

《Java 高级 API - 就业技能》

章节	Ch01 - 集合框架和泛型
题目 1)	Java 集合框架是什么？说出一些集合框架的优点？ 答：每种编程语言中都有集合，最初的 Java 版本包含几种集合类：Vector、Stack、HashTable 和 Array。随着集合的广泛使用，Java1.2 提出了囊括所有集合接口、实现和算法的集合框架。在保证线程安全的情况下使用泛型和并发集合类，Java 已经经历了很久。它还包括在 Java 并发包中，阻塞接口以及它们的实现。集合框架的部分优点如下： (1) 使用核心集合类降低开发成本，而非实现我们自己的集合类。 (2) 随着使用经过严格测试的集合框架类，代码质量会得到提高。 (3) 通过使用 JDK 附带的集合类，可以降低代码维护成本。 (4) 复用性和可操作性。
题目 2)	集合框架中的泛型有什么优点？ 答：Java1.5 引入了泛型，所有的集合接口和实现都大量地使用它。泛型允许我们为集合提供一个可以容纳的对象类型，因此，如果你添加其它类型的任何元素，它会在编译时报错。这避免了在运行时出现 ClassCastException，因为你将会在编译时得到报错信息。泛型也使得代码整洁，我们不需要使用显式转换和 instanceof 操作符。它也给运行时带来好处，因为不会产生类型检查的字节码指令。
题目 3)	Java 集合框架的基础接口有哪些？ 答：(1) Collection 为集合层级的根接口。一个集合代表一组对象，这些对象即为它的元素。Java 平台不提供这个接口任何直接的实现。 (2) Set 是一个不能包含重复元素的集合。这个接口对数学集合抽象进行建模，被用来代表集合，就如一副牌。 (3) List 是一个有序集合，可以包含重复元素。你可以通过它的索引来访问任何元素。List 更像长度动态变换的数组。 (4) Map 是一个将 key 映射到 value 的对象。一个 Map 不能包含重复的 key：每个 key 最多只能映射一个 value。 (5) 一些其它的接口有 Queue、Deque、SortedSet、SortedMap 和 ListIterator。
题目 4)	为何 Map 接口不继承 Collection 接口？ 答：(1) 尽管 Map 接口和它的实现也是集合框架的一部分，但 Map 不是集合，集合也不是 Map。因此，Map 继承 Collection 毫无意义，反之亦然。 (2) 如果 Map 继承 Collection 接口，那么元素去哪儿？Map 包含 key-value 对，它提供抽取 key 或 value 列表集合的方法，但是它不适合“一组对象”规范。
题目 5)	Iterator 是什么？ 答：Iterator 接口提供遍历任何 Collection 的接口。我们可以从一个 Collection 中使用迭代器方法来获取迭代器实例。迭代器取代了 Java 集合框架中的 Enumeration。迭代器允许调用者在迭代过程中移除元素。

章节	Ch02 - 实用类介绍
题目 1)	如何将字符串转化成 int？

	答：使用包装类 Integer。Integer.valueOf("2");其他基本数据类型都是类似
题目 2)	为什么在 Java 中存储密码要使用 char[]，而不使用 String？
	答：因为 String 是不可变的，当 String 创建之后，就不会再改变了。而 char[]是可变的，程序员在用完之后可以将其改变，不会留下任何原始数据。所以，在使用密码时，一般都是会用 char 数组，例如 JavaSwing 中的 JPasswordField，获取密码框的值就是使用 char 数据。而不使用 String。
题目 3)	如何将字符串转换成时间 Date？
	答：使用 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd"); Date date = sdf.parse("2017-12-10")。
题目 4)	字符串对象 String，StringBuilder，StringBuffer 之间的区别？
	答：String 不可变的，其余两个为可变的，也就是可以自由增加或者删除字符串。StringBuilder 和 StringBuffer，前者是线程不安全的，后者是线程安全的。理论上 StringBuilder 效率更高。字符串大量拼接时，建议使用 StringBuilder，StringBuffer。
题目 5)	计算一个字符串某个字符的出现次数？
	答：(1) 使用 charAt 方法截取之后，循环判断。 (2) 使用 apache commons lang 包中的 StringUtils： int n = StringUtils.countMatches("ababababab", "a"); System.out.println(n);

章节	Ch03 - 输入和输出处理
题目 1)	java 中有几种类型的流？JDK 为每种类型的流提供了一些抽象类以供继承，请说出他们分别是哪些类？
	答：(1) 字节流，字符流。 (2) 字节流继承于 InputStream OutputStream， (3) 字符流继承于 Reader Writer。在 java.io 包中还有许多其他的流，主要是为了提高性能和使用方便。
题目 2)	什么是 java 序列化，如何实现 java 序列化？
	答：序列化就是一种用来处理对象流的机制，所谓对象流也就是将对象的内容进行流化。可以对流化后的对象进行读写操作，也可将流化后的对象传输于网络之间。序列化是为了解决在对对象流进行读写操作时所引发的问题。 序列化的实现：将需要被序列化的类实现 Serializable 接口，该接口没有需要实现的方法，implements Serializable 只是为了标注该对象是可被序列化的，然后使用一个输出流(如：FileOutputStream)来构造一个 ObjectOutputStream(对象流)对象，接着，使用 ObjectOutputStream 对象的 writeObject(Object obj)方法就可以将参数为 obj 的对象写出(即保存其状态)，要恢复的话则用输入流。
题目 3)	使用处理流的优势有哪些？如何识别所使用的流是处理流还是节点流？
	答：优势：对开发人员来说，使用处理流进行输入/输出操作更简单；使用处理流的执行效率更高。 判别：处理流的构造器的参数不是一个物理节点，而是已经存在的流。而节点流都是直接以物理 IO 及节点作为构造器参数的。
题目 4)	什么是标准的 I/O 流？
	答：在 java 语言中，用 stdin 表示键盘，用 stdout 表示监视器。他们均被封装在 System 类的类变量 in 和 out 中，对应于系统调用 System.in 和 System.out。这样的两个流加上

	System.err 统称为标准流，它们是在 System 类中声明的 3 个类变量： <pre>public static InputStream in; public static PrintStream out; public static PrintStream err;</pre>
题目 5)	什么能被序列化?什么不能被序列化?
	<p>答：能被序列化：①属性（包括基本数据类型、数组、对其它对象的引用）；②类名； 不能被序列化：①static 的属性；②方法；③加了 transient 修饰符的属性。</p> <p>要想序列化对象，必须先创建一个 outputstream,然后把它嵌进 objectoutputstream。 这时就能用 writeobject()方法把对象写入 outputstream,。读的时候需要把 inputstream 嵌到 objectinputstream 中，然后在调用 readobject()方法。</p>

章节	Ch04 - XML
题目 1)	XML 是什么?
	<p>答：XML 可扩展标记语言(Extensible Markup language)，可以根据自己的需要扩展 XML。 XML 中可以轻松定义，等自定义标签，而在 HTML 等其他标记语言中须使用预定义标签， 比如，而不能使用用户定义的标。</p>
题目 2)	DTD 与 XML Schema 有何区别?
	<p>答：DTD 与 XML Schema 有以下区别：DTD 不使用 XML 编写而 XML Schema 本身就是 xml 文件，这表示 XML 解析器等已有的 XML 工具可以用来处理 XML Schema。并且 XML Schema 是设计于 DTD 之后的，它提供了更多的类型来映射 xml 文件不同的数据类型。DTD 是文档类型描述(Document Type definition)是定义 XML 文件结构的传统方式。</p>
题目 3)	XML CDATA 是什么?
	<p>答：CDATA 是指字符数据，它有特殊的指令被 XML 解析器解析。XML 解析器解析 XML 文档中所有的文本，比如 This is name of person，标签的值也会被解析，因为标签值也可能包含 XML 标签，比如 First Name。CDATA 部分不会被 XML 解析器解析。CDATA 部分以结束。</p>
题目 4)	XSLT 是什么?
	<p>答：XSLT 也是常用的 XML 技术，用于将一个 XML 文件转换为另一种 XML，HTML 或者其他的格式。XSLT 为转换 XML 文件详细定义了自己的语法，函数和操作符。通常由 XSLT 引擎完成转换，XSLT 引擎读取 XSLT 语法编写的 XML 样式表或者 XSL 文件的指令。XSLT 大量使用递归来执行转换。一个常见 XSLT 使用就是将 XML 文件中的数据作为 HTML 页面显示。XSLT 也可以很方便地把一种 XML 文件转换为另一种 XML 文档</p>
题目 5)	DOM 和 SAX 解析器有什么区别?
	<p>答：DOM 和 SAX 解析器的主要区别在于它们解析 XML 文档的方式。使用 DOM 解析时，XML 文档以树形结构的形式加载到内存中，而 SAX 是事件驱动的解析器。</p>

章节	Ch05 - Java 反射机制
题目 1)	反射获取字节码对象方式、创建对象的两种方式?

	<p>答：①获取字节码方式三种：</p> <p>(1)类名.class，例如：System.class</p> <p>(2)对象.getClass()，例如：new Date().getClass();</p> <p>(3)Class.forName("类名")，例如：Class.forName("java.util.Date");</p> <p>②创建对象的两种方式：</p> <p>(1)直接用字节码创建对象，只能调用默认的构造方法：字节码.newInstance();</p> <p>(2)获取构造方法 Constructor，然后调用构造方法创建，可以通过参数不同调用不同的构造方式。</p>
题目 2)	怎么理解反射，反射的应用？
	<p>答：在运行状态下，通过类的字节码文件去解剖一个类，使用该类的构造方法、成员变量、成员函数。反射就是把 Java 类中的各种成分映射成相应的 Java 类。一般情况下我们要解决某个问题，先找到相关的类，创建该类的对象，然后通过该对象调用对应的方法来解决。反射是一个正好相反的过程，开始可能并没有类可以解决这个问题，而我们却先用一个当时可能并不存在的方法解决了这个问题，后来才有的这个类。这其实就是框架的原理，现有的框架再有的解决问题的类。框架描述了整体，制订了功能方法的规范，具体的实现之后按照这个规范编写。这些都需要靠反射来完成。使用框架有良好的扩展性，某部分功能的优化不需要涉及程序整体，只需要修改特定的部分就好了，然后通过配置文件，获取对应的类名，就可以了。</p>
题目 3)	什么是 reflection？
	<p>答：Java 中的反射是一种强大的工具，它能够创建灵活的代码，这些代码可以在运行时装配，无需在组件之间进行链接。反射允许在编写与执行时，使程序代码能够接入装载到 JVM 中的类的内部信息，而不是源代码中选定的类协作的代码。这使反射成为构建灵活应用的主要工具。需要注意的是，如果使用不当，发射的成本会很高。</p>
题目 4)	反射的原理是什么？
	<p>答：反射是为了能够动态的加载一个类，动态的调用一个方法，动态的访问一个属性等动态要求而设计的。它的出发点就在于 JVM 会为每个类创建一个 java.lang.Class 类的实例，通过该对象可以获取这个类的信息，然后通过使用 java.lang.reflect 包下得 API 以达到各种动态需求。</p>
题目 5)	Class 类的含义和作用是什么？
	<p>答：每一个 Class 类的对象就代表了一种被加载进入 JVM 的类，它代表了该类的一种信息映射。开发者可以通过 3 种途径获取到 Class 对象。</p> <p>①.Class 的 forName()方法的返回值就是 Class 类型，也就是动态导入类的 Class 对象的引用。 public static Class<?> forName(String className) throws ClassNotFoundException</p> <p>②.每个类都会有一个名称为 Class 的静态属性，通过它也是可以获取到 Class 对象的，示例代码如下：</p> <pre>Class<Student> clazz = Student.class; // 访问 Student 类的 class 属性</pre> <p>③.Object 类中有一个名为 getClass 的成员方法，它返回的是对象的运行时类的 Class 对象。因为 Object 类是所有类的父类，所以，所有的对象都可以使用该方法得到它运行时类的 Class 对象，示例代码如下：</p> <pre>Student stu = new Student(); Class<Student> clazz = stu.getClass(); // 调用 Student 对象 getName 方法</pre>

章节	Ch06 - 项目实战-嗖嗖移动业务大厅（一）
----	-------------------------

题目 1)	什么是 Java 集合 API?
	答: Java 集合 API 是用来表示和操作集合的统一框架, 包括接口、实现类以及帮助程序员完成一些编程的算法。
题目 2)	线程和进程有什么区别?
	答: 线程是进程的子集, 一个进程可以有很多线程, 每条线程并行执行不同的任务。不同的进程使用不同的内存空间, 而所有的线程共享一片相同的内存空间。
题目 3)	如何在 Java 中实现线程?
	答: 在语言层面有两种方式。java.lang.Thread 类的实例就是一个线程但是它需要调用 java.lang.Runnable 接口来执行, 由于线程类本身就是调用的 Runnable 接口所以你可以继承 java.lang.Thread 类或者直接调用 Runnable 接口来重写 run()方法实现线程。
题目 4)	Iterator 和 ListIterator 的区别是什么?
	答: (1) Iterator 可以遍历 Set 和 List 集合, 而 ListIterator 只能遍历 List。 (2) Iterator 只可以向前遍历, 而 ListIterator 可以双向遍历。 (3) ListIterator 从 Iterator 接口继承, 然后添加了一些额外的功能, 比如添加一个元素、替换一个元素、获取前面或后面元素的索引位置。
题目 5)	HashMap 和 Hashtable
	答: 两者都是用 key-value 方式获取数据。Hashtable 是原始集合类之一 (也称作遗留类)。HashMap 作为新集合框架的一部分在 Java2 的 1.2 版本中加入。它们之间有以下区别: (1) HashMap 允许 key 和 value 为 null, 而 Hashtable 不允许。 (2) Hashtable 是同步的, 而 HashMap 不是。所以 HashMap 适合单线程环境, Hashtable 适合多线程环境。 (3) HashMap 提供对 key 的 Set 进行遍历, 因此它是 fail-fast 的, 但 Hashtable 提供对 key 的 Enumeration 进行遍历, 它不支持 fail-fast。 (4) Hashtable 被认为是个遗留的类, 如果寻求在迭代的时候修改 Map, 应使用 ConcurrentHashMap。

章节	Ch07 - 项目实战-嗖嗖移动业务大厅 (二)
题目 1)	如何决定选用 HashMap 还是 TreeMap?
	答: 对于在 Map 中插入、删除和定位元素这类操作, HashMap 是最好的选择。然而, 假如你需要对一个有序的 key 集合进行遍历, TreeMap 是更好的选择。基于你的 collection 的大小, 也许向 HashMap 中添加元素会更快, 将 map 换为 TreeMap 进行有序 key 的遍历。
题目 2)	在 Hashtable 中上下文同步是什么意思
	答: 同步意味着在一个时间点只能有一个线程可以修改哈希表, 任何线程在执行 hashtable 的更新操作前需要获取对象锁, 其他线程等待锁的释放。
题目 3)	哪些集合类提供对元素的随机访问
	答: ArrayList、HashMap、TreeMap 和 Hashtable 类提供对元素的随机访问。
题目 4)	有三个线程 T1, T2, T3, 怎么确保它们按顺序执行?
	答: 在多线程中有多种方法让线程按特定顺序执行, 你可以用线程类的 join()方法在一个线程中启动另一个线程, 另外一个线程完成该线程继续执行。为了确保三个线程的顺序你应该先启动最后一个(T3 调用 T2, T2 调用 T1), 这样 T1 就会先完成而 T3 最后完成。
题目 5)	5.Java 多线程中调用 wait() 和 sleep()方法有什么不同?
	答: Java 程序中 wait 和 sleep 都会造成某种形式的暂停, 它们可以满足不同的需要。wait() 方法用于线程间通信, 如果等待条件为真且其它线程被唤醒时它会释放锁, 而 sleep()方法仅

	仅释放 CPU 资源或者让当前线程停止执行一段时间，但不会释放锁。
--	-----------------------------------

面试题海