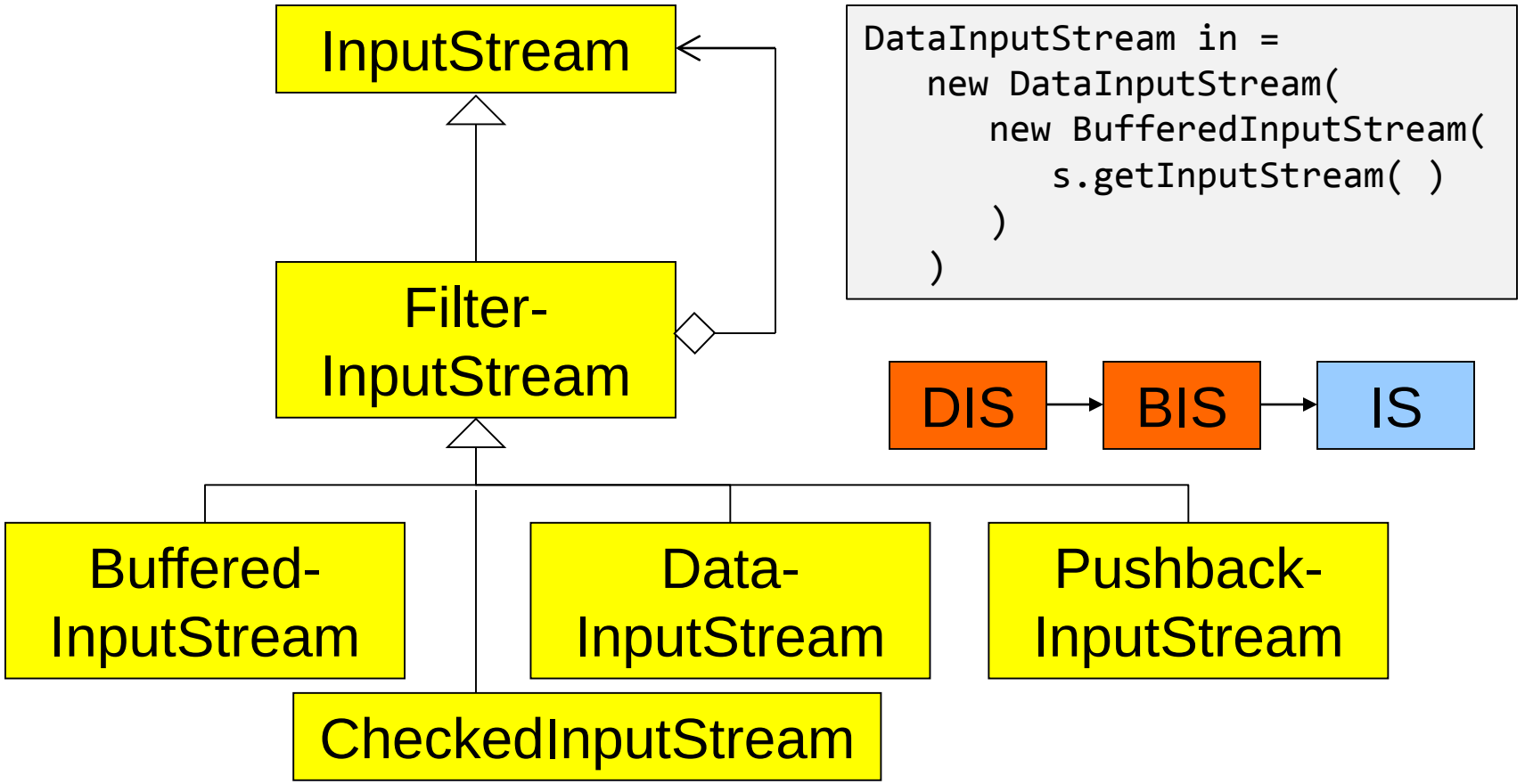
InputStream



InputStream

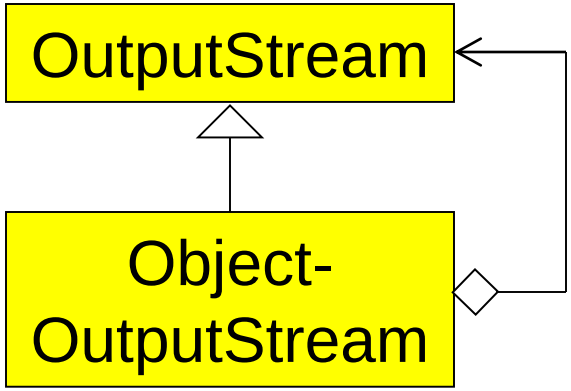
```
abstract class InputStream implements Closeable {
   public abstract int read()              //      blocking        -1=EOS
                  throws IOException;
   public int read(byte[] input)           //      blocking        -1=EOS
                  throws IOException;
   public int read(byte[] input, int offset, int length)
                  throws IOException;      //      blocking        -1=EOS
   public long skip(long n)
                  throws IOException;
   public int available()                  // >= 0 [default impl: 0]
                  throws IOException;
   public void close()                     //      [default impl: {}]
                  throws IOException

   public void mark(int readAheadLimit);
   public void reset() throws IOException;
   public boolean markSupported();         //      [default impl: false]
}
```

FilterInputStreams

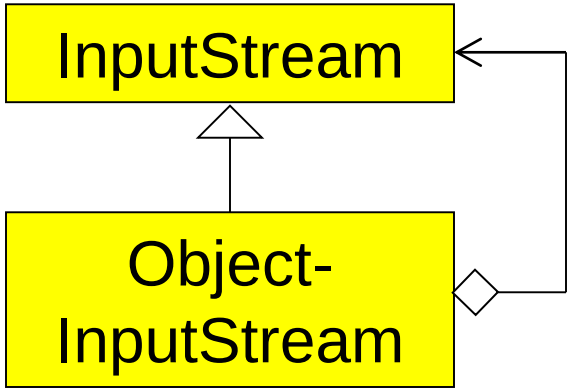


ObjectStreams

```
abstract class ObjectOutputStream {
   public final void writeObject(
                            Object obj)
       throws IOException;
   ...
}
```



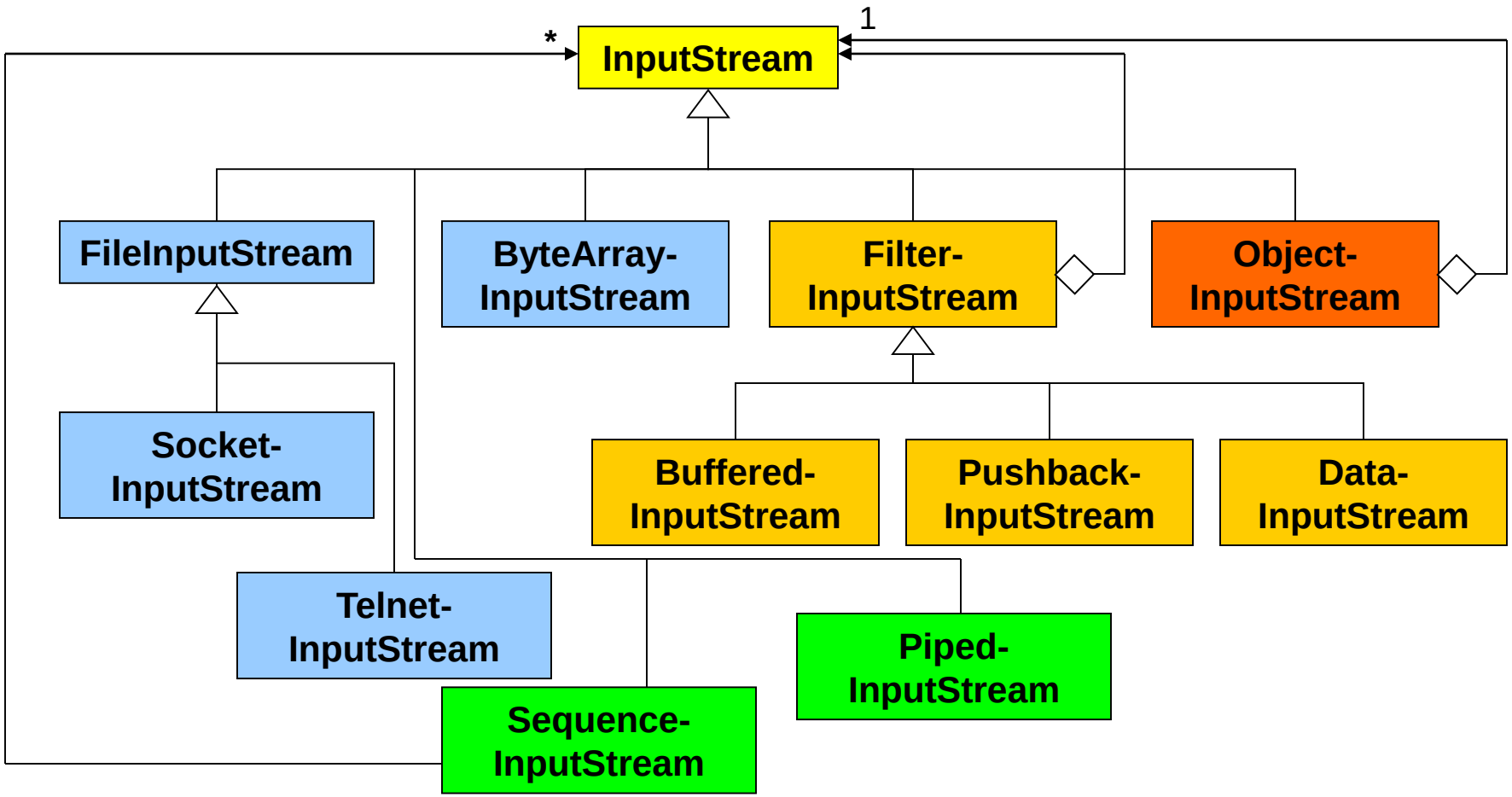

```
abstract class ObjectInputStream {
   public final Object readObject()
       throws IOException,
              ClassNotFoundException,
              OptionalDataException;
   ...
}
```



Decorator Classes

- **ObjectInputStream / ObjectOutputStream**
  - Reads serialized objects
- **DataInputStream / DataOutputStream**
  - Provides data access methods
- **InflaterInputStream / DeflaterOutputStream**
  - Uncompressing / compressing data (GZIP, Zip)
  - GZIPInput/OutputStream and ZipInput/OutputStream
- **DigestInputStream / DigestOutputStream**
  - Computes a message digest while reading / writing (SHA-1/SHA-256)
- **CipherInputStream / CipherOutputStream**
  - Data is decrypted or encrypted upon reading / writing
- **CheckedInputStream / CheckedOutputStream**
  - Stream which computes a check sum (CRC32, Adler32 => long)
- **BufferedInputStream / BufferedOutputStream**
  - Stream which supports mark/unread and which keeps data in a buffer

Whole Picture: InputStreams



Whole Picture: OutputStreams