第5章 Java API

老师：李沁洳

Java API 概述

API (Application Programming Interface): 应用程序编程接口

简单理解： API就是别人已经写好的东西，我们不需要自己编写，直接使用即可。

```
public static void main(String[] args){
    Random r = new Random();
    int number = r.nextInt(100);
}
```

Java API: 指的就是JDK中提供的各种功能的Java类

这些类将底层的实现封装了起来，我们不需要关心这些类是如何实现的，只需要学习这些类如何使用即可。

Java API 概述

已经使用或者学习过的API

Scanner 键盘录入

Random 随机数

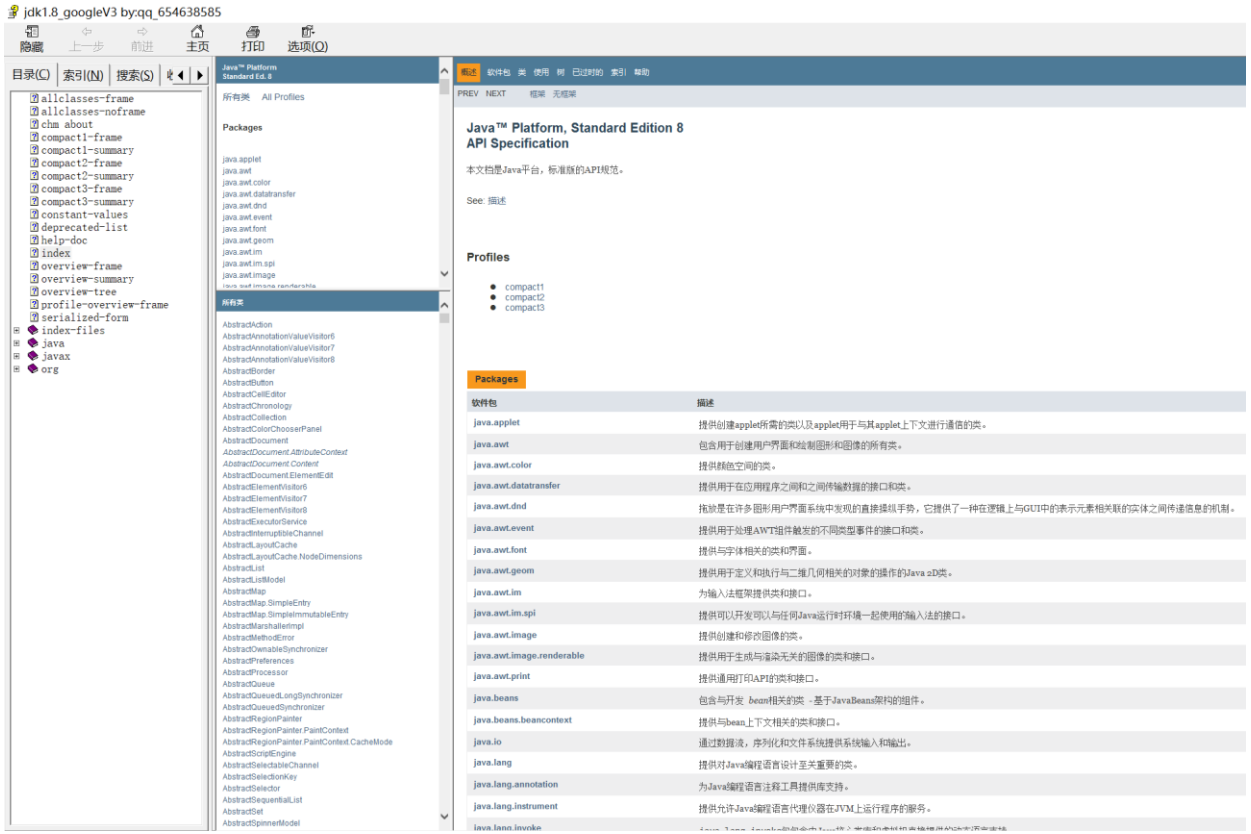
BigDecimal

…

在线文档

https://docs.oracle.com/javase/8/docs/api/

离线文档



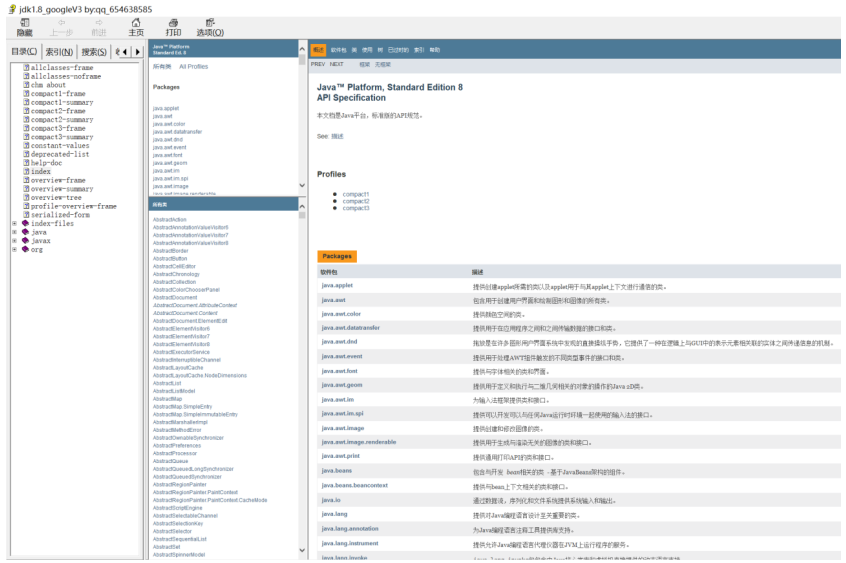
Java API 概述

Java API 和 API帮助文档:

Java API: 指的就是JDK中提供的各种功能的Java类

这些类将底层的实现封装了起来，我们不需要关心这些类是如何实现的，只需要学习这些类如何使用即可。

API帮助文档: 帮助开发人员更好的使用API和查询API的一个工具。



1. 打开AP帮助文档
2. 点击显示，并找到索引下面的输入
3. 在输入框中输入类名并点击显示
4. 查看类所在的包
5. 查看类的描述
6. 查看构造方法
7. 查看成员方法

/5.1

字符串类

5.1 字符串类

“abc”

“123”

“西华师范大学”

“abc” + true = “abctrue”

“123” + “风花雪月” = “123风花雪月”

“西华师范大学” + 666 = “西华师范大学666”

只要有字符串加，他们参与的都是拼接操作。

5.1 字符串类

字符串学习内容

String, StringBuilder, StringJonier, StringBuffer, Pattern, Matcher

可以掌握字符串的一些常见操作了。

实际开发中的一些常见案例

掌握分析问题，解决问题的能力。

字符串相关的底层原理

掌握原理更好的通过面试，处理开发中的一些复杂问题。

5.1 字符串类

Java.lang.String类代表字符串，Java程序中的所有字符串文字（例如“abc”）都为此类的对象。

```
String name = "李沁洳";
String schoolName = "西华师范大学";
```

String的注意点：

字符串的内容是不会发生改变的， 它的对象在创建后不能被改变。

3个

```
String name = "张三";
String schoolName = "西华师大";
System.out.println(name + schoolName);
```

2个

```
String name = "张三";
name = "KunKun";
```

5.1.1 String类的初始化

创建String对象的两种方式：

1. 直接赋值

```
String name = "李沁洳";
String schoolName = "西华师范大学";
```

2. new

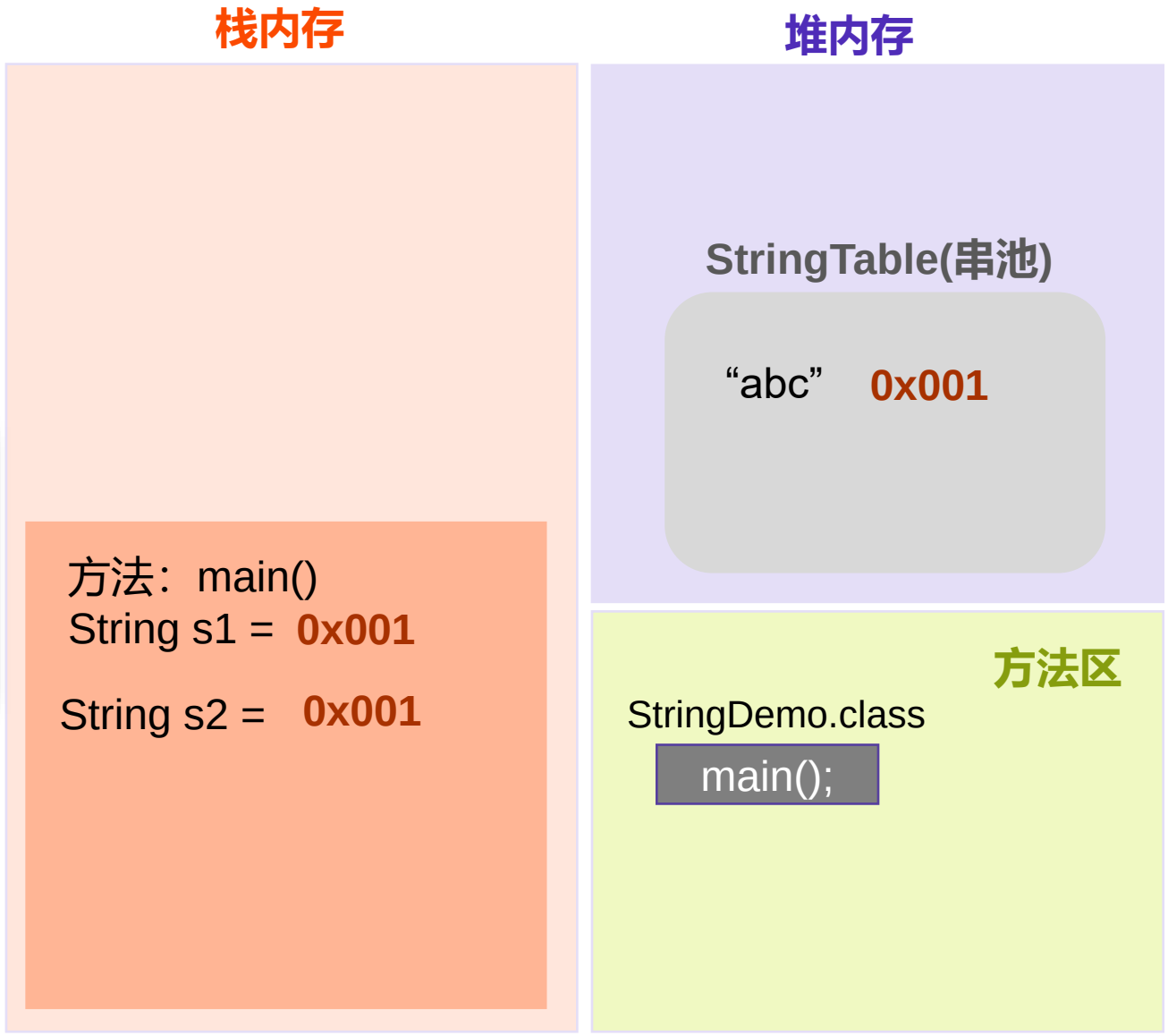
构造方法	说明
public String()	创建空白字符串，不含任何内容
public String(String original)	根据传入的字符串，创建字符串对象
public String(char[ ] chs)	根据字符数组，创建字符串对象
public String(byte[ ] chs)	根据字节数组，创建字符串对象

5.1.1 String类的初始化

Java的内存模型

注意：StringTable(串池)
在JDK7版本开始从方法区中挪到了堆内存

```
public class StringDemo{
    public static void main(String[] args){
        String s1 = "abc";
        String s2 = "abc";
    }
}
```



5.1.1 String类的初始化

```
Public class StringDemo{
    public static void main(String[] args){
        String s1 = "abc";
        String s2 = "abc";
    }
}
```

当使用双引号直接赋值时，系统会检查该字符串在串池中是否存在。

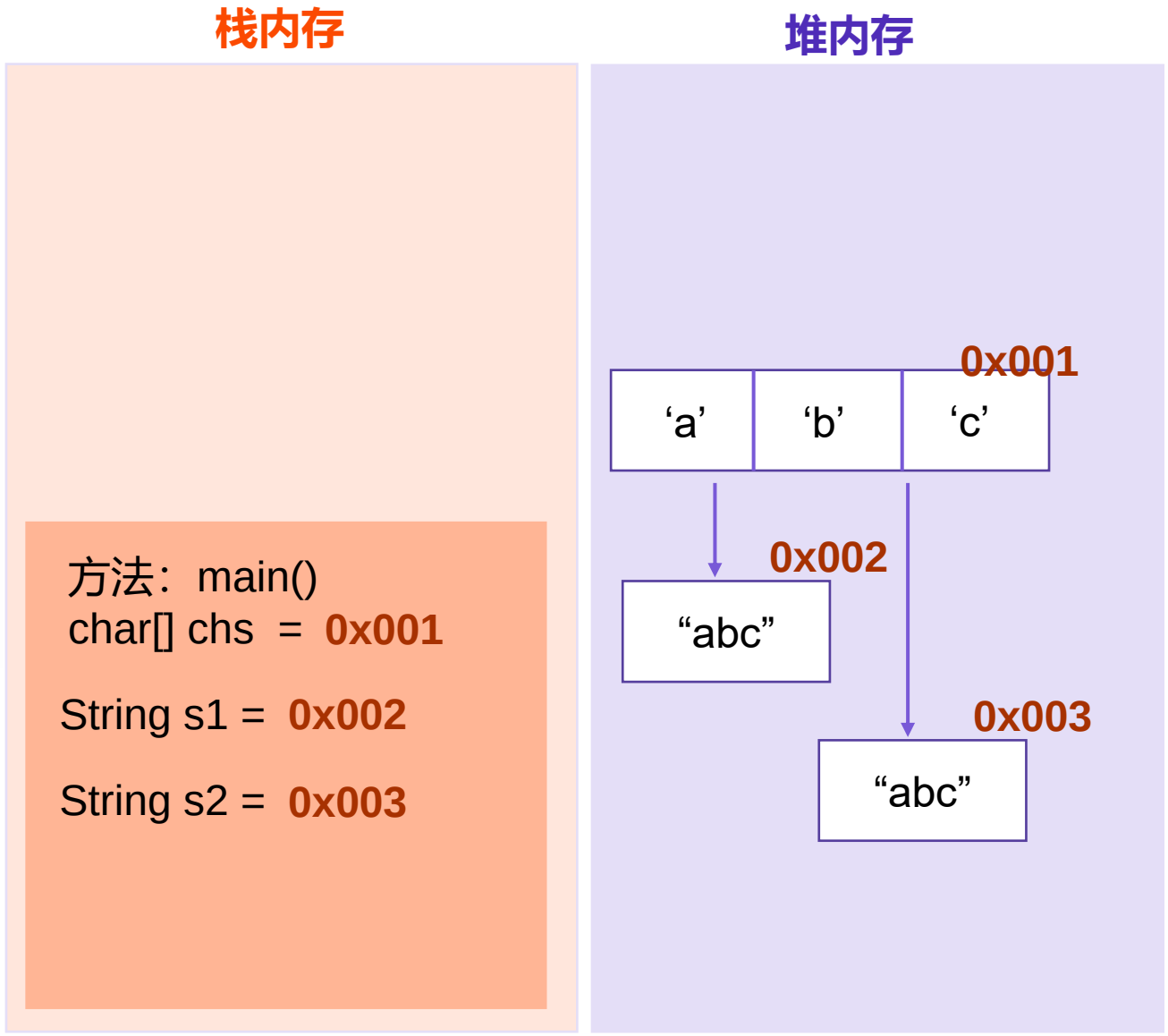
不存在：创建新的

存在：复用

5.1.1 String类的初始化

Java的内存模型

```
public class Test{
    public static void main(String[] args){
        char[] chs = {'a','b','c'};
        String s1 = new String(chs);
        String s2 = new String(chs);
    }
}
```



5.1.2 String类的常见操作

1. 字符串的比较

```
String s1 = "abc";
String s2 = "abc";
System.out.println(s1==s2);
```

true

```
String s3 = new String("abc");
String s4 = new String("abc");
System.out.println(s3==s4);
```

false

```
String s1 = "abc";
String s2 = new String("abc");
System.out.println(s1==s2);
```

false

5.1.2 String类的常见操作

== 号比的到底是什么?

基本数据类型

```
int a = 10;
int b = 20;
System.out.println(a == b);
```

基本数据类型比较的是数据值

引用数据类型

```
String s3 = new String("abc");
String s4 = new String("abc");
System.out.println(s3==s4);
```

引用数据类型比较的是地址值

5.1.2 String类的常见操作

1. 字符串的比较

- boolean equals方法（要比较的字符串）　　完全一样的结果才是true,否则为false
- boolean equalsIgnoreCase(要比较的字符串)　　忽略大小写的比较

两种应用场景：

密码的比较

验证码的比较

5.1.2 String类的常见操作

2. 字符串的获取功能

- public char charAt(int index)； 根据索引返回字符 西华师范大学666
- public int length(); 返回此字符串的长度
- 数组的长度：数组名.length (数组的长度是属性，所以不加小括号)
- 字符串的长度：字符串对象.length() 西华师范大学666

案例：统计字符次数

描述：键盘录入一个字符串，统计该字符串中大写字母字符，小写字母字符，数字字符出现的次数（不考虑其他字符）

5.1.3 StringBuffer类

操作字符串的效率：

类名	优点	缺点
String		内容和长度不可改变，效率最低
StringBuffer	内容和长度可以改变，线程安全	效率中等
StringBuilder	内容和长度可以改变，效率最高	非线程安全

5.1.4 StringBuilder类

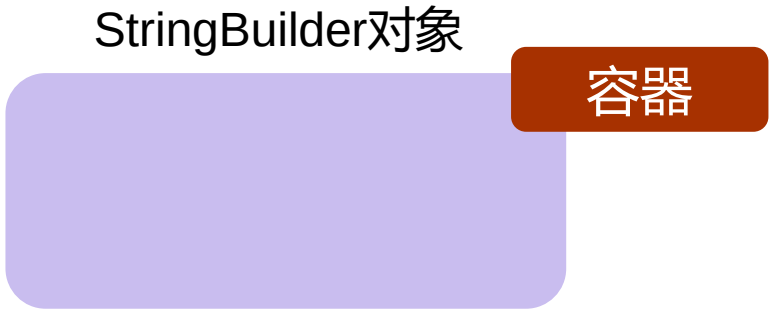
StringBuilder概述

StringBuilder可以堪称是一个容器，创建之后里面的内容是可变的。

- 作用：提高字符串的操作效率

```
String s1 = "aaa";
String s2 = "bbb";
String s3 = "ccc";
String s4 = "ddd";
String s5 = "eee";
```

```
String s6 = s1 + s2 + s3 + s4 + s5;
```



5.1.4 StringBuilder类

StringBuilder构造方法

方法名	说明
public StringBuilder()	创建一个空白可变字符串对象，不含任何内容
public StringBuilder(String str)	根据字符串的内容，来创建可变字符串对象

```
StringBuilder sb = new StringBuilder();
```

```
StringBuilder sb = new StringBuilder("abc");
```

5.1.4 StringBuilder类

StringBuilder常用方法

方法名	说明
public StringBuilder append(任意类型)	添加数据，并返回对象本身
public StringBuilder reverse()	反转容器中的内容
public int length()	返回长度（字符出现的个数）
public String toString()	通过toString()就可以把StringBuilder转换为String

5.1.4 StringBuilder类

对称字符串

- 需求：键盘接受一个字符串，程序判断出该字符串是否是对称字符串，并在控制台打印是或者不是。

- 对称字符串：123321，111
- 非对称字符串：123123

使用场景：字符串的拼接，字符串的反转

/5.2

System类与Runtime类

5.2.1 System类

System也是一个工具类，提供了一些与系统相关的方法

方法名	说明
public static void exit(int status)	终止当前运行的Java虚拟机，若状态码为0，表示正常终止，非0为异常终止
public static long currentTimeMillis()	返回当前系统的时间毫秒值形式
public static void arraycopy(数据源数组，起始索引，目的地数组，起始索引，拷贝个数)	数组拷贝
public static void gc()	运行垃圾回收器，用于对垃圾进行回收
public static Properties getProperties()	取得当前的系统属性
public static String getProperty(String key)	获得指定键描述系统属性

5.2.1 System类

计算机中的时间原点： 1970年1月1日 08:00:00

原因：

1969年8月，贝尔实验室的程序员肯汤普逊利用妻儿离开一个月的机会。开始着手创造一个全新的革命性的操作系统。

他使用B编程语言在老旧的PDP-7机器上开发出了Unix的一个版本。

随后，汤普逊和同事丹尼斯里奇改进了B语言，开发出了C语言，重写了UNIX。

1970年1月1日 算C语言的生日

1秒 = 1000毫秒　　1毫秒 = 1000微秒　　1微秒 = 1000纳秒

5.2.1 System类

arraycopy()

- src:表示源数组
- dest:表示目标数组
- srcPos:表示源数组中复制元素的起始位置
- destPos:表示复制到目标数组的起始位置
- length:表示复制元素的个数

5.2.2 Runtime类

Runtime表示当前虚拟机的运行环境

方法名	说明
public static Runtime getRuntime()	当前系统的运行环境对象
public void exit(int status)	停止虚拟机
Public int availableProcessors()	获得CPU的线程数
public long maxMemory()	JVM能从系统中获取总内存大小（单位byte）
Public long totalMemory()	JVM已经从系统中获取总内存大小（单位byte）
public long freeMemory()	JVM剩余内存大小（单位byte）
public Process exec(String command)	运行cmd命令

5.2.2 Runtime类

运行cmd命令：shutdown: 关机

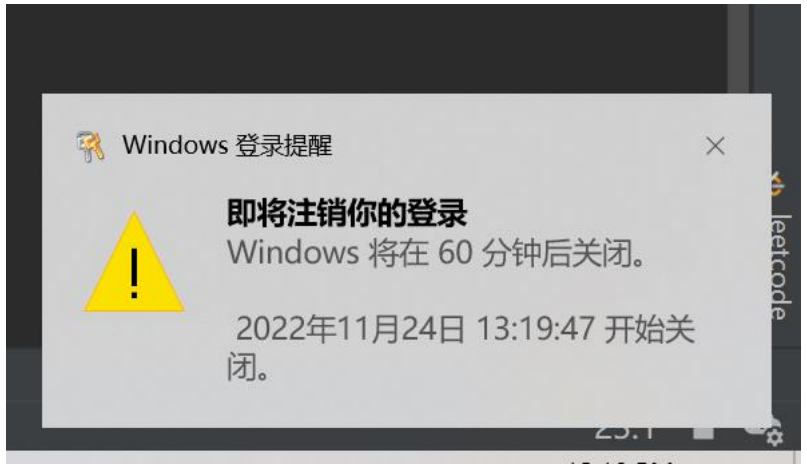
加上参数才能执行

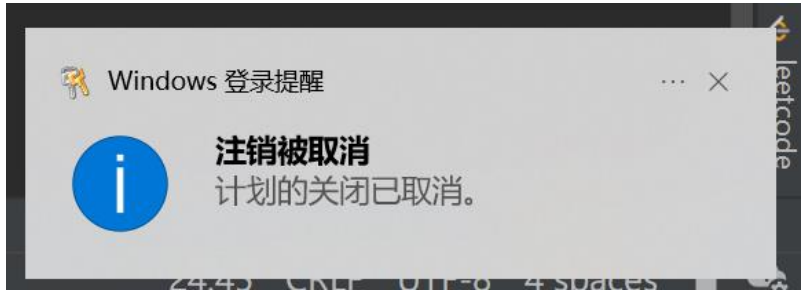
-s: 默认在一分钟之后关机

-s –t 指定时间：指定关机时间

-a:取消关机操作

-r: 关机并重启





/5.3

Math类与Random类

5.3 Math类与Random类

是一个帮助我们用于进行数学计算的工具类; 私有化构造方法，所有方法都是静态的。

方法名	说明
public static int **abs**(int a)	获取参数绝对值
public static double **sqrt**(double a)	计算方根
public static double **ceil**(double a)	向上取整
public static double **floor**(double a)	向下取整
public static int **round**(float a)	四舍五入
public static int **max**(int a, int b)	获取两个int中的较大值
public static int **min**(int a, int b)	获取两个int中的较小值
public static double **pow**(double e, double b)	返回a的b次幂，即a^b的值
public static double **random**()	返回值为double的随机值，范围[0.0, 1.0)

5.3 Math类与Random类

Random的构造方法

方法名声明	功能描述
Random()	构造方法，用于创建一个伪随机数生成器
Random (long seed)	构造方法，使用一个long型的seed(种子)创建伪随机数生成器

Random(): 传空参，没有指定种子，系统会以当前时间戳作为种子来产生随机数。

Random(long seed): 如果指定了相同的种子，则每个实例对象产生的随机数具有相同的序列。

5.3 Math类与Random类

Random类的常用方法

方法名声明	功能描述
double nextDouble()	生成double类型的随机数
float nextFloat()	生成float类型的随机数
int nextInt()	生成int类型的随机数
int nextInt(int n)	生成0~n int类型的随机数

/5.4

日期时间类

5.4 日期时间类

时间的相关知识点

世界标准时间是格林威治时间。

全球有24个时区，本初子午线东边的12个时区，都是增加时间，西边的12个时区减少时间。

中国标准时间：时间标准时间 **+8小时**

格林威治时间(Greenwich Mean Time) 简称GMT

原子钟：利用铯原子的震动的频率计算出来的时间，作为世界标准时间(UTC)

5.4 日期时间类

JDK8新增时间相关类

ZoneId: 时区

Instant: 时间戳

ZoneDateTime: 带时区的时间

DateTimeFormatter: 用于时间的格式化和解析

LocalDate: 年，月，日

LocalTime: 时，分，秒

LocalDateTime: 年，月，日，时，分，秒

Duration: 时间间隔（秒，纳秒）

Period: 时间间隔（年，月，日）

5.4 日期时间类

Instant时间戳

类的名称	功能描述
static Instant **now()**	获取当前时间的Instant对象（标准时间）
static Instant **ofXxxx**(long epochMilli)	根据（秒/毫秒/纳秒）获取Instant对象
ZonedDateTime **atZone**(ZoneId zone)	指定时区
Boolean **isXxx**(Instant otherInstant)	判断系列的方法
Instant **minusXxx**(long millisToSubtract)	减少时间系列的方法
Instant **plusXxx**(long millisToSubtract)	增加时间系列的方法

5.4 日期时间类

DateTimeFormatter用于时间的格式化和解析

方法名	说明
static DateTimeFormatter ofPattern(格式)	获取格式对象
String formate(时间对象)	按照指定方式格式化

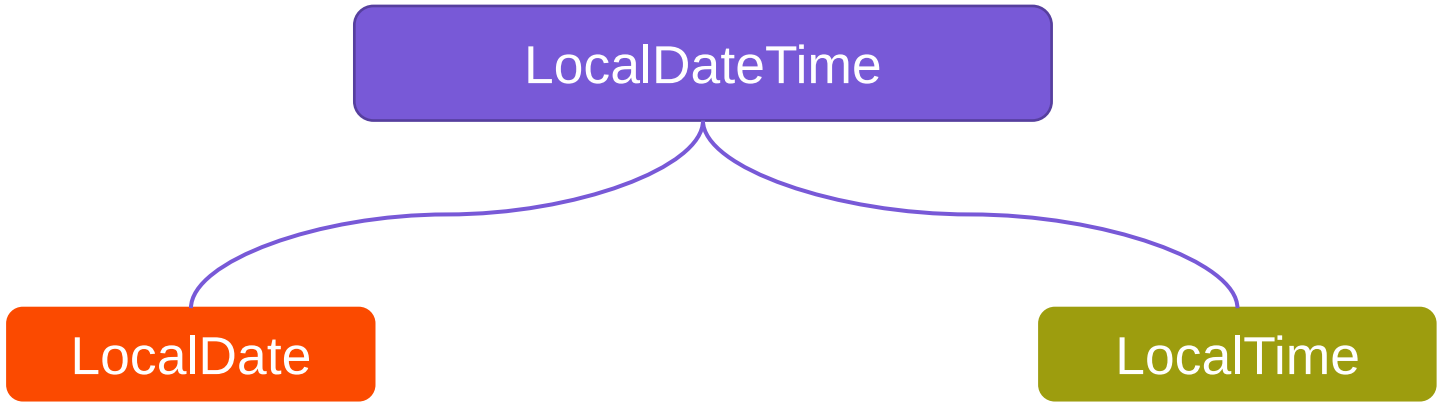
5.4 日期时间类

LocalDate, LocalTime, LocalDateTime

方法名	说明
static XXX now()	获取当前时间的对象
static XXX of(…)	获取指定时间的对象
get开头的方法	获取日历中的年，月，日，时，分，秒等信息
isBefore, isAfter	比较两个LocalDate
with开头的	修改时间系列的方法
minus开头的	减少时间系列的方法
plus开头的	增加时间系列的方法

5.4 日期时间类

LocalDate, LocalTime, LocalDateTime



方法名	说明
public LocalDate toLocalDate()	LocalDateTime转换成一个LocalDate对象
public LocalTime toLocalTime()	LocalDateTime转换成一个LocalTime对象

5.4日期时间类

Duration： 用于计算两个“时间”间隔（秒，纳秒）

Period： 用于计算两个“日期”间隔（年，月，日）

/5.5

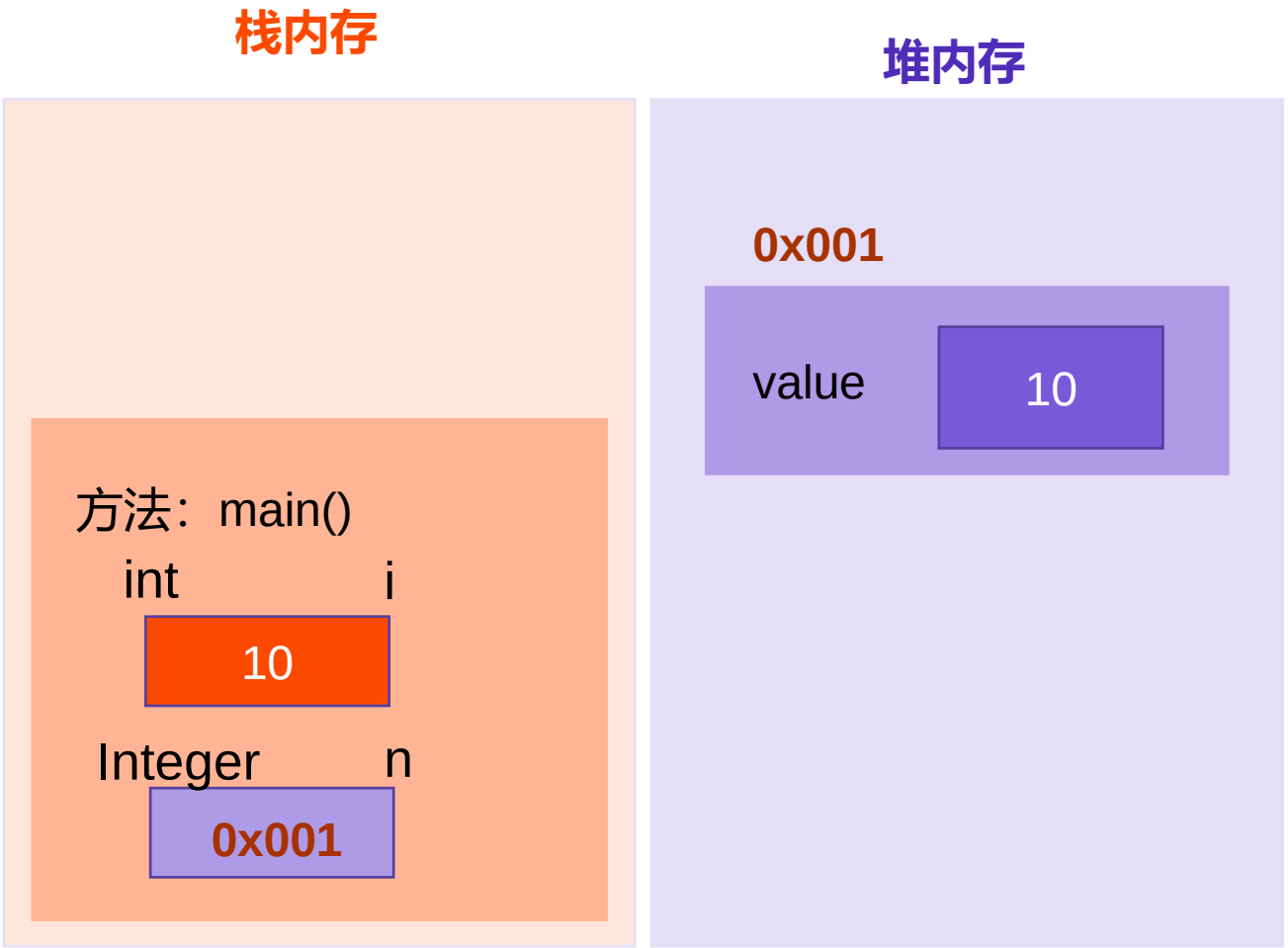
包装类

5.5 包装类

包装类：基本数据类型对应的引用类型

```java
public class A10_PackingDemo {
    public static void main(String[] args) {
        int i = 10;
        Integer n = new Integer( value: 10);
    }
}
```

包装类：用一个对象，把基本数据类型给包起来



5.5 包装类

为什么要学习包装类?

```
public void method(Object obj){
    ...
}
```

1. 由于多态的存在，我们经常传object，但如果没有包装类，我们传int类型这些方法就会接收不了。
2. 而且在集合当中，我们不能存基本数据类型，只能存对象。

5.5 包装类

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

5.5 包装类

Integer，JDK5以后

自动装箱：把基本数据类型自动变成其对应的包装类

自动拆箱：把包装类自动的变成其对象的基本数据类型

5.5 包装类

方法声明	功能描述
Integer valueOf(int i)	返回一个表示指定的int值的Integer实例
Integer ValueOf(String s)	返回保存指定的String值得Integer对象
int parseInt(String s)	讲字符串作为有符号得十进制整数进行解析
intValue()	将Integer类型的值以int类型返回

除了Character以外，包装类都有ValueOf, parseXxx(String s)的静态方法，将字符串转换为对应的基本数据类型

注意：参数不能为null, 而且字符串必须解析为相应的基本数据类型。

/5.6

正则表达式

5.6 正则表达式

正则表达式的作用：

作用一： 校验字符串是否满足规则

作用二： 在一段文本中查找满足要求的内容

5.6 正则表达式

字符类（只匹配一个字符）

[abc]	只能是a, b, 或c
[^abc]	除了a, b, c之外的任何字符
[a-zA-Z]	a到z A到Z，包括（范围）
[a-d[m-p]]	a到d，或m到p
[a-z&&[def]]	a-z和def的交集。为：d, e, f
[a-z&&[^bc]]	a-z和非bc的交集。（等同于[ad-z]）
[a-z&&[^m-p]]	a到z和除了m到p的交集。（等同于[a-lq-z]）

预定义字符（只匹配一个字符）

.	任何字符
\d	一个数字：[0-9]
\D	非数字：[^0-9]
\s	一个空白字符：[\t\n\x0B\f\r]
\S	非空白字符：[^\s]
\w	[a-zA-Z_0-9] 英文、数字、下划线
\W	[^\w] 一个非单词字符

5.6 正则表达式

数量词

X?	X，一次或0次
X*	X，零次或多次
X+	X，一次或多次
X {n}	X，正好n次
X {n, }	X，至少n次
X {n,m}	X，至少n但不超过m次

5.6 正则表达式

练习：请使用正则表达式完成如下需求

需求：

请编写正则表达式验证用户输入的手机号码是否满足要求。

请编写正则表达式验证用户输入的邮箱号是否满足要求。

请编写正则表达式验证用户输入的电话号码是否满足要求。

示例：

验证手机号码：13112345678 13712345667 15824678882 18990012490

验证座机电话号码： 020-2324242 0214224222 0817-2319155

验证邮箱号码： 3232323@qq.com Zhangsan@cwnu.edu.cn Lisi@163.com

爬虫练习

Java自从95年问世以来，经历了很多版本，目前企业在用的最多的是Java8和Java11，因为这两个是长期支持版本，下一个长期支持版本是Java17，相信在未来不久Java17也会逐渐登上历史舞台。

要求:找出里面所有的JavaXX

第五章总结

1. 字符串类
2. System类与Runtime类
3. Math类与Random类
4. 日期时间类
5. 包装类
6. 正则表达式