```
package com.sxt.Io02;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.Closeable;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
/**
 *
 * @author DELL
 *
 */
public class FileUtils {
	public static void main(String[] args) {
		//文件到文件
		try {
				InputStream is=new
FileInputStream("abc.txt");
				OutputStream os=new
FileOutputStream("abcd.txt");
				copy(is,os);
			} catch (FileNotFoundException e) {
				e.printStackTrace();
			}

		//文件到字节数组
		byte[] datas=null;
```

```
            try {
                                InputStream is=new
FileInputStream("abc.txt");
                                ByteArrayOutputStream baos=new
ByteArrayOutputStream();
                                copy(is,baos);
                                datas=baos.toByteArray();
                                System.out.println(datas.length);
                        } catch (FileNotFoundException e) {
                                e.printStackTrace();
                        }
            //字节数组到文件
            try {
                        InputStream is=new ByteArrayInputStream(datas);
                                OutputStream os=new
FileOutputStream("abcd.txt",true);
                                copy(is,os);
                        } catch (FileNotFoundException e) {
                                e.printStackTrace();
                        }


        /**
                         * 对接输入输出流
          */


            }
        public static void copy(InputStream is,OutputStream os) {
            byte[] flush=new byte[1024];   //缓冲容器
```

```
        int len=-1;   //接收长度
        try {
                        while((len=is.read(flush))!=-1) {
                                os.write(flush,0,len);
                        }
                        os.flush();
                } catch (IOException e) {
                        e.printStackTrace();
                }finally {
                        close(is,os);
                }
}
/**
                        *   释放资源
  */
public static void close(InputStream is,OutputStream os) {
        try {
                        if(null!=os) {
                        os.close();
                        }
                } catch (IOException e) {
                        e.printStackTrace();
                }
                try {
                        if(null!=os) {
                        os.close();
                        }
                } catch (IOException e) {
                        e.printStackTrace();
                }
```

```
    }
    /**
     * 释放资源(最好用这种)
     */
    public static void close(Closeable...ios) {
        for(Closeable io:ios) {
            try {
                if(null!=io) {
                    io.close();
                }
                } catch (IOException e) {
                    e.printStackTrace();
                }
        }
    }

}
```