RandomAccessFile:

File类不能操作文件的内容，如果想操作文件的内容，需要用一个RandomAccessFile类

此类的实例支持随机访问文件，对文件进行读取和写入操作

此类基于指针的操作，总是在当前位置读写子接

两种模式：“rw”: 可读可写，”r“：只读

两个构造方法：

RandomAccessFile(String path, String mode)

RandomAccessFile(File file, String mode)

该类方法的读和写都是参照虚拟内存而言的

虚拟内存（程序）🡨读入—物理内存（文件）

虚拟内存（程序）--写入🡪物理内存（文件）

Write():写入一个字节，写入低八位-void—写的是当前指针指向的字节，写完后指针后移到下一位

Read(): 读取一个字节，读取低八位-int—读的是当前指针指向的字节—返回值为读回得字节数，存入得是int类型得低八位上：

--一次读一个字节，读完后指针自动后移到下一个字节处，如果返回值为-1表示没有数据可读

底层是封装了两个流—输入和输出

RandomAccessFile向一个文件中写入一个字节

1. 如果写入得字节数大于原文件中的字节数，则进行全部覆盖。
2. 如果写入的字节数小于原文件中的字节数，则进行写入字节数的等量覆盖

文件的复制粘贴

1. 一个一个字节的复制
2. 使用一个字节的数组

写入字符串，

字符变成字节—写入

将当前的字符串按照系统默认的字符集编码转换成对应的字节，返回值为对应的字节数组

重载的方法：

String的getBytes(String charset)：将当前的字符串按照指定的字符集编码转换成对应的字节返回值为对应的字节数组

字符集编码：256\*256 = 65536 256

1. gbk：英文一个字节表示一个字符，中文两个字节表示一个字符
2. utf-8：万国码，英文一个字节表示一个字符，中文是三个字节表示一个字符（少数的两字节，四个字节）
3. gb2312：繁体字
4. Unicode：一个字符一个字节—只服务于英文

RandomAccessFile：这个类是基于指针的操作

getFilePointer():获取指针的当前位置（返回值为long类型）在该位置发生读取或写入操作。

Seek(Long l)：调整指针的当前位置，设置到此文件开头的测量位置的偏移量，在该位置发生读取或者写入操作，无返回值，可以任意位置偏移（前后都可以偏移），偏移量的是从0开始

表示指针调整字节为2的位置

SkipByte(int n) 尝试跳过输入的n个字节，以丢弃跳过的字节、只能往后跳，不能往前跳，使用频率较低