1. 標準的輸入輸出流：

*System.in:标准的输入流，默认从键盘输入  
System.out:标准的输出流，默认从控制台输出*

修改默認的輸入和輸出行為：

*System类的setIn(InputStream is) / setOut(PrintStream ps)方式重新指定输入和输出的流。*

1. 打印流：

*PrintStream 和PrintWriter*

說明：

*提供了一系列重载的print()和println()方法，用于多种数据类型的输出*

*System.out返回的是PrintStream的實例*

1. 數據流

*DataInputStream 和 DataOutputStream*

作用：

*用于读取或写出基本数据类型的变量或字符串*

示例代碼：

*/\**

*练习：将内存中的字符串、基本数据类型的变量写出到文件中。  
注意：处理异常的话，仍然应该使用try-catch-finally.  
 \*/*@Test  
public void test3() throws IOException {  
 *//1.* DataOutputStream dos = new DataOutputStream(new FileOutputStream("data.txt"));  
 *//2.* dos.writeUTF("刘建辰");  
 dos.flush();*//刷新操作，将内存中的数据写入文件* dos.writeInt(23);  
 dos.flush();  
 dos.writeBoolean(true);  
 dos.flush();  
 *//3.* dos.close();  
}

*/\*  
将文件中存储的基本数据类型变量和字符串读取到内存中，保存在变量中。  
  
注意点：读取不同类型的数据的顺序要与当初写入文件时，保存的数据的顺序一致！  
  
 \*/*@Test  
public void test4() throws IOException {  
 *//1.* DataInputStream dis = new DataInputStream(new FileInputStream("data.txt"));  
 *//2.* String name = dis.readUTF();  
 int age = dis.readInt();  
 boolean isMale = dis.readBoolean();  
  
 System.*out*.println("name = " + name);  
 System.*out*.println("age = " + age);  
 System.*out*.println("isMale = " + isMale);  
  
 *//3.* dis.close();  
  
}