作业一（编程题）：

开发 Minicat V4.0，在已有 Minicat 基础上进一步扩展，模拟出 webapps 部署效果 磁盘上放置一个 webapps目录，webapps 中可以有多个项目，比如 demo1,demo2,demo3... 具体的项目比如 demo1 中有 serlvet（也即为：servlet 是属于具体某一个项目的 servlet），这样的话在 Minicat 初始化配置加载，以及根据请求 url 查找对应 serlvet 时都需要进一步处理

！！！重要

备注：读取项目磁盘统一路径： appBase="/Users/webapps"，*并且提交自己的 webapps 以及访问路径*

【答】：

在原有的 Minicat（v3.0）上修订，修订后具体代码见文件夹【Minicat】所示。

（1）在 resources 下新增 server.xml。并写入一下内容

```
<?xml version="1.0" encoding="UTF-8" ?>
<server>
   <services>
   <connector port="8080"></connector>
        <engine>
            <host name="localhost" appBase="E:\IDEA-code\Minicat\webapps"></host>
        </engine>
   </services>
</server>
```

（2）新增 Mapper.java、Host.java、Context.java、Wrapper.java 组成 Tomcat 简版体系结构。

（3）在 Bootstrap.java 的 start 方法中进行如下修订：

[1] 新增 loadServer()加载解析 server.xml 相关配置。

[2] 并在 loadServer()中的 loadWebConfig()加载解析 appBase 路径下的 web.xml 相关配置。

[3] 并将解析处理的数据封装到 Wrapper、Context、Host 对象中，总体封装到 Mapper 中。

（4）在 RequestProcessor.java 的 run 方法中请求过滤代码。request 请求 url 到来时，与我们封装的 Mapper 数据进行一层一层的比较匹配。直到找到 Wrapper 中的 Servlet，然后执行 Servlet 中 service 方法（执行其中的 doGet 或 doPost 方法）。

（5）将生成的 webapps 代码拷贝到文件夹 Minicat 下面。运行 Bootstrap.java 中的 main 方法运行，进行测试。

作业二（简答题）： 请详细描述 Tomcat 体系结构（图文并茂）

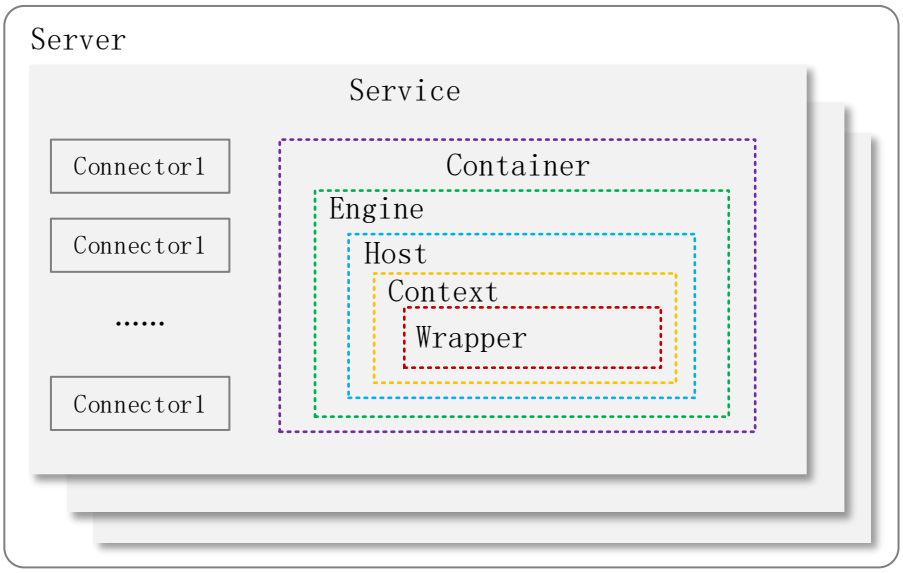
作业具体要求参考以下链接文档：

https://gitee.com/lagouedu/test/raw/master/%E7%AC%AC%E4%BA%8C%E9%98%B6%E6%AE%B5/tomcat%E4%BD%9C%E4%B8%9A%E9%A2%98/Tomcat%E4%BD%9C%E4%B8%9A%E5%A4%A7%E9%A2%98.zip

【答】：

（1）Tomcat 体系结构

Tomcat 即是 Server 服务器，一个 Server 服务器可以包含多个 Service 服务。Service 主要由 Connector 和 Container 两部分组成，Connector 用于接受请求并处理请求，Container 用于封装和管理 Servlet。具体如下图所示。



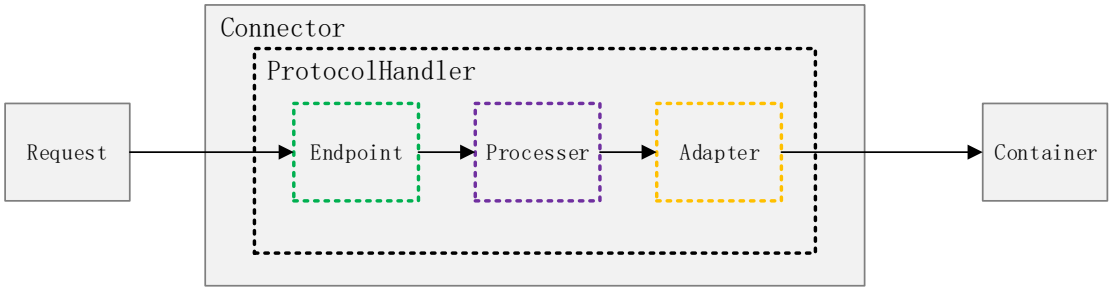
（2）Connector 连接器

Connector 通过 ProtocolHandler 来处理请求，其中包含了三大组件。

Endpoint：用来处理底层的 Socket 网络连接。

Processor：用于将 Endpoint 接收到的 Socket 封装成 Request。

Adapter：用于将 Request 交给 Container 进行具体的处理。



（3）container 容器

Container 容器主要是指 Servlet 容器，负责加载和管理 Servlet。 Tomcat 包含了 4 种组件，分别为 Engine、Host、Context、Wrapper。

Engine：表示虚拟主机的引擎，一个 Tomcat Server 只有一个引擎。

Host：表示虚拟主机，一个容器可以有多个虚拟主机。

Context：表示一个应用容器，一个虚拟主机可以拥有多个应用。

Wrapper：即对具体 Servlet 的封装。