**课程项目：**

参考git实现原理，使用Java实现blob、tree、commit核心存储结构

功能：

1. Commit：提交commit，可以使用”git log”查看commit历史
2. 回滚 使用”git reset” 回滚到指定分支
3. 分支 创建分支、实现分支间切换

第一周任务：

（1）文件的key-value：

实现key-value存储（封装为class）：

Key：文件名

Value：文件中内容（key根据value计算得出）

功能：

1. 给定value，向存储中添加对应的key-value
2. 给定key，查找得到对应的value

（2）文件夹的key-value存储：

遍历文件夹目录， 如果遇到子文件则转化为blob并保存

如果遇到子文件夹则递归调用内部的文件夹或文件，转化为blob与tree

并保存

（3）单元测试（unittest）

通过代码自动化的检测key-value存储实现是否正确，确保大量文件进行存储时的效率与正确率。

存储目标分析：

1. blob文件的value：文件内容

Key：value的hash值

1. tree文件夹的value：内部blob文件的key、每个子文件夹tree的key、子文件以及子

文件夹的名称

Key：tree文件夹value的hash值

实现设计：

1. hash类：计算文件blob、文件夹tree的hash值。

-计算字符串（string型）参数的hash方法

-计算file的hash方法

-返回得到的hash值

1. <优化后>KeyValueObject: blob、tree的父类，子类blob与tree继承自object类

计算key值

创建类型为blob类型的文件

创建类型为tree类型的文件

1. blob类：

计算该blob文件的key

创建以key命名的blob文件

1. tree类：

计算该tree文件的key值

以key命名的tree文件