**package** javaIO2;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.net.URL;

**public** **class** DowlandFromInet {

**public** **static** **void** main (String[] args) **throws** IOException {

URL url = **new** URL("http://www.google.com");

InputStream in = url.openStream();

**int** b;

**while**(( b = in.read()) != -1) {

System.***out***.println((**char**) b);

}

in.close();

}

}

1. Не пишите в консоль, а создайте две копии этой страницы (("C:/Users/JavaUser/Videos/page0.html", (("C:/Users/JavaUser/Videos/page1.html") в файловой системе (читайте из одного InputStream, пишите в оба OutputStream);
2. Гарантированно закройте потоки ввода и вывода (в finally);
3. Читайте/ пишите массивами;
4. Добавьте корректную обработку IOException ( в любом случае 3 потока должны быть закрыты), в случае сбоя в записи – удалите недописанные записи и выкиньте исключение (throw new IOEception(“…”, e);

**package** javaIO2;

**import** java.io.BufferedInputStream;

**import** java.io.BufferedOutputStream;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.OutputStream;

**import** java.net.URL;

**public** **class** DowlandFromInet {

**public** **static** **void** main (String[] args) **throws** IOException {

InputStream in = **null**;

OutputStream outPage0 = **null**;

OutputStream outPage1 = **null**;

**try** {

in = **new** URL("http://www.google.com").openStream();

outPage0 = **new** FileOutputStream ("C:/Users/JavaUser/Videos/page0.html");

outPage1 = **new** FileOutputStream ("C:/Users/JavaUser/Videos/page1.html");

*copy*(in, outPage0, outPage1);

} **catch** (IOException e) {

**new** File("C:/Users/JavaUser/Videos/page0.html").delete();

**new** File("C:/Users/JavaUser/Videos/page1.html").delete();

**throw** **new** IOException("Write error occurred", e);

}

**finally** {

**if** (in != **null**) {

**try** {

in.close();

} **catch** (IOException e) {

System.***err***.println("InputStream tread has not been closed: " + e);

}

} **if** (outPage0 != **null**) {

**try** {

outPage0.close();

} **catch** (IOException e) {

System.***err***.println("OutputStream tread has not been closed: " + e);

}

} **if** (outPage1 != **null**) {

**try** {

outPage1.close();

} **catch** (IOException e) {

System.***err***.println("OutputStream tread has not been closed: " + e);

}

}

} // finally

} // main (String[]) method

**private** **static** **void** copy(InputStream in, OutputStream outPage0, OutputStream outPage1) **throws** IOException {

**byte**[] buff = **new** **byte**[64];

**int** count;

**while** ((count = in.read(buff)) != -1) {

outPage0.write(buff, 0, count);

outPage0.flush();

outPage1.write(buff, 0, count);

outPage1.flush();

}

} // copy(InputStream, OutputStream, OutputStream) method

} // DowlandFromInet class