

# 计算机网络课程期末项目

---

## 1. 项目要求

### 1.1. 功能要求（55分）

#### 1.1.1. 基本功能（30分）

#### 1.1.2. 辅助功能（15分）

#### 1.1.3. Bonus（10分，自选一个完成即可）

### 1.2. 性能要求（30分）

### 1.3. 现场演示（15分）

## 2. 项目实施及评测方法

### 2.1. Demo 项目结构

### 2.2. 需要完成的内容

### 2.3. JDK 版本问题

## 3. 提交内容及重要时间节点

## 1. 项目要求

### 1.1. 功能要求（55分）

基于 Socket 编程实现一个简易的云盘 **EasyCloudDisk**，支持文件的上传、下载等功能，采用单线程/多线程方式实现，并通过MD5校验和机制保障文件的传输。

#### 1.1.1. 基本功能（30分）

- 支持用户从本地上传文件到云盘上（10分）
- 支持用户下载云盘上的文件到本地（10分）
- 支持用户 **多线程** 上传文件（10分）

#### 1.1.2. 辅助功能（15分）

- 支持用户获取云盘上文件目录和大小（5分）
- 支持用户批量上传/下载文件（即一次性上传/下载多个文件）（5分）
- 采用 MD5 校验和保障文件传输（5分）

### 1.1.3. Bonus (10分, 自选一个完成即可)

- 支持用户从本地上传 **文件夹** 到云盘上, 要求保留目录结构
- 支持用户 **多线程** 下载文件
- 支持断点重传, 要求客户端重新上传 **缺失的文件**
- 服务器端采用 **NIO** 方式实现

## 1.2. 性能要求 (30分)

- 单线程上传 1M 文件的耗时 < 1 秒 (**10分**)
- 多线程上传 1M 文件的耗时 < 1 秒 (**10分**)
- 循环 10 次单线程下载 100KB 文件的耗时 < 1 秒 (**10分**)

注: 根据实验报告统计后续可能调整性能要求时间

## 1.3. 现场演示 (15分)

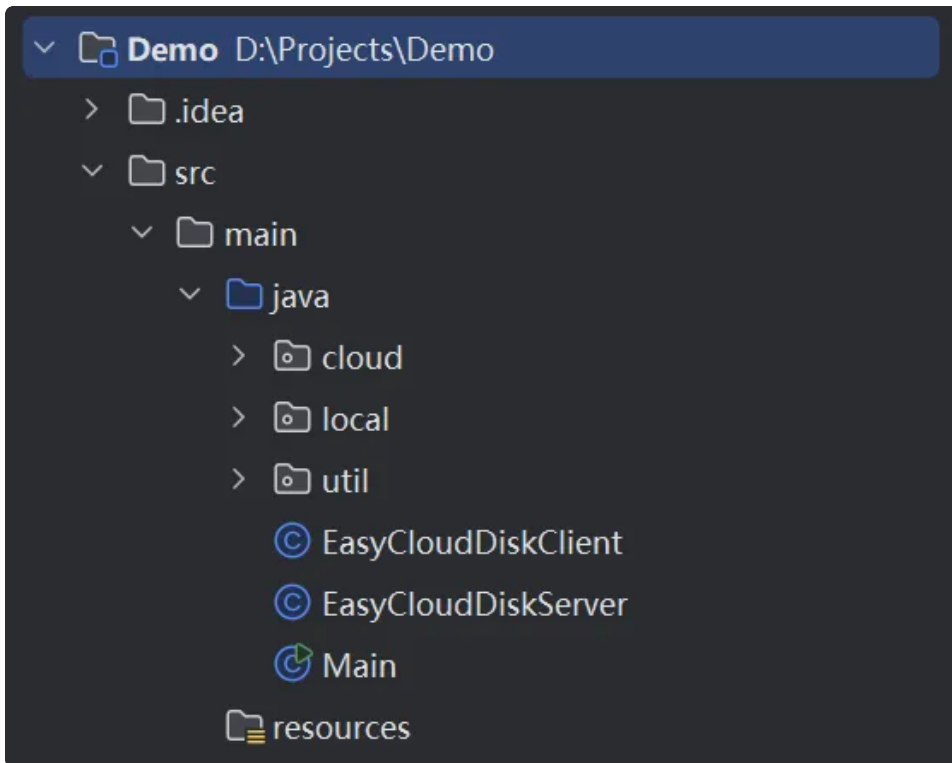
现场演示将考察代码的完成情况以及你对代码的理解, 具体包含以下三部分内容:

1. 执行提供测试类的 `main` 函数
2. 抽查展示云盘功能并简述实现原理
3. 展示 Bonus 并简述实现原理

# 2. 项目实施及评测方法

## 2.1. Demo 项目结构

期末项目将提供一份 Demo 项目代码, 你需要在现有代码的基础之上对项目所需的各项功能进行实现, Demo 结构如下:



- `cloud` : 模拟云盘存储路径
- `local` : 模拟本地文件存储路径
- `util` : 辅助评测函数
- `main` : 用于功能评测和性能评测
- `EasyCloudDiskClient` 和 `EasyCloudDiskClient` : 需要实现的类

注：你可以自行添加其他类、函数以辅助实现各项功能，但请勿修改现有代码结构（类名、方法名等）；可能会对你提交的代码进行检查，**请勿抄袭!!!**

## 2.2. 需要完成的内容

Demo 中包含若干个 TODO，你需要对其进行实现，确保实现后的代码能够成功通过 `main` 函数测试

```
1  ▼ public class EasyCloudDiskClient {
2  ▼      /**
3      * 启动客户端
4      */
5  ▼      public void start() {
6          // TODO
7      }
8
9  ▼      /**
10     * 单线程上传文件
11     *
12     * @param localFilePath 本地文件路径
13     * @param remoteFilePath 云盘文件路径
14     */
15  ▼      public void uploadFileSingleThread(String localFilePath, String remote
FilePath) {
16          // TODO
17      }
18
19
20  ▼      /**
21     * 多线程上传文件
22     *
23     * @param localFilePath 本地文件路径
24     * @param remoteFilePath 云盘文件路径
25     */
26  ▼      public void uploadFileMultiThread(String localFilePath, String remoteF
ilePath) {
27          // TODO
28      }
29
30  ▼      /**
31     * 下载文件
32     *
33     * @param remoteFilePath 云盘文件路径
34     * @param localFilePath 本地文件路径
35     */
36  ▼      public void downloadFile(String remoteFilePath, String localFilePath)
{
37          // TODO
38      }
39  }
40
```

评测通过后输出结果如下：

评测通过输出

Shell

1

[功能测试]

单线程上传文件功能通过

2

[功能测试]

多线程上传文件功能通过

3

[功能测试]

单线程下载文件功能通过

4

[性能测试]

单线程上传 10MB 文件的耗时 耗时：0.377 s

5

[性能测试]

多线程上传 10MB 文件的耗时 耗时：0.145 s

6

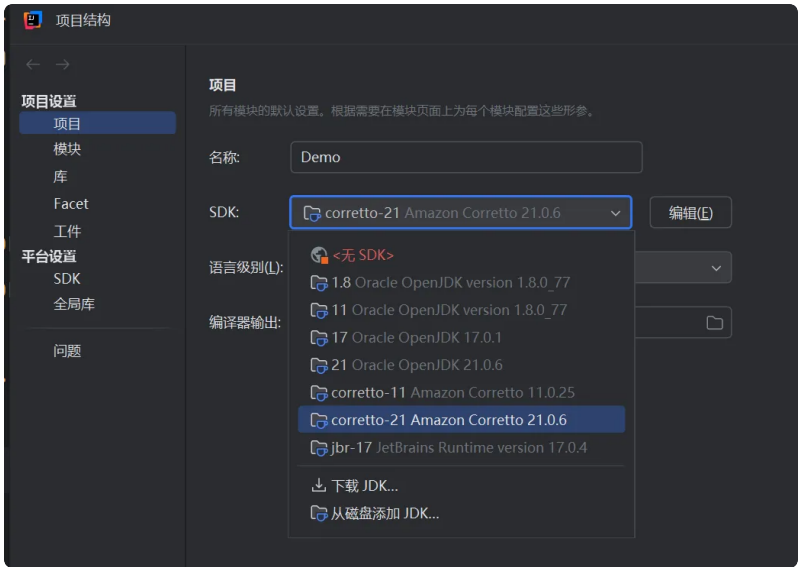
[性能测试]

循环 10 次单线程下载 1MB 文件的耗时 耗时：0.380 s

## 2.3. JDK 版本问题

项目提供的 JDK 版本为 21，如果本地的 JDK 版本与项目版本不一致，可以参照如下方法进行修改：

- 1. 修改项目结构中的 SDK 版本为你本地安装的版本



- 2. 修改 pom.xml 中的变量（如 JDK1.8 请修改为 8），然后重新构建 maven 项目

```
<properties>
  <maven.compiler.source>21</maven.compiler.source>
  <maven.compiler.target>21</maven.compiler.target>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>
```

## 3. 提交内容及重要时间节点

- 现场演示
  - 提交方式：线下，具体地点**待定**

- 演示时间：六月，具体时间待定
- 实验报告及项目代码
  - 提交方式：线上钉钉群提交
  - 提交时间：六月，具体时间待定
  - 要求：实验报告应讲述功能的实现原理和设计思路，避免简单复制粘贴代码和运行截图；实验报告中的上机实践名称为**期末项目**，上机时间编号、组号、上机实践时间不用填写；提交时请将实验报告和项目代码打包在同一个文件夹中放在同一级目录下，格式如下：

```
▼ Shell |
1  姓名_学号_期末项目/
2  |— Demo/
3  |— 实验报告.docx
```