一、基础常识

linux, API

POSIX:Portable Operating System 可移植系统

可以跨平台编译，但是不能跨平台运行，除非提供了ABI

.dll windows

.so linux

ABI： Application Binary Interface 二进制程序接口，不同系统的不同

JAVA 包含四个独立又彼此相关的技术：

Java 程序设计语言

Java API

Java Class 文件格式，定义了字节码格式

JVM： java virtual machine

\*.java 通过java complier 编译成\*.class文件，class loader 类加载器，在JVM上运行，需要装载对应的API库

bytecode: 字节码 必须由解释器运行

\*.java ——> \*.class (bytecode)

将字节码转换成跟平台相关的代码：解释的过程

JVM的实现方式：

1、一次性解释器，解释字节码并执行；

2、即时编译器（just-in-time complier） 缓存解释后的文件，依赖于更多内存缓存解释后的结果

3、自适应编译器。将执行频率比较高的代码缓存

JVM：

官方提供的，包含：

JRE：Java 运行时环境 (运行)

JDK：Java Development Kit Java开发组件 (开发（编译）+运行)

OpenJDK

开发+运行的开源实现

JSP： Java Server Page

.jsp ——> （.java）——> Servlet ——> .class

使用jasper

CGI 通用网关接口，同web进行交互

serverlet 容器 ：提供了JDK，监控文件改变，编译

多线程的

运行时数据区域，占用的内存空间，包括以下：

线程私有内存区：

\*\*\*\*\*JAVA虚拟机栈（保存本地变量，动态链接等，多线程的，每个线程独有内存）

程序计数器（保存java运行在何处，多线程的）

线程共有内存区：

\*\*\*\*\*方法区，一类对象共有的方法（常量池）

\*\*\*\*\*堆 （实例化后的对象，就是存放对象的）（最大的内存空间），java自动内存回收，GC（Garbage Collector)

监控新生对象，老龄化对象，对象的引用计数器为0时，就批量回收。垃圾回收器可能会产生碎片，有不同的算法

本地方法栈（调用外部的方法，如系统调用）

垃圾回收算法：

1、标记-清除 会产生碎片

2、复制 二分之一可用 一个可用一个不用，不会产生碎片，但是浪费空间

3、标记-整理 1和2的结合

万一溢出了，需要监控，最有可能的就是堆区域。

垃圾回收器：

serial 单个回收

parnew 多回收，支持多线程，必须多CPU

parallel scavenge 针对新生对象的，降低垃圾回收所占据的CPU时间

serial old

parallel old

CMS: Concurrent Mark Sweep 并行标记清除

特点：并发收集、低停顿

缺点：无法收集浮动垃圾，线程在运行的无法回收

java 配置参数方法：

-XX:+<option> :开启此参数指定的功能

-XX:-<option> :关闭功能

-XX:<option>=<value> ：给option指定的选项赋值

java -XX:+PrintFlagsFinal 显示支持的所有参数及默认值

-D<name>=<value>: set a system property

类的生命周期：

1、加载 用类加载器加载

2、验证 不符合规范的不运行，格式验证、字节码验证等

3、准备

4、解析 resolution

5、初始化 init

6、使用

7、卸载

JDK 监控和故障处理工具：

jps，JVM Process Status Tool：显示指定系统内所有的HotSpot虚拟机进程的列表

jstat, JVM statistics monitoring tool：收集并显示HotSpot虚拟机各方面的运行数据

jinfo：显示正在运行的某HotSpot虚拟机配置信息

jmap： 生成某HosSpot虚拟机的内存转储快照，调试的时候才会用到

可视化工具：

jconsole： Java的监控与管理控制台

jvisualvm： 可视化的VM

Servlet: CGI

硬编码，html

JSP：实现java以标签方式，嵌入到html里面

Jasper 将 jsp 转换成 java

即 JSP ——> Servlet

web container: 包含了servlet

JAVA EE 应用服务器：

商业：websphere

weblogic 被oracle收购了

oc4j

开源：

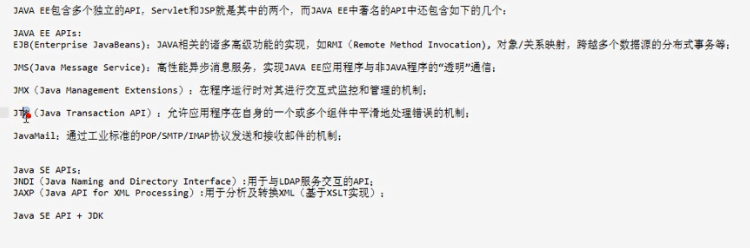
JBoss：核心是tomcat，对tomcat二次开发

JOnAS

Geronimo

Glassfish

代码还是catalina，但是程序名叫做tomcat



Tomcat 实例

Service 引擎servlet container

host context

连接器，连接server和引擎

Tomcat 核心配置文件

Server.xml

Tomcat各组件间的关系：

<Server>

</Server>

bin

catalina.sh

conf

server.xml

tomcat-users.xml 用于用户账号密码认证

web.xml 默认应用程序部署描述符

部署：将一个web应用程序所依赖到的类装载进JVM

context.xml

catalina.properties 定义属性

loggging.properties 定义日志相关属性

catalina.policy 定义安全策略

lib

类库

Logs

日志

export CATALINA\_HOME=/usr/local/tomcat

export PATH=$PATH:$CATALINA\_HOME/bin

WEB-INF META-INF 私有资源，通过web是访问不到的

工作车间在work目录下

语法测试必须在tomcat停止状态测试