程序设计文档

1. 题目要求

实现Commit

• 给定一个工作区目录，生成对应的blob和tree(上周已完成)以及commit

• 写代码之前先理清思路，写设计文档

• 提示：

需要存储指向当前最新commit的HEAD指针

每次新生成一个commit前，需要把根目录的tree key与已有的最新commit的tree key进行比较，发现不相同时（即文件发生了变动）才添加这个commit

1. 需求分析

本任务实质上是任务一和任务二的进阶版。它要求在已实现的底层逻辑之上实现Git功能的关键功能：Commit。我们可以总结出如下几个个功能点：

* 比较版本之间的异同：当需要传入新的目录或文件，获取当前的key并与之前版本的key比较，如果更新才写入。
* 将文件夹和其内容的commit返回：
  + 包含根目录tree对象的key
  + 包含前一次commit的key
  + 以上两行构成本次commit的value，本次commit的key就是以上两行内容的哈希.
* 存储当前版本的指针HEAD:保存一个HEAD文件，其中的内容为最后一次commit对象的地址。

1. 设计思路
2. 得到新路径
3. 判断是否为第一次更新：
   1. 如果是，根据commit规则，生成本次commit值，指针HEAD指向当前commit对象；
   2. 如果不是，比较前一版本和新版本异同
      1. 如果相同，指针HEAD不变
      2. 如果不同，生成本次commit值，指针HEAD指向当前commit对象；
4. 返回commit的key。
5. 详细设计

* 类commit：commit功能的实现类
  + 数据域：
    - filePath ：传入的文件夹目录
    - HEAD：指针文件地址
    - commitKey:最终返回的key值
  + 方法：
    - gen\_commit:生成本次commit。根据state状态来执行两种方案：前一次为空；前一次不为空。调用teamtask2()生成tree的key,然后合并再调用gen\_hash().
      * 参数：state（是否为第一次的状态，布尔类型），filePath（根目录），HEAD（前一次commit的指针）
      * 返回：commitKey
    - isFirst:类型为布尔，判断是否为第一次提交。判断HEAD是否为空或者不存在即可。
      * 参数：HEAD
      * 返回：true or false.
    - isSame: 类型为布尔，判断两版本是否相同。
      * 参数：tree \_value1,tree\_value2;
      * 返回：true or false.
    - Process:处理流程。给commitKey赋初值为之前版本的key.调用isFirst（），如果是第一次调用gen\_commit（），HEAD更新内容并保存，返回commitkey；如果不是，调用isSame（）判断tree的key异同，相同则返回commitkey.不同则根据此状态调用gen\_commit（），HEAD更新，返回commitkey。
      * 参数：commitKey，HEAD，filepath
      * 返回：commitKey。