

# 技术报告

具体项目信息见readme.md，此处主要汇报一下项目设计。

## 一、项目介绍：

项目实现了一个名为 `minimatrix` 的纯python矩阵库（详见 `minimatrix.py`），实现了二维矩阵的储存和加、减、按位乘、矩阵乘、求逆、转置、Kronecker积、方幂、求秩、求行列式和初等行变换、高斯消元等矩阵基本运算。在此基础上此库实现了各种运算符重载以及一系列函数实现了一定程度上的易用化。

## 二、项目架构：

考虑到各个函数之间存在一定的互相调用与依赖关系，秉承着尽可能降低各个函数的耦合度的理念，本库实现了两个类，分别是名为 `Matrix_base` 的基类——其中实现了各种最基本的且会被反复调用的功能，并在此基础上派生出了实现具体业务逻辑的 `Matrix` 类。

## 三、相关技术

开发过程中整合了 `git` 用于分布式版本管理（最初也打算用这个来实现多人并行开发，但最后没有来得及找到人）与 `pytest` 单元测试框架（详见 `unit_test.py`）对库进行测试，结合了 `numpy` 对计算结果的准确性进行了测试。在库的编写上使用的相对高级的语法功能有：

1.装饰器。装饰器主要用于向量化函数部分以及单元测试模块中的自动重复测试以及性能估算部分。使得可以用相对简洁的功能实现一些相对复杂但是固定的逻辑。

2.类的继承与派生。在整个库的设计上通过简单的基类和派生类的分割实现的代码的解耦与重用，使得代码在逻辑上更加清晰而富有层次感。

3.列表推导、三目运算、模式化结构匹配、f-string等。代码中通过对这些特性的灵活使用，在保证代码可读性的同时让代码更为简洁优雅。

## 四、关于此项目

组长&架构设计&代码编写&代码测试&DEBUG&报告撰写：LWT

项目规模：整个项目大约550行代码，用时大约两天