

# 第一次实验

## 实验目标

- 学习使用 `git` 相关操作
- 通过代码实例了解表达式的简单建模

## 任务一：git 学习

### step 0 git 安装与配置

#### git 的安装

##### Linux

```
sudo apt-get install git
```

##### Mac OS X

从 AppStore 安装 Xcode，运行 Xcode，选择菜单 "Xcode" -> "Preferences"，在弹出窗口中找到 "Downloads"，选择 "Command Line Tools"，点 "Install" 即可完成安装。

##### Windows

在[官网](#)上选择最新版本进行下载安装。

#### git 的配置

```
git config --global user.name "你的名字"  
git config --global user.email "21xxxxxx@buaa.edu.cn"
```

利用上述指令配置 git，注意将“你的名字”和“[21xxxxxx@buaa.edu.cn](#)”分别替换成你真实姓名和北航邮箱。

#### 配置 ssh key

参考[官网](#)上给出的介绍说明进行 ssh key 的配置

### step 1 新建仓库

在本地新建一个空文件夹，在此目录下打开终端（bash/git bash/powershell/...）

输入

```
git init
```

新的文件夹中多了 `.git` 文件夹，说明你得到一个新的 `git` 仓库

## step 2 关联远程仓库

目前为止，`step 1` 建立的文件夹还是一个空文件夹，需要关联两个远程仓库，首先考虑第一个远程仓库

第一个远程仓库是课程组提供的远程公共仓库 `share`

```
git remote add share  
git@gitlab.oo.buaa.edu.cn:oo_course_2023/unit1/exp1_code.git
```

## step 3 从远程仓库拉取文件

关联后可以从远程仓库拉取文件

执行命令

```
git pull share main
```

从 `share` 仓库的 `main` 分支下得到待补全的代码。

## step 4 关联个人实验 1 仓库，并尝试一次提交

```
git remote add origin 你的个人实验1的远程仓库链接  
git add .  
git commit -m "Init commit"  
git push -u origin main
```

关联到远程仓库 `origin`，并尝试提交（其中个人实验 1 仓库的名字为 `homework_2023_你的学号_exp_1`）

至此，任务一的内容已全部介绍完毕，如果按照上述步骤操作后得到预期结果（例如可以正常 `pull` 和 `push`），即可开始进行任务二。

## 任务二：基础的表达式运算任务

第一单元需要同学们完成含参数表达式的解析工作，为了帮助同学们更好的完成第一单元的作业，我们提供了代码框架（即任务一中克隆下来的代码），你需要补全这份代码。同学们可以新建一个项目，将任务一待补全的代码复制进来，补全代码后再将原文件夹的待补全代码文件替换掉，完成后再次执行上文中 `add`、`commit`、`push` 操作提交即可。

### 问题描述:

给定含有变量的表达式和变量的取值，要求实现表达式的解析并输出正确运算结果。

## 实验要求:

代码中需要补全的部分已经用 `TODO` 标出, 请根据题意完善代码, 并提交到个人实验 1 仓库中以完成本次实验

## 官方包解读

```
src
| - MainClass.java // 程序入口, 负责读入变量、表达式和输出最终结果
| - Parser.java // 负责表达式的解析
| - | - parse() // 解析表达式
| - | - findAddOrSub() // 工具方法, 找到表达式中最右边的加号或者减号的位置
| - | - findMul() // 工具方法, 找到表达式中最右边的乘号的位置
| - Add.java // 加法类
| - Sub.java // 减法类
| - Mul.java // 乘法类
| - Num.java // 数字类
| - Operator.java // 上面四个类的基类
```

## 输入格式:

第一行为一个非负整数  $n$ , 代表有  $n$  个变量。

接下来的  $n$  行, 每行一个变量和它的赋值, 中间用空格隔开。

最后一行为需要解析运算的含有变量的表达式。为了简化处理, 本行的表达式只涉及 `+`, `-`, `*` 三类运算, 没有括号, 数字不含前导零与连续的加减/正负号。

## 输出格式:

输出正整数, 代表解析后的表达式结果。

## 样例:

stdin:

```
2
x 1
y 2
x + y - 2 * x
```

stdout:

```
1
```

## 参考资料

1. [Git使用心得 & 常见问题整理](#)
2. [Git 廖雪峰教程](#)