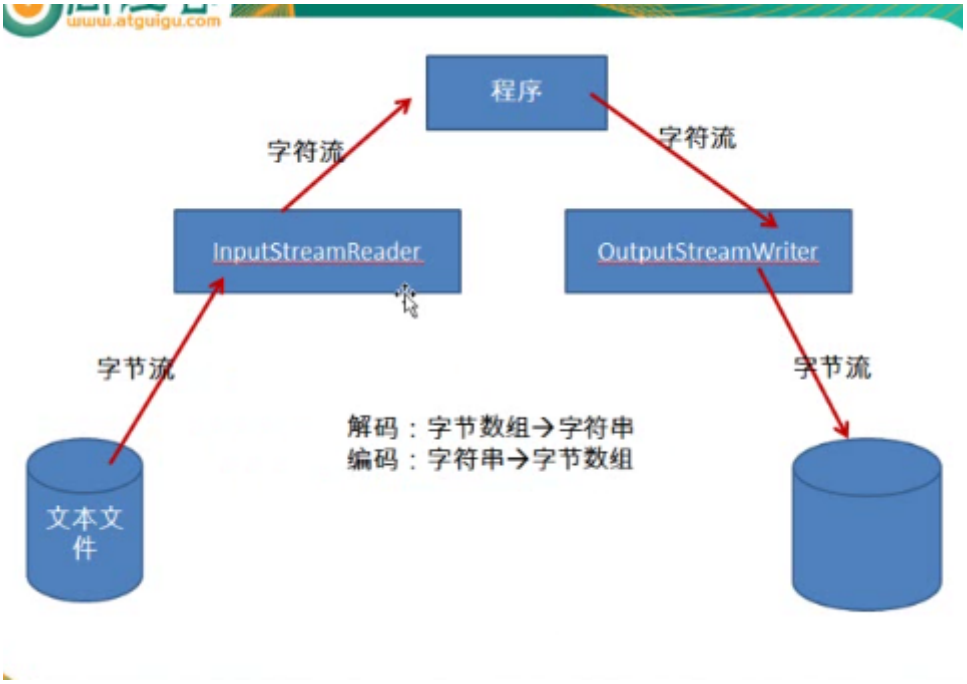
复制文本文件的时候，应该使用字节流进行复制FileInputStream 和FileOutputStream

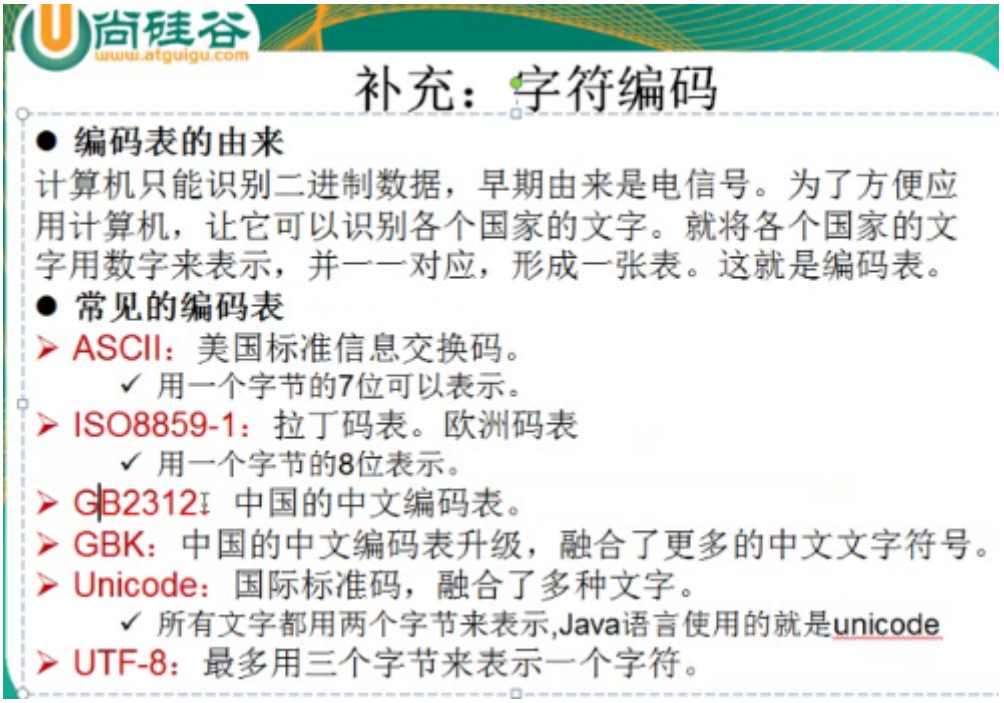
二，转换流：转换流提供了在字节流与字符流之间的转换；java api提供了俩个转换流；inputstreamreader和outputstreamwriter，字节流中的数据都是字符时，转成字符流操作效率更高

转换流：InputStreamReader和OutputStreamWriter

编码：字符流----->字节数组

解码：字节数组----->字符串





标准的输入流：system.out
标准的输入流：system.in

一，复习：
1，java.io包下，file类，java程序中的此类的一个对象，就对应着硬盘中的一个文件或者网络中的一个资源
File file1 = new File（“e：//hello.txt”）；
File file2 = new File（“e：//hello”）；
①file既可以表示一个文件（.doc .txt .mp3 .avi .dat）,也可以表示一个文件目录
②file的对象与平台无关
③file类针对于文件或者文件目录，只能进行新建，删除，重命名，上层目录等等的操作，如果涉及访问文件中的内容，file类是无能为力的，只能使用io流下提供的相应的输入输出流来实现
④常常把file类的对象作为参数传递给相应的输入输出流的对象的构造器中

2，IO流的结构
抽象基类：四个

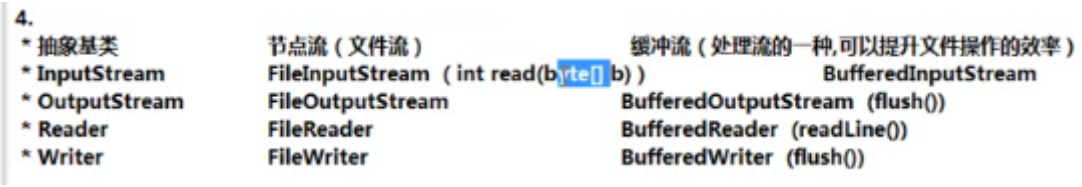
分类	字节输入流	字节输出流	字符输入流	字符输出流
抽象基类	InputStream	OutputStream	Reader	Writer
访问文件	FileInputStream	FileOutputStream	FileReader	FileWriter
访问数组	ByteArrayInputStream	ByteArrayOutputStream	CharArrayReader	CharArrayWriter
访问管道	PipedInputStream	PipedOutputStream	PipedReader	PipedWriter
访问字符串			StringReader	StringWriter
缓冲流	BufferedInputStream	BufferedOuputStream	BufferedReader	BufferedWriter
转换流			InputStreamReader	OutputStreamWriter
对象流	ObjectInputStream	ObjectOutputStream		
	FilterInputStream	FilterOutputStream	FilterReader	FilterWriter
打印流		PrintStream		PrintWriter
推回输入流	PushbackInputStream		PushbackReader	
特殊流	DataInputStream	DataOutputStream		

3，IO流的划分

1）按照流的流向的不同，输入流和输出流，站在程序的角度

2）按照流中的数据单位的不同，字节流与字符流（纯文本文件使用字符流，除此之外使用字节流）

3）按照流的角色的不同，节点流（流直接作用于文件上除此之外都是处理流）和处理流



注意：①从硬盘中读入一个文件，要求文件一定得存在，报FileNotFoundException的异常

②从程序中输出一个文件到硬盘，此文件可以不存在，若不存在，就创建一个实现输出，则将

一 存在的文件覆盖

③真正开发时，就使用缓冲流来代替节点流

④主要最后要关闭相应的文件流，先关闭输出流，在关闭输入流，将此放在finally中

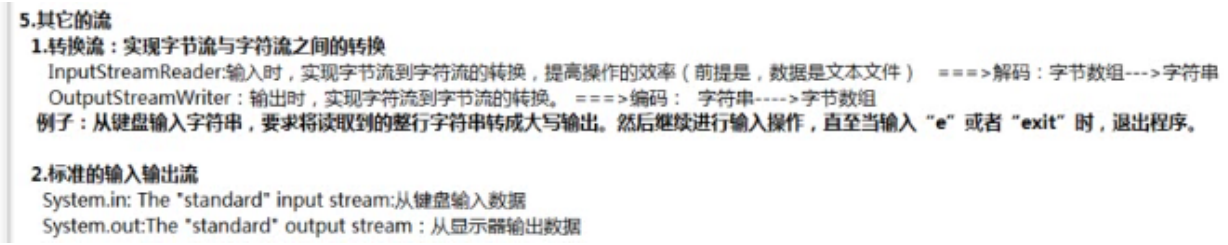
4，其他的文件

1，转换流：实现字节流与字符流之间的转换

InputStreamReader输入时，实现字节流到字符流的转换，提高操作的效率（前提是，数据是文

本文件）----->解码字节数组---->字符串

OutputStreamWriter输出时，实现字符流到字节流的转换，----->编码（字符串-----》字节数组）



//字符流复制文件

```
        BufferedReader bReader = null;
        BufferedWriter bWriter = null;
        try {
                bReader = new BufferedReader(new FileReader(new
File("C://Users//Anly//Desktop//1.txt")));
                bWriter = new BufferedWriter(new FileWriter(new
File("C://Users//Anly//Desktop//55.txt")));
                char b[] = new char[1024];
                int len;
                while((len = bReader .read(b)) != -1){
                        bWriter .write(b,0,len);
                }

        } catch (Exception e) {
                // TODO: handle exception
                e.printStackTrace();
        }
        finally {
                if (bWriter != null) {
```

```
            try {
                    bWriter.close();
            } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
            }
    }
    if (bReader != null) {
            try {
                    bReader.close();
            } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
            }
    }
```