# 高级课 -3rd

## 上午

1. coredate 版本的迁移(基于,创建新版本) (add model version)
2. coredata 关系的建立
3. 创建新的entity类
4. 添加新的界面,添加班级,获取班级信息
5. 点击班级进到下个学生界面,通过传递班级信息获取到该班级的学生列表.(通过断言查找,同时配合sord排序-->NSPredicate和NSSortDescriptor)
6. 添加学生时,获取到当前班级,学生的班级属性设为当前班级实体(其实数据库中存的是班级的ID)

## 下午

# cocoaPods

## *CocoaPods, the Cocoa library package manager.*

1. 检查是否安装了cocoapods(pod --version),若安装则不用一下步骤
2. 检查ruby环境(ruby -v) 若没有ruby环境,装ruby环境.
3. 换cocoapads的镜像(其实就是下载地址,国外的访问不了(注意:网址不要写错了)
    1. gem sources --remove https://rubygems.org/
    2. gem sources - a http://ruby.taobao.org/
4. 进行安装cocoapods sudo gem install cocoapods

## pods命令行

1. pod search

2. pod install

# cocoapods使用

1. 创建podfile(第三方库的名字的集合),进入项目文件夹下进行创建文件夹,(注意:没有后缀,名字就是Podfile)
2. 找到要使用的第三方的,看他的使用说明,如何在cocoapods中使用,或者使用pod search命令进行关键字查找
3. 将找到的第三方pod 使用语句写入podfile中,(将podfile中的全部删掉后写入)
4. 执行pod install方法
5. 关掉xcode中的项目,然后打开.xcworkspace的项目
6. 需要添加第三方库时,只需要将podfile文件中加入pod 使用语句,然后执行pod update命令即可

# 插件使用

https://github.com/kattrali/cocoapods-xcode-plugin

# git

# 常用命令

1. git clone url
2. git init
3. git config
4. git add file name
5. git commit -a "info"
6. git push push当前分支到远程分支
7. git branch 查看所有分支
8. git checkout 分支 ,切换到某个分支
9. git status
10. git log -oneline
11. git reset 撤销本地修改
12. git clean 同上

13. git rebase 重写历史记录
14. git reflog 写历史记录
15. git branch
16. git merge
17. git checkout (remote commit branch)
18. git checkout -b创建一个新分支
19. git remote
20. git fetch
21. git pull
22. got push
23. git tab -a 0.2 -m "这是一个tag" master
24. git push origin master --tags

# git工作流

1. Centralized workflow(同svn)
2. feature branch workflow(按照特性功能来分支)
3. gitflow workflow(功能分支,发布分支,维护分支)
    1. 开发分支: 特性分支在develop分支上,其中一个分支完成后,进行合并到develop分支上,不与master分支进行直接交互
    2. 释放分支: 主要用来清理释放,测试和更新文档,当开发分支有足够的功能发布是,可以衍生一个release分支,准备好了上架,就可以合并到master并标记一个版本号
    3. 维护分支: 快速给发布产品修复bug或者微调功能,从master分支直接衍生出来.修复后要合并回master合develop分支
4. forking workflow

# 重要特性

1. 文件三种状态(modified,staged,committed)
2. 近乎所有操作都是在本地操作
3. 时刻保持数据完整性

# 基本流程

1. 在git版本控制的目录下修改某个文件
2. 使用git add命令对修改后的文件快照,保持到暂存区
3. 使用git commit命令提交更新,将保存在暂存区域的文件快照仍旧存储到git目录中

-- git pull = git fetch + git rebase