

Git的使用

a.如何创建一个新的 git 分支？如何进行分支切换？如何删除一个分支？什么时候可以安全的删除一个分支？

- 创建一个新的git分支可以使用
`git branch <newBranch>`
- 切换分支可以使用
`git checkout <newBranch>` 或者 `git switch <newBranch>`
也可以使用
`git checkout -b <newBranch>` 或者 `git switch -c <newBranch>` 同时创建并切换分支
- 删除分支可以使用
`git branch -d <newBranch>`
- 强行删除分支可以使用
`git branch -D <newBranch>`
- 当该分支的修改已经合并到其他分支后可以安全地删除当前分支

b.如何撤销保存在暂存区的修改？如何仅撤销最近一次提交的某一个文件(test.txt)？

- 使用 `git reset HEAD <file>` 可以撤销保存在暂存区的修改，将文件从暂存区移回了工作区
- 使用 `git reset --hard HEAD^` 可以回到上一个版本的已提交状态
- 使用 `git reset HEAD~1 -- test.txt` 可以撤销最近一次提交中的test.txt文件

c.解释 git fetch 和 git pull 的区别

- `git fetch` 从远程库下载最新代码，并不自动合并到本地分支，而是将最新代码同步到本地仓库的一个特殊分支
- `git pull` 可以从远程仓库获取最新代码并自动合并到本地分支，相当于 `git fetch` 和 `git merge`

Debug

1. 为什么用 main.cpp:30L 替换 main.cpp:31L 会出现报错？

error: use of deleted function 'Student::Student()'

在 Student.hpp 函数中，`Student() = delete` ,函数 `Student::Student()` 已经被删除，不能再被使用

2. 为什么用 main.cpp:32L 替换 main.cpp:33L 会出现报错？

Candidate constructor not viable: cannot convert from base class pointer 'Human *::value_type' (aka 'Human *') to derived class pointer 'Student *

在 main.cpp L11，`check`函数的参数类型为 `Student*`，但是`vec`中第一个元素为 `Human*` 类型的数据，需要先将基类指针转化为子类指针，即需要先进

行 `auto student1 = static_cast<Student*>(vec.back());` 的类型转换后，将 `student1` 作为 `check` 的参数，而在main.cpp:32L中仍使用未转化的指针作为参数

3. 为什么取消注释 main.cpp:35-39L 不会报错，但运行会出现段错误？

- 在 main.cpp:35-39L 内，并未出现语法错误，段错误通常因为程序非法访问内存产生，在运行时进行访问内存，在编译时不会出错，所以运行前不会报错。
- 利用ASAN定位段错误得到 `heap-use-after-free on ...` 的报错信息，所以是在堆被释放后仍在使用导致的错误。`list`列表只有一个元素，在遍历`list`列表时，访问该元素时，调用 `remove` 函数删除了所有与该元素相同的元素，减小了列表大小，改变量列表结构，使列表为空，列表中元素所占的内存空间被释放，而指向该内存的指针并未被清除，从而出现段错误。