一、1-10题

1. 异常的分类

(1) java提供了两种异常错误的异常类，分别为ERROR和EXCEPTION，错误和异常，他们的父类是Throwable.

(2) error表示程序在运行期间出现了严重的错误，并且该错误是不可恢复的。

(3) exception表示可恢复的异常，是编译器可以捕捉到的。分两类：检查异常(checked exception)和运行时异常（runtime exception）

(4) 可不检测异常是针对检查来说的也就是运行时异常。\

2. AOP和OOP的关系

(1) 面向方面编程( AOP) ： 偏重业务处理过程的某个步骤或阶段，强调降低模块之间的耦合度，使代码拥有更好的移植性。

(2) 面向对象编程 (OOP) 则是对业务分析中抽取的实体进行方法和属性的封装。

(3) 核心业务还是要OOP来发挥作用，与AOP的侧重点不一样，前者有种纵向抽象的感觉，后者则是横向抽象的感觉， AOP只是OOP的补充，无替代关系。

3. 数组是一种对象，在堆里分配内存。

4. 数据库系统实现事务隔离级别

在数据库操作中，为了有效保证并发读取数据的正确性，提出的事务隔离级别；为了解决更新丢失，脏读，不可重读（包括虚读和幻读）等问题在标准SQL规范中，定义了4个事务隔离级别，分别为未授权读取，也称为读未提交（read uncommitted）；授权读取，也称为读提交（read committed）；可重复读取（repeatable read）；序列化（serializable）。

5. Java标识符规则

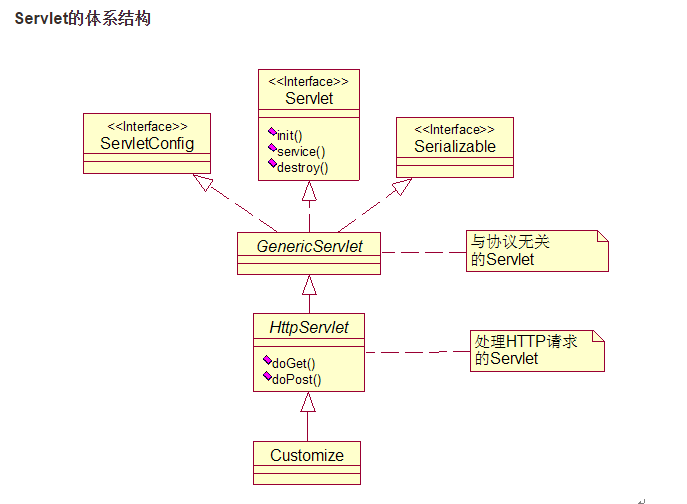
(1) 标识符由26个英文字符大小写（a~z，A~Z）、数字(0~9)、下划线(\_)和美元符号($)组成；

(2) 不能以数字开头，不能是关键字；

(3) 严格区分大小写；

(4) 标识符的可以为任意长度；

6. Servlet相关类的作用



7. 并发ThreadLocal类

(1) ThreadLocal重要作用在于多线程间的数据独立。

(2) ThreadLocal是采用哈希表的方式来为每个线程都提供一个变量的副本。

(3) ThreadLocal保证各个线程间数据安全，每个线程的数据不会被另外线程访问和破坏。

二、11-20题

1. 接口可以多继承接口，普通类只能单继承。

三、21-30题

1. javac –d 目录设置类文件的目标目录。如果某个类是一个包的组成部分，则 javac 将把该类文件放入反映包名的子目录中，必要时创建目录。

2. 序列化和反序列化

(1) 一个对象只要实现了Serilizable接口，这个对象就可以被序列化，java的这种序列化模式为开发者提供了很多便利，我们可以不必关系具体序列化的过程，只要这个类实现了Serilizable接口，这个类的所有属性和方法都会自动序列化。

(2) 这个类的有些属性需要序列化，而其他属性不需要被序列化；

java 的transient关键字为我们提供了便利，你只需要实现Serilizable接口，将不需要序列化的属性前添加关键字transient，序列化对象的时候，这个属性就不会序列化到指定的目的地中。

3. Forward和Redirect

(1) 从地址栏显示来说

forward是服务器请求资源,服务器直接访问目标地址的URL,把那个URL的响应内容读取过来,然后把这些内容再发给浏览器.浏览器根本不知道服务器发送的内容从哪里来的,所以它的地址栏还是原来的地址.

redirect是服务端根据逻辑,发送一个状态码,告诉浏览器重新去请求那个地址.所以地址栏显示的是新的URL.

(2) 从数据共享来说

forward:转发页面和转发到的页面可以共享request里面的数据.

redirect:不能共享数据.

(3) 从运用地方来说

forward:一般用于用户登陆的时候,根据角色转发到相应的模块.

redirect:一般用于用户注销登陆时返回主页面和跳转到其它的网站等.

(4) 从效率来说

forward:高.

redirect:低.

四、31-40题

1. 集合ArrayList

List<Integer> NumberList =newArrayList<Integer>();

NumberList.add(2);

NumberList.add(4);

NumberList.add(1);

NumberList.add(3);

NumberList.add(5);

for(int i =0;i<NumberList.size();++i)

{

int v = NumberList.get(i);

if(v%2==0)

{

NumberList.remove(v);

}

}

System.out.println(NumberList);

输出结果为：会出现越界情况

注意要点：

(1) ArrayList删除元素后，剩余元素会依次向前移动，因此下标一直在变，size()也会减小；

(2) remove()方法调用的是remove(int index)，而不是remove(Object o)，因此删除的是index索引处的元素；

该题具体流程：1.i=0,v=2,remove(2)删除掉了元素1，因此NumberList剩余元素为【2,4,3,5】；

2.i=1,v=4,remove(4),此时线性表中只有四个元素，不可能删除索引为4的元素，因此会报数组下标越界异常

2. 匿名内部类只能使用一次，局部内部类可以在定义范围内多次使用。

3. 构造方法的优先级一般比代码块低。--已整理

静态成员变量或静态代码块>mian方法>非静态成员变量或非静态代码块>构造方法

4. 数组有length属性，字符串只有length()方法。

5 . 在java 中，声明一个数组时，不能直接限定数组长度，只有在创建实例化对象时，才能对给定数组长度.。

五、41-50

1. String的两个方法 – 已整理

(1) String(byte[] bytes, String charsetName)

通过使用指定的 charset 解码指定的 byte 数组，构造一个新的String。

(2) getBytes(String charsetName)

使用指定的字符集将此String编码为 byte 序列，并将结果存储到一个新的 byte 数组中。

2. 形式参数

对于形式参数只能用final修饰符，其它任何修饰符都会引起编译器错误 。但是用这个修饰符也有一定的限制，就是在方法中不能对参数做任何修改。

3. 包装类和基类

包装类的“==”运算在不遇到算术运算的情况下不会自动拆箱

包装类的equals()方法不处理数据转型

如：Integer i = 42;

Long l = 42l;

Double d = 42.0;

错误：(i == l) i.equals(d) 都是不会自动拆箱的。

六、51-60题

1. 接口中默认修饰变量属性用：public static final；而默认方法的修饰符是：public abstract

为什么是public：因为接口必然是要被实现的，如果不是public，这个属性就没有意义了；

为什么是static：因为如果不是static，那么由于每个类可以继承多个接口，那就会出现重名的情况；

为什么是final：这是为了体现java的开闭原则，因为接口是一种模板，既然是模板，那就对修改关闭，对扩展开放。

2. 结果集(ResultSet)是数据中查询结果返回的一种对象，可以说结果集是一个存储查询结果的对象，但是结果集并不仅仅具有存储的功能，他同时还具有操纵数据的功能，可能完成对数据的更新等。

结果集读取数据的方法主要是getXXX() ，他的参数可以使整型表示第几列（是从1开始的），还可以是列名。返回的是对应的XXX类型的值。如果对应那列时空值，XXX是对象的话返回XXX型的空值，如果XXX是数字类型，如Float等则返回0，boolean返回false。

3. java的50个关键字是： abstract assert boolean break byte case catch char class const continue default do double else extends enum final finally float for goto if implements import instanceof int interface long native new package private protected public return short static strictfp super switch synchronized this throw throws transient try void volatile while 另外三个:true false null是保留字