常用类 String

charAt 返回第i个字符

concat 在尾部添加字符串

equals 比较两个字符串是否一样

indexOf 返回一个字符第一次出现的位置

startsWith 是否以某个字符串为前缀开始

toCharArray() 转换为一个数组

trim() 清除前后的空格

```
public static void main(String[] args) {
        String str = new String("HelloWorld");
        System.out.println(str.charAt(1));
        System.out.println(str.concat("nihao"));
        System.out.println(str.equals("nihao"));
        System.out.println(str.indexOf('l', 5));
        System.out.println(str.startsWith("Hel"));
        char[] ch = str.toCharArray();
        System.out.println(ch[0]);
        str = "  Hello   ";
        System.out.println(str);
        System.out.println(str.trim());
    }
```

StringBuffer

--1.3.1StringBuffer类的常用方法：

```
1.StringBuffer sb=new StringBuffer();
创建一个空的StringBuffer对象
2.StringBuffer sb=new StringBuffer(String str);
接收一个String内容并变为StringBuffer内容
3.public StringBuffer append(数据类型 变量)
  sb.append(String str1);
对StringBuffer内容进行连接
4.public StringBuffer insert(索引，数据类型 变量)
    sb.insert(int insert,String str2);
在指定索引位置插入数据
5.public StringBuffer delete(起始索引，终点索引)
    sb.delete(0,2);
删除指定索引范围的数据
6.publlic StringBuffer reverse()
  sb.reverse();
内容翻转
```

Math 和 Random

--3.1Math类

```
Math.abs(-10)   // 绝对值

*Math.sqrt(X)//计算平方根
Math.cbrt(X)//计算立方根
*Math.pow(a, b)//计算a的b次方
Math.max( 1,2 );//计算最大值
*Math.min( 3,4 );//计算最小值
Math.ceil(X) 接近此数的大的整数的值
   Math.floor(X) 接近此数的小的整数的值

Math.random()   [0,1)


Math.round(X)         round 四舍五入，float时返回int值，double时返回long值
```

--3.2Random类

Random()
创建一个新的随机数生成器。

Random(long seed)
使用单个 long 种子创建一个新的随机数生成器。

```java
Random random = new Random();
for (int i = 0; i < 10; i ++ ) {
    System.out.println(random.nextInt(11)); // [0, 10]
}
```

包装类

基本数据类型	包装类型
byte	Byte
boolean	Boolean
short	Short
char	Character
int	Integer
long	Long
float	Float
double	Double

--自动装箱

自动装箱即自动将基本数据类型转换成包装类型，在 Java 5 之前，要将基本数据类型转换成包装类型只能这样做，看下面的代码。

```java
//早期基本数据类型->包装类型

Integer i1 = new Integer(8);  //int->Integer

Integer i2 = Integer.valueOf(8); //int->Integer

//现在是自动装箱(自动基本数据类型->包装类型转换)
Integer i3 = 8;//8是int类型
```

--自动拆箱

自动拆箱即自动将包装类型转换成基本数据类型，与自动装箱相反，有装就有拆，很好理解。

```java
//早期包装数据类型->基本类型转换
Float i=3.3;
float i5 = i.floatValue();

Integer i2=3;
int m=i2.intValue();

// 自动拆箱(自动包装数据类型->基本类型转换)
Integer i=3;
int i4 = i;
```

--日期与时间类

集合

List接口

> *继承了Collection的所有,又增加了一些特有*
>
> *常用方法:*
>
> *。public void add(int index, E element) : 将指定的元素，添加到该集合中的指定位置上*
>
> *。public E get(int index) :返回集合中指定位置的元素*
>
> *。public E remove(int index) : 移除列表中指定位置的元素, 返回的是*被移除的元素**
>
> *。public E set(int index, E element) :用指定元素替换集合中指定位置的元素,*返回值的更新 前的元素**。
>
> *其子类 ArrayList Vector LinkList*

多线程

```
sleep()方法  :sleep()使当前线程进入阻塞状态，在指定时间内不会执行。自动苏醒
wait()方法: 不会主动苏醒，需要另外的线程调用notify()/notifyAll()方法唤醒。
t.join()方法只会使主线程(或者说调用t.join()的线程)进入等待池并等待t线程执行完毕后才会被唤醒
interrupt()的作用是中断本线程。
notify和notifyAll来唤醒在某些条件下等待的线程
      sleep()方法和wait()方法
 相同点：一旦执行方法以后，都会使得当前的进程进入阻塞状态。
 不同点：
 1.方法声明方式不同，sleep()方法在Thread类中声明，wait()方法在Object类中声明。
 2.调用的条件不同，sleep可以在任何需要的场景下调用，wait必须使用在同步代码块或者同步方法中。
 3.是否释放锁，sleep不会释放，wait会释放
```

```
public class Testing {
    public static void main(String[] args) {
        Thread1 thread1 = new Thread1();
        Thread2 thread2 = new Thread2();
        Thread thread = new Thread(thread2);
        thread1.start();
        thread.start();
        for (int i = 0; i < 10; i ++ ) {
            System.out.println("main_thread" + " i=" + i);
        }
    }
}

class Thread1 extends Thread {
```

```
    @Override
    public void run() {
        try {
            Thread1.sleep(5000);
        } catch(InterruptedException e) {
            e.printStackTrace();
        }

        for (int i = 0; i < 10; i ++ ) {
            System.out.println(Thread.currentThread().getName() + " i=" + i);
        }
    }
}

class Thread2 implements Runnable {
    @Override
    public void run() {
        for (int i = 0; i < 10; i ++ ) {
            System.out.println(Thread.currentThread().getName() + " i=" + i);
        }
    }
}
```

IO输入输出

最重要的就是5个类和一个接口。5个类指的是File、OutputStream、InputStream、Writer、Reader；一个接口指的是Serializable

主要的类如下：

1. File（文件特征与管理）：用于文件或者目录的描述信息，例如生成新目录，修改文件名，删除文件，判断文件所在路径等。
2. InputStream（二进制格式操作）：抽象类，基于字节的输入操作，是所有输入流的父类。定义了所有输入流都具有的共同征。
3. OutputStream（二进制格式操作）：抽象类。基于字节的输出操作。是所有输出流的父类。定义了所有输出流都具有的共同征。
4. Reader（文件格式操作）：抽象类，基于字符的输入操作。
5. Writer（文件格式操作）：抽象类，基于字符的输出操作。

IO流的分类
- 根据处理数据类型的不同分为：字符流和字节流
  字符流的由来： Java中字符是采用Unicode标准，即一个字符使用两个字节来表示，本质其实就是基于字节流读取时
- 根据数据流向不同分为：输入流和输出流

Reader 和 writer 是转么用来读取字符的，避免了很多的字符乱码问题，读取字符十分的方便。但是不能像InputStream 和 OutputStream 用于读取图片，视频等，只能用于读取字符。

```
Reader reader = new FileReader(url);
 XX.read(char[],int off,int len)

 char[] 目的缓冲区   off开始存储的下标   len读取最大字符数
 XX .read(char[]) 读取到数组里
 XX.close()

 Writer writer = new FileWriter(url); //尾部追加 ,true
  Writer 是把数据写入到内存中，要使用flush刷新内存中的数据到硬盘

url : 写入的路径
true :  append 允许向后添加写入。不覆盖

 1.write(char[] chars, int off, int len) 写到内存
 2.flush() 刷到硬盘
 3.close() 关闭流.
```

```
import java.io.*;
```

```
public static void main(String[] args) throws IOException {
        Reader reader = new FileReader("D:\\document\\text1.txt");
        Writer writer = new FileWriter("D:\\document\\text2.txt", true);  // 这个
true是append
        char[] cpy = new char[1024];
        int len = 0;
        while ((len = reader.read(cpy)) != -1) {
            writer.write(cpy, 0, len);
            writer.flush();
        }

        reader.close();
        writer.close();
    }
```

字节流读取

FileInputStream read

FileOutputStream write

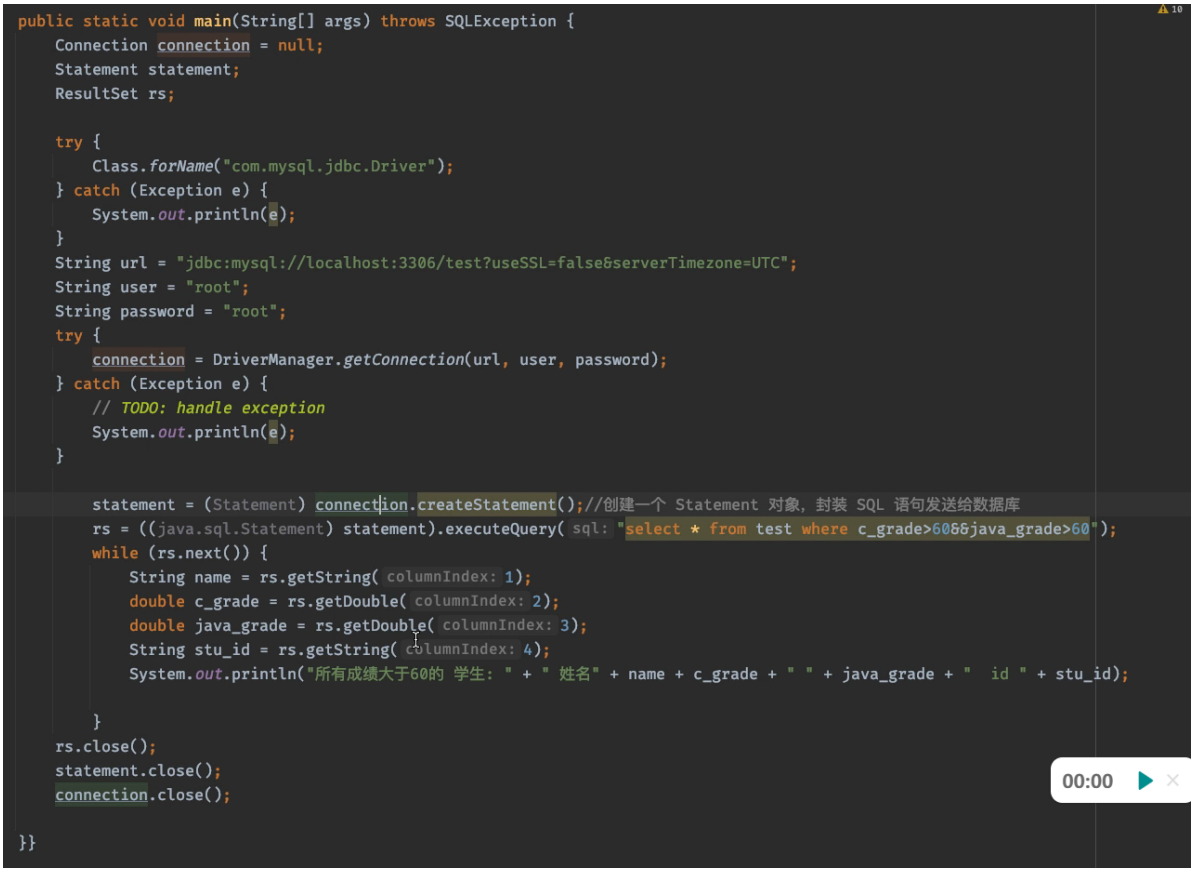
```
InputStream is = new FileInputStream("D:\\document\\text1.txt");
        OutputStream os = new FileOutputStream("D:\\document\\text2.txt", true);
        byte[] bytes = new byte[1024];
        int len = 0;
        while ((len = is.read(bytes)) != -1) {
            os.write(bytes, 0, len);
            os.flush();
        }

        is.close();
        os.close();
```

File类
    创建
   File f1=new File("路径://FileTest1.txt")
    File对象一经创建， 就可以通过调用它的方法来获得文件或目录的属性。
      1) public boolean exists( ) 判断文件或目录是否存在
       2) public boolean isFile( ) 判断是文件还是目录
       3) public boolean isDirectory( ) 判断是文件还是目录
       4) public String getName( ) 返回文件名或目录名
       5) public String getPath( ) 返回文件或目录的路径。
       6) public long length( ) 获取文件的长度
       7) public String[ ] list ( ) 将目录中所有文件名保存在字符串数组中返回。
       File类中还定义了一些对文件或目录进行管理、操作的方法，常用的方法有:
       1) public boolean renameTo(File newFile);   重命名文件
       2) public void delete( );   删除文件
       3)   public boolean mkdir( ); 创建目录

```
File file = new File("D:\\document\\text3.txt");
        System.out.println(file.exists());
        System.out.println(file.getName());
        System.out.println(file.mkdir());
        System.out.println(file.isDirectory());
```

JDBC



```
public static void main(String[] args) throws SQLException {
    Connection connection = null;
    Statement statement;
    ResultSet rs;

    try {
        Class.forName("com.mysql.jdbc.Driver");
    } catch (Exception e) {
        System.out.println(e);
    }
    String url = "jdbc:mysql://localhost:3306/test?useSSL=false&serverTimezone=UTC";
    String user = "root";
    String password = "root";
    try {
        connection = DriverManager.getConnection(url, user, password);
    } catch (Exception e) {
        // TODO: handle exception
        System.out.println(e);
    }

        statement = (Statement) connection.createStatement();//创建一个 Statement 对象，封装 SQL 语句发送给数据库
        rs = ((java.sql.Statement) statement).executeQuery( sql: "select * from test where c_grade>60&&java_grade>60");
        while (rs.next()) {
            String name = rs.getString( columnIndex: 1);
            double c_grade = rs.getDouble( columnIndex: 2);
            double java_grade = rs.getDouble( columnIndex: 3);
            String stu_id = rs.getString( columnIndex: 4);
            System.out.println("所有成绩大于60的 学生: " + " 姓名" + name + c_grade + " " + java_grade + "  id " + stu_id);

        }
    rs.close();
    statement.close();
    connection.close();

}}
```