# 文件同步器设计说明书

1. **设计背景**

在计算机软件中，我们常常可以看见办公软件有文件上传云文档备份，和从云文档中的下载文档的功能。云文档就是一个文件服务器，帮助用户管理文件，控制文件版本。为了模拟云文档的功能，可以自己用编程语言实现一个文件同步器，来做到文件上传，下载和删除的功能。

1. **环境搭建**

1)操作系统: windows 10

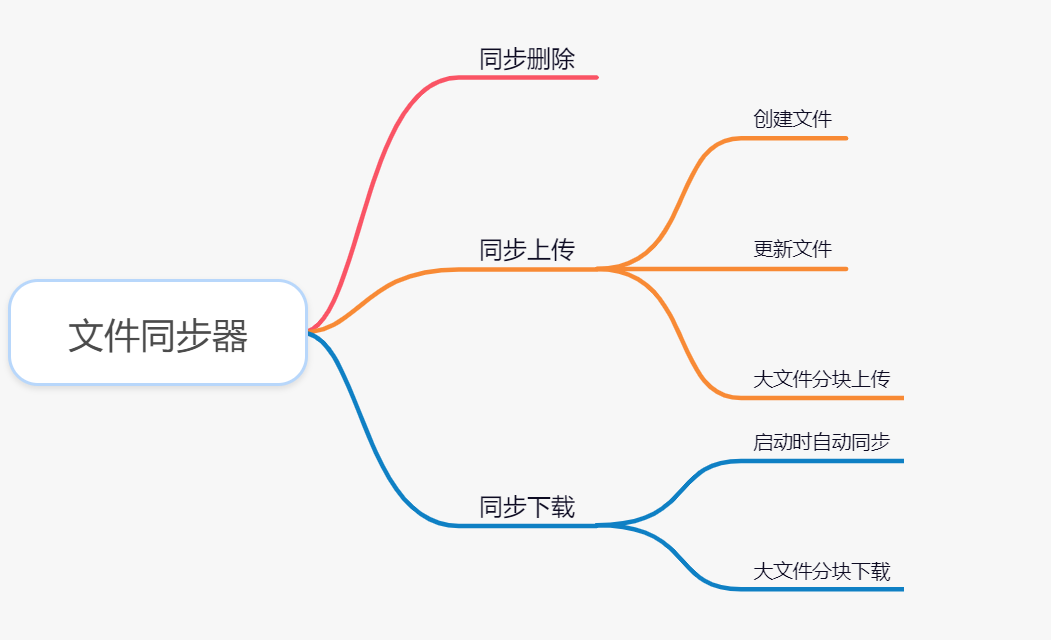
2)编程语言: java(jdk1.8)

3)集成开发环境: IntelliJ IDEA 2020

4)服务器客户端接口: S3 Browser

5)jar依赖包: aws-java-sdk 1.11.1030 commons-codec 1.11

**三．功能设计**



1. 同步删除:在本地文件夹中删除文件时，文件同步器监听到删除文件的事件，然后开启异步任务去删除远程仓库的文件。
2. 同步上传:同步上传可以通过创建文件事件和更新文件事件的发生而触发。在上传之前需要检测文件大小，如果大于20MB则需要进行文件分块上传。
3. 同步下载:程序启动时，程序需要将远程仓库的文件拉取下来，同时解决文件冲突问题，如文件相同则不需要下载，文件重名则需要保存在一个新的副本中。下载文件也需要检测文件是否需要分块下载。

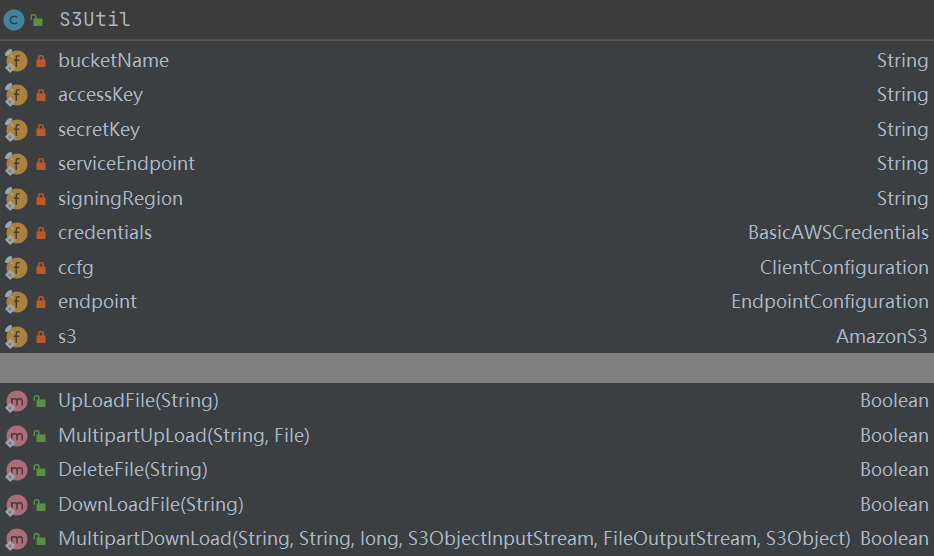
**四．程序实现**

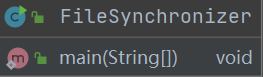
1）程序使用到了两个类来实现功能。

I）FileSynchronizer类:文件同步器类。用于只有一个main方法，是程序的启动入口，用于监听文件夹的变化，便于进行文件同步。

II）S3Util类:S3客户端操控类。内部实现了文件的上传，删除，下载的静态方法。FileSynchronizer类监听到事件后，会相应地调用S3Util的静态方法来进行文件同步。

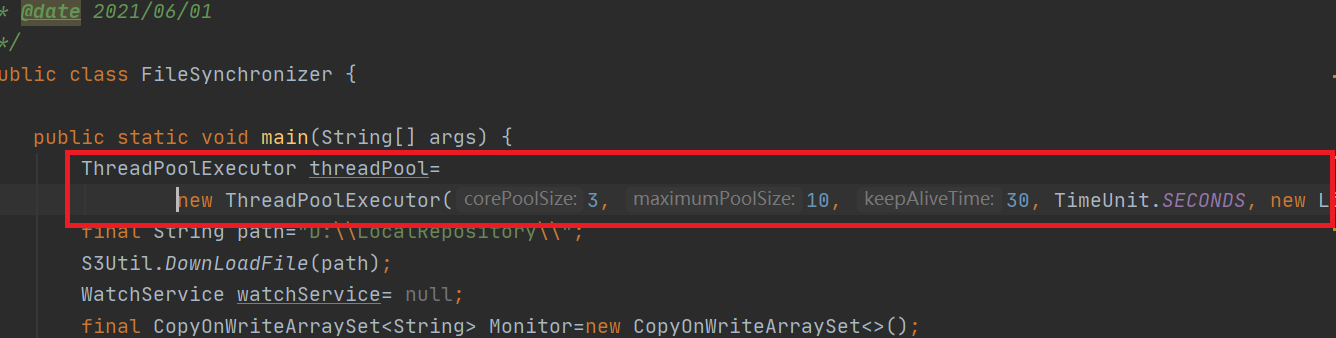
1. UML类图



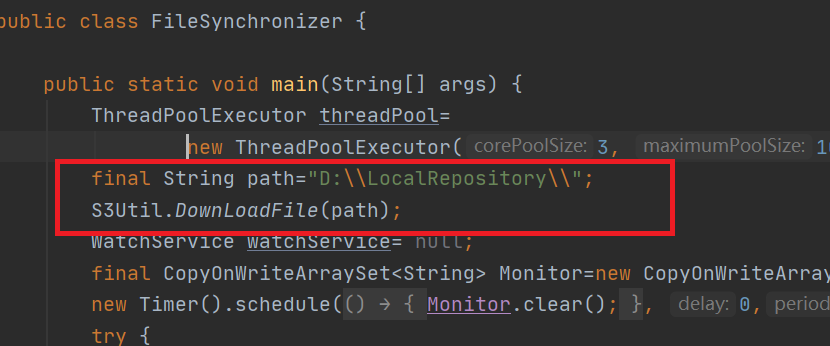


1. FileSynchronizer类的实现原理

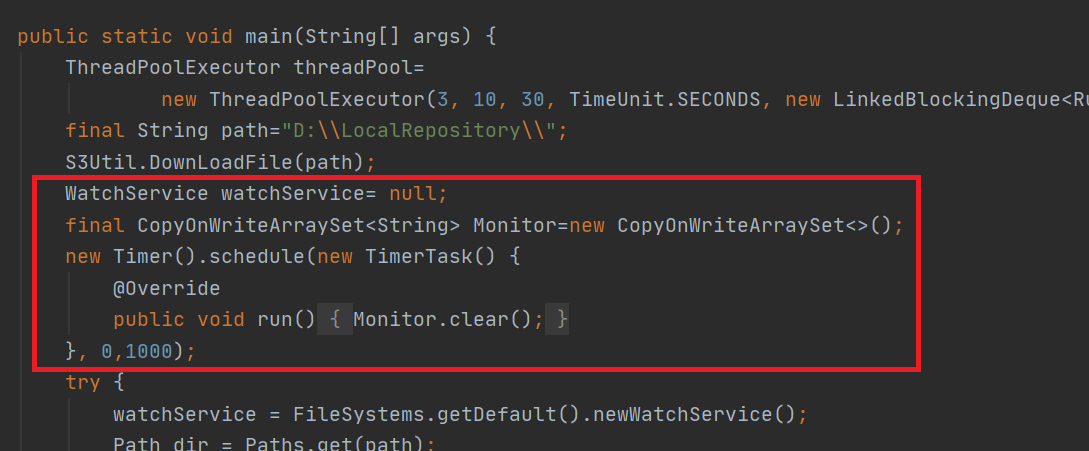
在main()方法中，定义了一个线程池，用于后面进行非阻塞的异步任务，由于本程序是IO型任务比较多，因此使用多线程可以提高程序的并发能力，提高文件同步的效率。



然后设置程序监控的本地文件夹，使用S3Util类的静态方法DownLoadFile方法先把远程仓库的文件同步到本地的文件夹。

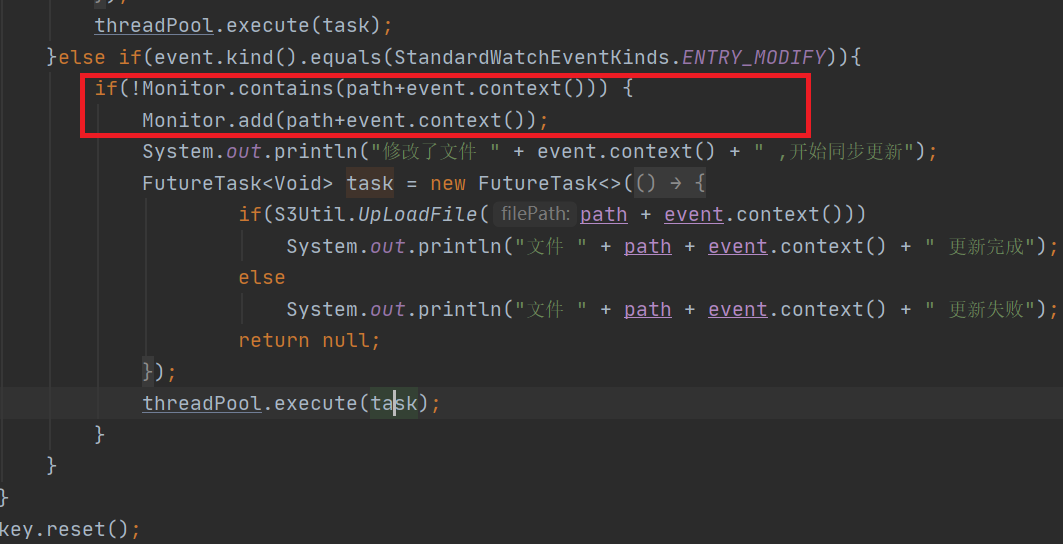


定义一个WatchService类变量，用于后面监控文件夹。然后实例化一个CopyOnWriterArraySet对象Monitor，这个变量是用于记录哪些文件发生了创建事件，防止创建的同时触发修改事件时，短时间内同一个文件多次上传，占用网络带宽。然后定义一个定时任务，每隔1s将Monitor的元素清空，回复文件修改的有效性。

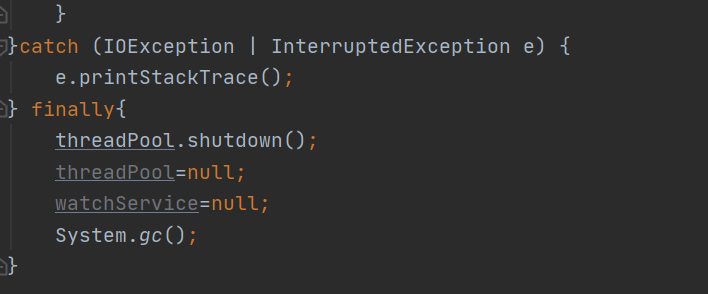


在try语句块中，实例化WatchService对象waterService，用于监听定义好的文件夹，然后注册监听的事件ENTRY\_CREATE, ENTRY\_DELETE, ENTRY\_MODIFY。在一个无限循环中，调用waterService的阻塞方法take(),开始监听文件夹内发生的事件，如果有ENTRY\_CREATE, ENTRY\_DELETE, ENTRY\_MODIFY事件的发生，将会触发相应的操作。在ENTRY\_MODIFY触发时，先会察看Monitor内是否还有文件对应的记录，如果没有则进行文件同步，有则判为无效，不同步。这个目的主要是创建文件时会在短时间内触发1次ENTRY\_CREATE和1~2次ENTRY\_MODIFY，导致同一个文件多次上传，所以必须过滤掉多余的上传操作。而定时器每秒会清空Monitor，恢复监听事件的有效性。文件同步操作都是交给线程池去执行，提高并发量。



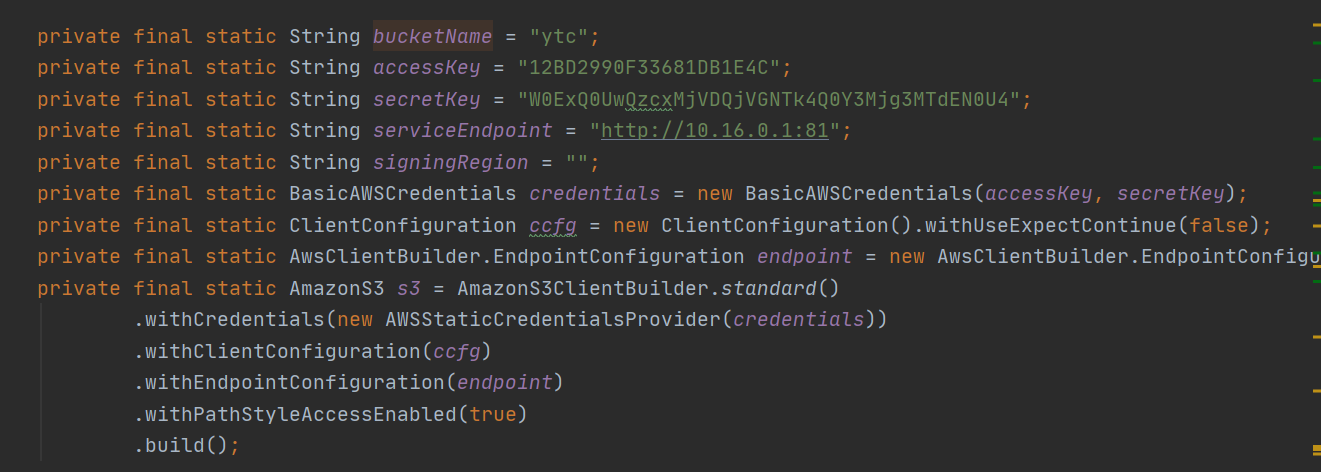


程序最后要关闭资源，进行垃圾回收。



1. S3Util类的实现

S3Util类数据域定义了一些静态常量，主要是作为连接S3客户端的参数。



S3Util的方法都是静态方法，因此在不需要实例化S3Util对象就能直接使用方法。主要的方法有文件上传UpLoadFile(...),文件分块上传MultipartUpLoadFile(...),文件删除DeleteFile(...),文件下载DownLoadFile(...),文件分块下载MultipartDownLoadFile(...)。其中主要的代码逻辑都和实验手册给的示例一样，这里主要介绍下载文件时冲突的解决方法。冲突主要有:

I)文件同名冲突。可以在文件后面加序号来解决。

II)文件相同冲突。这种情况不需要进行文件同步，利用文件名字，长度，MD5对比可以判断文件是否相同。

