

# GPA 计算器

---

一个更高级的GPA计算器。

本文将从以下几个方面介绍这个程序的设计思路和整体结构：

- 使用方法
- 主要功能和特色
- 整体架构与模块介绍
- 文件结构
- 不足和改进

## 使用方法

---

### 需求

- python 3.X
- terminal (OS X 或 Linux) / cmd (windows)

### 运行

在 `终端` (OS X 或 Linux) 或 `命令行` (windows) 下，进入文件所在目录，然后输入：

```
python main.py
```

即可运行程序。

## 主要功能和特色

---

### 主要功能

1. 录入成绩只需粘贴成绩单字符串至窗口中
2. 查询所修学分数
3. 查询平均学分绩（保留两位小数）
4. 查询绩点高于平均学分绩的科目

### 特色

1. 允许用户与控制台的交互，进行命令解析与合法性检验
2. 允许对成绩单字符串的部分内容进行修改
3. 允许多次添加不同的成绩单字符串
4. 增加窗口界面，涉及两个前端和同一个后端交互

# 整体架构与模块介绍

本程序的整体架构分为三大层，从下到上一次是 `Calc` `GUI` `Shell`，它们是依次包含关系。

- `Calc` 模块负责核心功能的实现。它负责对成绩单字符串进行处理并生成字典列表，在此基础上实现统计所修学分、计算平均学分、统计绩点高于平均学分的科目等功能。
- `GUI` 模块负责用户录入与修改字符串成绩单时的窗口实现。它的功能主要为根据用户输入的命令显示一个文本框，将用户录入或更改后的信息传递给 `Calc` 的实例，以便后续操作。
- `Shell` 模块负责命令行UI，它是核心功能、窗口界面与用户之间的桥梁。

由于它们都是相对独立的模块，因此将它们的实现文件均置于 `src` 文件夹中。

## 文件结构

### 根目录

- `src`：所有模块的实现文件，包括：
  - `calc.py`： `Calc` 模块的实现文件
  - `GUI.py`： `GUI` 模块的实现文件
  - `Shell.py`： `Shell` 模块的实现文件
- `doc`：说明文档，包括：
  - `README.md`：说明文档的 `md` 格式文件
  - `README.pdf`：说明文档的 `pdf` 格式文件
  - `input.txt`：输入文件（成绩单字符串）
- `main.py`：打开程序时需要运行的文件

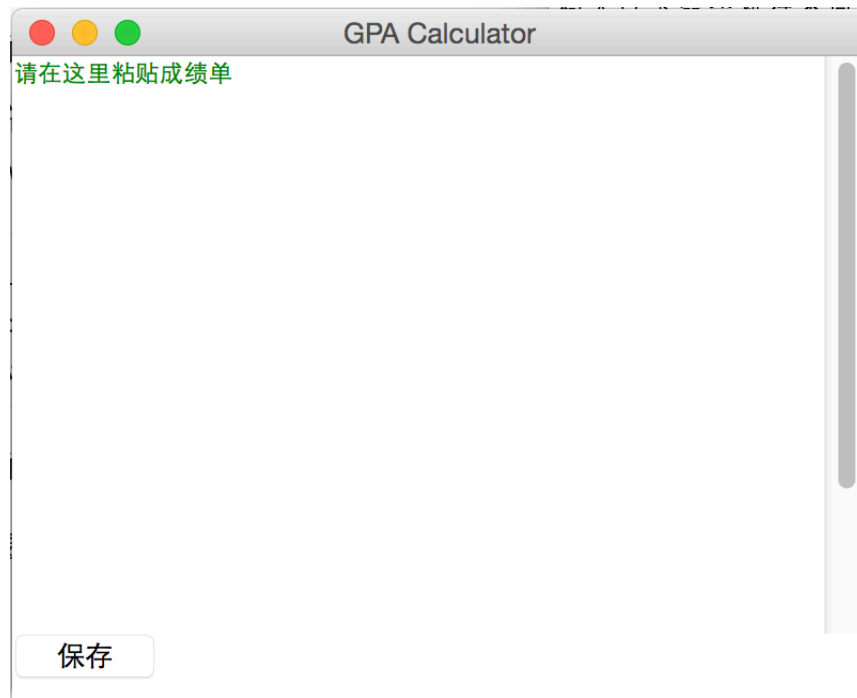
## 不足和改进

1. 不支持多语言界面。
  - 可能的解决方案：实现界面多语言框架。
2. 命令行界面不支持切换到历史命令等高级功能。且现在 `Shell` 中命令解析部分的实现依然是传统命令式编程的if-else风格。
  - 可能的解决方案：学习使用命令窗口专用函数，设计新的框架或套用已有框架。
3. 在 OS X 系统下，录入成绩单并保存后会出现一小段时间的无响应状态，并在后续打开窗口时出现以下内容。但在 windows 系统下就不会发生该情况。

```
IMK Stall detected, *please Report* your user scenario in
<rdar://problem/16792073> - (imkxpc_bundleIdentifierWithReply:) block
performed very slowly (XX secs)
```

- 可能的解决方案：询问老师、同学或助教。
4. 在 OS X 中窗口显示正常，如图一。但在 windows 系统下窗口的细节会出现问题，如图二。
    - 可能的解决方案：学习 `tkinter` 模块的高级功能，优化代码。

图一



图二

