编写程序实现文件的分布式存储。

系统的结构为

多个客户端程序，FileClient应用。客户端程序可以向文件服务节点应用程序上传文件，下载文件，删除文件。暂时不考虑文件夹功能，也不考虑文件名重名的问题。

一个FileServer应用（暂时不考虑多个FileServer应用），提供文件存储节点StorageNode的管理功能，提供文件的管理功能。

多个StorageNode应用，提供文件的存储能力。提供向备份节点服务器进行文件备份功能。

一个监控程序。

StorageNode应用程序提供文件的网络存储能力，在每个拥有硬盘文件存储能力的服务器节点上运行该程序。

该应用程序读取运行文件夹下的名称为storage\*.properties

文件，该文件的内容如下：

NodeName=aName

NodeIP=ip address

NodePort=aPort

RootFolder=aFolder

Volume=100GB

FileServerIP=file server ip address

FileServerPort=file server port

NodeName属性，存储节点的名称，命名要求唯一，例如Node1, Node2, Node3, Node4, ...

在大作业检查时，至少要启动4个存储节点进程。在运行过程中，还需要动态启动Node5, Node6两个进程，进行动态的存储扩展。

还需要在运行过程中动态的将一个节点关闭掉。NodeIP地址，由于没有多台计算机，所以NodeIP属性都设置为127.0.0.1

但是每个节点的端口不能设置为同一个，可以设置为8001， 8002， 8003， 8004， ...

文件夹也需要创建不同的文件夹，例如在某个文件夹下设置node1, node2, node3, node4这样的子文件夹。将这些文件夹设置为存储节点的根文件夹。

StorageNode需要知道FileServer服务器的地址信息，FileServerIP和FileServerPort就是UDP协议使用的地址和端口。

RootFolder就是存储节点服务器存放客户端上传文件的文件夹，Volume是该存储节点提供的最大存储能力。单位是MB, GB

需要创建文件夹来模拟不同的StorageNode服务器运行环境，文件夹下可以创建子文件夹来保存用户上传的文件。

在该文件夹下有可执行程序StorageNode

运行该文件java StorageNode config=storage\*.properties

或者使用当前文件夹下的配置文件 java StorageNode

StorageNode运行后，通过UDP协议向FileServer应用进行存储注册，通知FileServer应用，该存储节点可用。

StorageNode应用运行后，每隔一定的间隔时间向服务器发送工作正常的UDP通知包，如果FileServer很长时间收不到

该通知包，则认为该服务节点已经停止运行。StorageNode可以作为客户端向其备份StorageNode节点进行文件备份。

FileServer应用提供文件存储节点的管理。提供存储文件的管理功能。

FileClient在进行文件上传和下载时，必须通过FileServer来获取文件存储在哪个节点，哪个备份节点的一些文件信息。

FileServer管理每个文件都需要通过文件编号来完成，在其内存中通过Map集合来管理所有文件。

每个文件都需要在两个节点上进行1+1备份存储。FileServer分配存储节点需要考虑负载均衡。

在存储节点上存储的文件名称都是文件编号，禁止使用明文（原文件名），文件内容必须加密。备份节点上的文件和主节点上的文件内容和文件名称都必须一样。

上传成功后，FileServer需要将文件编号信息传递给客户端，客户端以后通过该文件编号进行文件下载，文件删除。在FileServer管理所有文件信息。文件编号采用UUID, UUID.randomUUID().toString()

在FileServer中需要管理所有存储节点信息。

FileServer在内存中需要管理后端FileStorage服务器的信息，包括名称，ip，端口，

容量，实际容量，剩余容量，文件数量，是否可用等信息。

文件信息包括：编号，文件原始名称，文件大小，主存储节点信息，备份节点信息，等等。

这些信息建议使用集合来进行存储，同时要求支持序列化到文件中。在FileServer启动后，需要读取这些序列化信息，服务器退出时，需要

保存这些序列化信息到文件中。

FileClient就是文件存储服务的客户端程序。提供如下运行方式

上传文件

java FileClient upload afile

该程序输出一个新存储文件的uuid，就是在服务器端保存的文件的唯一标识。

下载文件

java FileClient download uuid

删除文件

java FileClient remove uuid

FileClient应用程序运行时，需要读取配置信息，在配置中包含了FileServer的IP和端口。注意该端口是FileServer和FileClient之间的通信端口。

FileClient进行文件上传和下载时，首先建立到FileServer的TCP连接，获取主存储节点和备份存储节点的信息。然后断掉和FileServer的连接。

之后优先和主存储节点进行通信。当和主存储节点通信失败时，再和备份节点进行通信。如果和主节点进行文件操作成功，那么主节点负责和

备份节点之间的文件同步，例如文件备份，文件删除这两个功能。

需要编写一个监控程序，可以使用AWT, SWING, WEB(JSP)等任何方式，来远程访问FileServer中的文件信息和节点信息，建议使用两个表格来显示

存储节点服务器信息和文件信息。监控信息可以定时刷新，也可以实时刷新显示。

代码需要注意编码规范，类，函数，变量命名要符合规范。注意程序结构要合理，系统可扩展功能。演示时，需要在同一台计算机上模拟多台计算机，多个进程，

在存储节点上的文件名称和内容都需要加密。文件内容传输过程需要压缩。