IO

输入流

Input 从磁盘读文件，者网落读取信息，控制台输入内容

FileInputStream 字节流

```
package com.hqyj;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;

public class FileInputStreamDemo {

    public static void main(String[] args) {
        FileInputStream inputStream = null;
        try {
            //创建FileInputStream对象，参数可以是文件名或者File对象
            inputStream = new FileInputStream("d:\\test.txt");
            byte[] b = new byte[512];
            //调用read读取流里面的内容，参数是一个byte数组，数组的大小根据文件大小而定，是字节的整数倍。
            //返回的值就是读取的长度，能读到值是大于0
            while(inputStream.read(b)>0) {
                System.out.println(new String(b));
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }finally {
            try {
                //关闭input流
                inputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

    }

}
```

FileReader 字符流

```
package com.hqyj;

import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

public class FileReaderDemo {

    public static void main(String[] args) {
        FileReader reader = null;
        try {
            reader = new FileReader("d:\\test.txt");
            char[] cbuf = new char[1024];
            while(reader.read(cbuf)>0) {
                System.out.println(cbuf);
            }

        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }finally {
            try {
                reader.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

}
```

BufferedReader 带有缓存，可以按行读

```
package com.hqyj;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class BufferedReaderDemo {

    public static void main(String[] args) {
        FileReader reader = null;
        BufferedReader bufferedReader = null;
        try {
            //实例化一个reader对象，用于创建BufferedReader的参数
            reader = new FileReader("d:\\test.txt");
            bufferedReader = new BufferedReader(reader);
            String line = null;
            while((line = bufferedReader.readLine())!= null) {
                System.out.println(line);
            }
```

```
        } catch (Exception e) {
            e.printStackTrace();
        }finally {
            try {
                bufferedReader.close();
                reader.close();
            } catch (IOException e) {
                e.printStackTrace();
            }

        }


    }

}
```

输出流

Output 把内容写到文件，发送给网络其它程序，输出控制台

FileOutputStream 字节流

```
package com.hqyj;

import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;

public class FileOutputStreamDemo {

    public static void main(String[] args) {
        FileOutputStream outputStream = null;
        try {
            //创建输出流对象，参数可以是文件名，或者File
            outputStream = new FileOutputStream("d:\\out.txt");
            byte[] b = new byte[512];
            String s = "FileOutputStream test 测试";
            b = s.getBytes();
            //输出内容到流
            outputStream.write(b);
//          outputStream.write("FileOutputStream test".getBytes());

        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }finally {
            try {
                //关闭流
                outputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
```

```
            }
        }

    }

}
```

FileWriter 字符流

```
package com.hqyj;

import java.io.FileWriter;
import java.io.IOException;

public class FileWriterDemo {

    public static void main(String[] args) {
        FileWriter writer = null;

        try {
            writer = new FileWriter("d:\\out.txt");
//          writer.write("This is a test. 这是一个测试");
            writer.append("调用append方法\n");
            writer.append("第二次调用append方法");
        } catch (IOException e) {
            e.printStackTrace();
        }finally {
            try {
                writer.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

    }

}
```

BufferedWriter 带有缓存，按行写文件

```
package com.hqyj;

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;

public class BufferedWriterDemo {

    public static void main(String[] args) {
        FileWriter writer = null;
        BufferedWriter bufferedWriter = null;
        try {
            writer = new FileWriter("d:\\out.txt");
            bufferedWriter = new BufferedWriter(writer);
```

```
            bufferedWriter.append("第一行");
            bufferedWriter.newLine();  //换行
            bufferedWriter.append("第二行");
            bufferedWriter.flush(); //刷新缓存（强制写磁盘）
        } catch (IOException e) {
            e.printStackTrace();
        }finally {
            try {
                bufferedWriter.close();
                writer.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }


    }

}
```

PrintWriter 提供一系列print输出内容

```
package com.hqyj;

import java.io.FileNotFoundException;
import java.io.PrintWriter;

public class PrintWriterDemo {

    public static void main(String[] args) {
        PrintWriter writer = null;
        try {
            writer = new PrintWriter("d:\\out.txt");
            writer.println("第一行");
            writer.println("第二行");
            writer.println("第三行");
            writer.flush();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }finally {
            writer.close();
        }

    }

}
```

序列化

把对象保存为文件叫序列化，读取序列化文件重新还原为对象叫反序列化。

用于休眠，网络传输对象。

被序列化的对象必须要实现Serializable接口，实现后需要生成序列化版本号，否则报警告

创建一个Student类，实现序列化接口

```
package com.hqyj;

import java.io.Serializable;

public class Student implements Serializable{
    /**
     * 序列化的版本号
     */
    private static final long serialVersionUID = -5274887811639087134L;
    private int stuNum;
    private String stuName;

    public Student() {
        super();
    }
    public Student(int stuNum, String stuName) {
        super();
        this.stuNum = stuNum;
        this.stuName = stuName;
    }
    public int getStuNum() {
        return stuNum;
    }
    public void setStuNum(int stuNum) {
        this.stuNum = stuNum;
    }
    public String getStuName() {
        return stuName;
    }
    public void setStuName(String stuName) {
        this.stuName = stuName;
    }
    @Override
    public String toString() {
        return "Student [stuNum=" + stuNum + ", stuName=" + stuName + "]";
    }


}
```

序列化和反序列化代码

```
package com.hqyj;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;

public class SerializeDemo {
```

```
    public static void main(String[] args) {
        //序列化（把一个对象保存为文件）。 如果对未实现序列化接口的对象进行序列化操作，会报异常NotSerializableException

        Student student = new Student(8888, "迈克");
        System.out.println(student);
        ObjectOutputStream outputStream = null;
        try {
            outputStream = new ObjectOutputStream(new FileOutputStream("d:\\student.obj"));
            outputStream.writeObject(student);

        } catch (IOException e) {
            e.printStackTrace();
        }finally {
            try {
                outputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }



        //反序列化（读之前序列化文件，重写生成对象）
        /*
        ObjectInputStream objectInputStream = null;
        try {
            objectInputStream = new ObjectInputStream(new FileInputStream("d:\\student.obj"));
            Object object = objectInputStream.readObject();
            Student stu = (Student) object;
            System.out.println(stu);
        } catch (Exception e) {
            e.printStackTrace();
        }finally {
            try {
                objectInputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        */


    }

}
```

今日作业

1. 在a目录下存一张图片。功能：拷贝图片文件到b目录
2. 读入一个文本文件，统计这个文件中A字符出现次数
3. 写一个文本文件，内容为一首诗（一首歌词）