简单文件同步器设计说明书

一、基本功能实现

1.普通下载download类

\* listObjects函数：列出桶中文件及其Etag值。

\* downLoad函数：下载文件并保存到本地。判断是否文件冲突，文件冲突就把原文件重命名，并重命名的文件上传到S3中。

\* downloadtofilepath函数：将桶中文件同步到本地目录中。调用listObjects函数，列出所有要下载的文件。文件的大小为判断条件，大于20MB调用分片下载函数MultipartDownload类的multipartdownload函数，否则调用downLoad函数。

2. 分片下载函数MultipartDownload类

\* multipartdownload函数，分片下载指定文件。同样的文件冲突处理。先设定每片的大小，下载每一片并保存到本地文件，最后将它们组合起来。

3.普通上传upload类

\* upLoad函数，将指定文件上传到对应的桶中。通过AmazonS3的putObject函数实现。

4.分片上传函数MultipartUpload类

\* multipartuplad函数，分片上传指定文件。先设定每片的大小，上传每一片，最后将它们组合起来。

5.删除桶中文件delete类

\* deleteFile函数，删除桶中指定文件。通过AmazonS3的deleteObject函数实现。

二、功能实现

1. 文件监听器 FileListener类

由文件监控类FileAlterationMonitor中的线程不停的扫描文件观察器FileAlterationObserver，如果有文件的变化，则根据相关的文件比较器，判断文件时新增，还是删除，还是更改。

\* onStart函数，启动监听。

\* onFileCreate函数，监测到文件夹中有新文件生成，判断大小，选择上传方式，将新文件上传到指定bucket中。

\* onFileChange函数，监测到文件夹中有文件被修改，判断大小，选择上传方式，将新文件上传到指定bucket中。

\* onFileDelete函数，监测到文件夹中有文件被删除，调用delete类的deleteFile函数删除桶中文件。

\* onStop函数，监听停止。

2. 监控Filemonitor类

\* filemonitor函数，根据给定的监控目录，设定轮询间隔。创建文件观察器，调用文件监听器 FileListener类设置文件变化监听器，创建文件变化监听器，启动监控器。

3.同步Synchronize类

\* getbucketname函数，通过输入得到指定的bucket名称。

\* getfilePath函数，通过输入得到指定的本地目录路径。

\* getfileName函数，通过调用File类的list函数得到本地路径的所有文件名。

\* synchronize函数，调用getbucketname函数、getfilePath函数和getfileName函数得到需要上传的文件，以及上传的bucket名称。根据文件大小选择调用不同的上传函数上传文件。

三、通过Main类main函数执行程序。

程序启动时，获取连接s3桶客户端。

1.把bucket中的文件同步到本地，调用download类的downloadtofilepath函数。

2.调用Filemonitor类的filemonitor函数，指定监测本地路径，监测到本地添加、修改了文件，新文件会上传到S3；本地删除了文件，会删除S3上对应的文件。

3.输入字符“Y”后，可指定本地某个目录与S3的某个Bucket进行文件同步。调用Synchronize类的synchronize函数。输入字符“E”可取消该功能。