**成绩**



**本科生《编译原理》课程实践报告**

**题　　目： 程序设计语言认知实验**

**学 院： 徐特立学院**

**专业名称： 计算机科学与技术**

**姓 名： 陈照欣-1120191086**

目录

[实验目的 3](#_Toc97220821)

[实验内容 3](#_Toc97220822)

[环境配置 3](#_Toc97220823)

[实验的具体过程和步骤 3](#_Toc97220824)

[实验截图 3](#_Toc97220825)

[实验结果 5](#_Toc97220826)

[语言易用性和程序规模对比分析 6](#_Toc97220827)

[实验心得体会 6](#_Toc97220828)

实验目的

了解程序设计语言的发展历史，了解不同程序设计语言的各自特点；感受编译执行和解释执行两种不同的执行方式，初步体验语言对编译器设计的影响，为后续编译程序的设计和开发奠定良好的基础。

实验内容

对于矩阵乘法这一功能，具体要求为：对于给定的矩阵规模，自动生成两个随机整数矩阵，计算两矩阵的乘积，输出结果。分别使用C/C++、Java、Python和Haskell实现，对这几种语言实现的编程效率、程序规模、程序的运行效率进行对比分析。

环境配置

处理器 Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz

机带 RAM 12.0 GB (11.8 GB 可用)

设备 ID FDE9AABD-B6A6-4EB0-9805-DE4193E9D97B

产品 ID 00331-10000-00001-AA098

系统类型 64 位操作系统, 基于 x64 的处理器

笔和触控 没有可用于此显示器的笔或触控输入

gcc (tdm64-1) 9.2.0

python 3.8.8

openjdk 11.0.14.1 2022-02-08

OpenJDK Runtime Environment Temurin-11.0.14.1+1 (build 11.0.14.1+1)

OpenJDK 64-Bit Server VM Temurin-11.0.14.1+1 (build 11.0.14.1+1, mixed mode)

GHC 8.8.4

Visual Studio Code 1.65.0

实验的具体过程和步骤

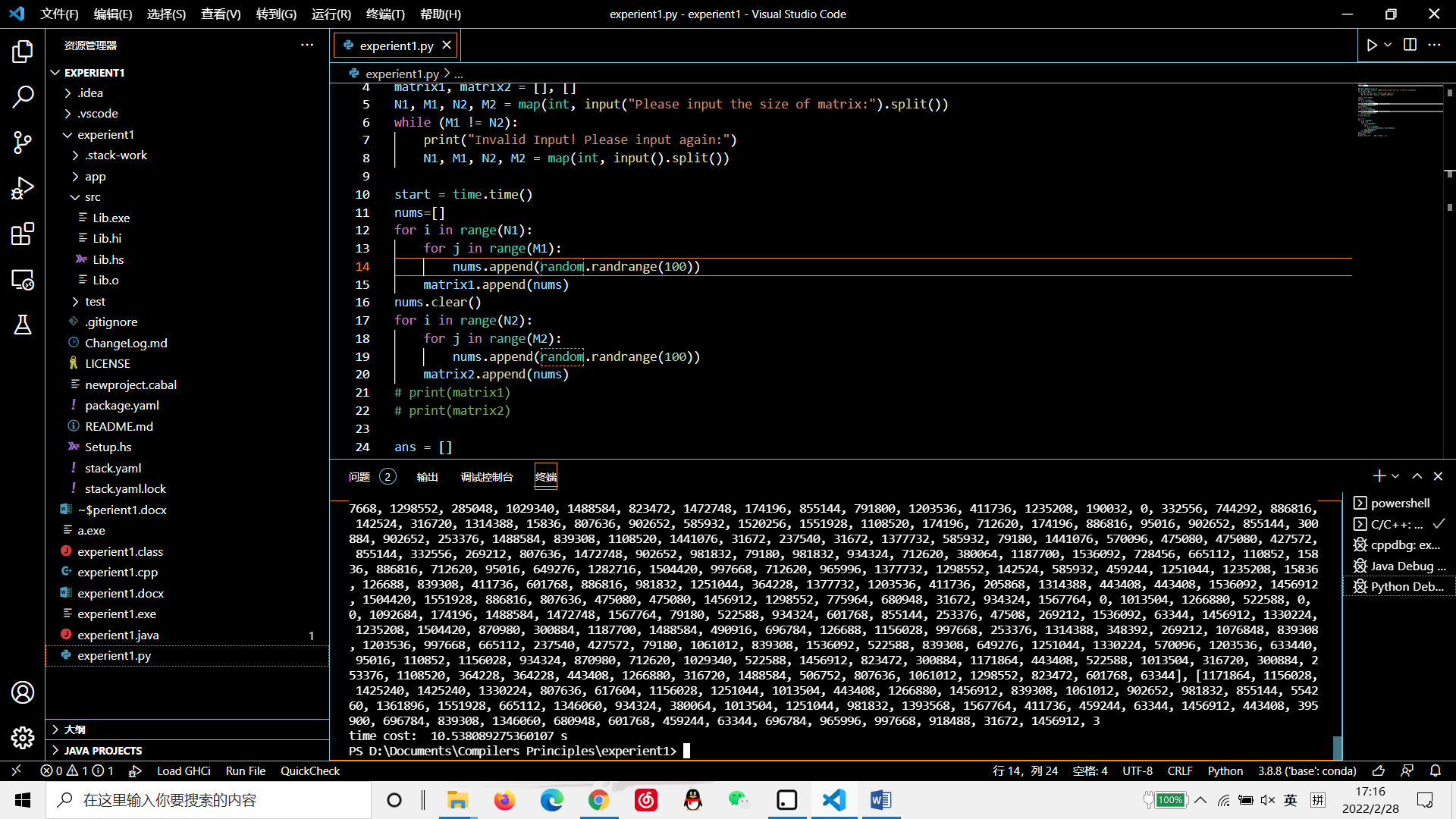
Step 1：设定矩阵的规模分别为200\*300和300\*400，每个元素的范围为1~100；

Step 2：规定程序运行时间为输入矩阵规模后到结果输出完成之间的时间间隔；

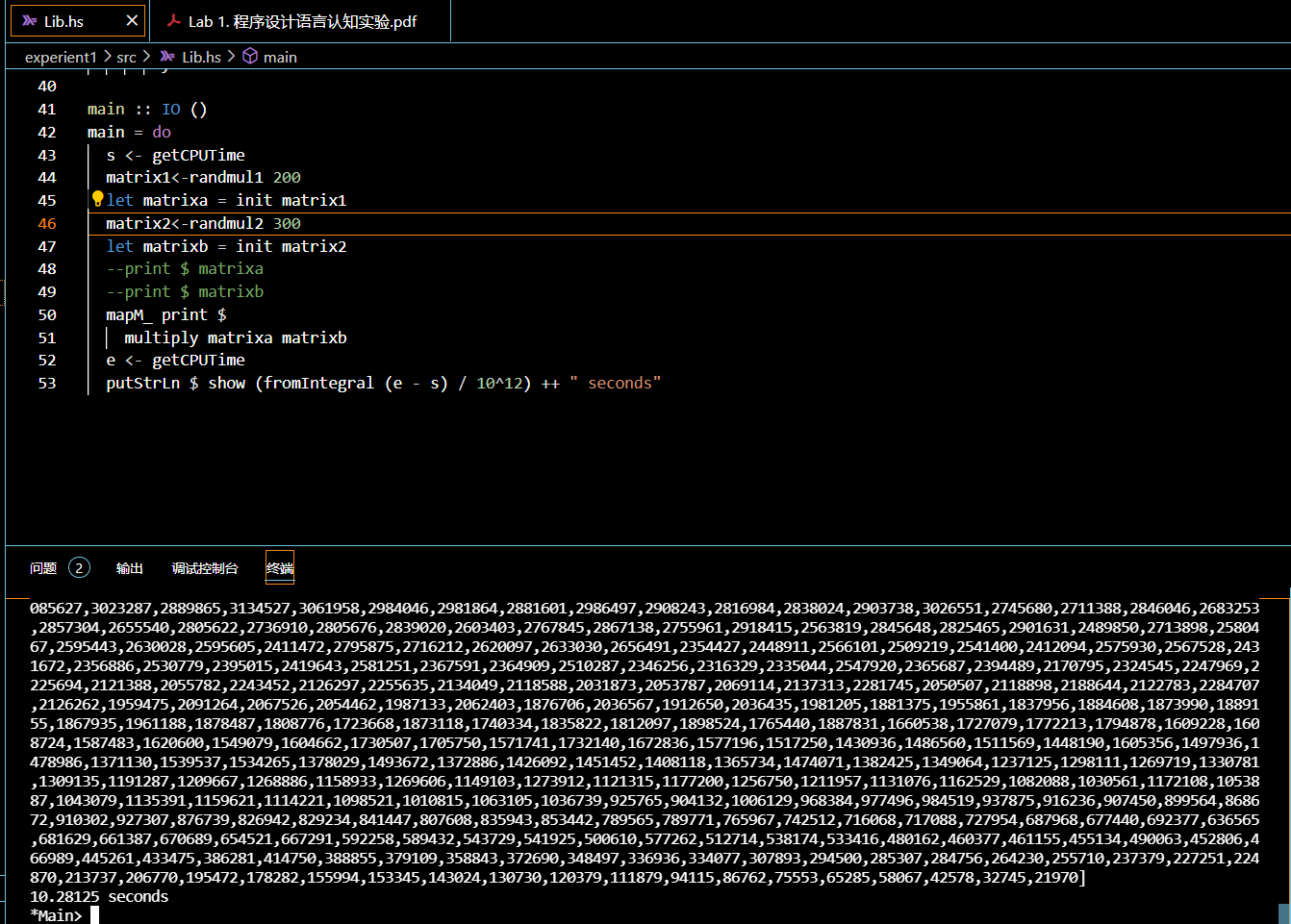
Step 3：对于每一个程序，执行5次，记录运行时间以及程序占用的内存大小求平均值

实验截图









实验结果

C++:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 第一次 | 第二次 | 第三次 | 第四次 | 第五次 | 平均 |
| 运行时间 | 187ms | 170ms | 183ms | 200ms | 179ms | 186ms |
| 占用内存 | 32.5MB | 34MB | 32.5MB | 31MB | 25MB | 31MB |

Python

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 第一次 | 第二次 | 第三次 | 第四次 | 第五次 | 平均 |
| 运行时间 | 6.15s | 5.48s | 5.52s | 5.75s | 6.04s | 5.79s |
| 占用内存 | 28MB | 25MB | 18MB | 23MB | 30MB | 25MB |

Java

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 第一次 | 第二次 | 第三次 | 第四次 | 第五次 | 平均 |
| 运行时间 | 50ms | 45ms | 46ms | 45ms | 54ms | 48ms |
| 占用内存 | 80MB | 76MB | 79MB | 77MB | 83MB | 79MB |

Haskell

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 第一次 | 第二次 | 第三次 | 第四次 | 第五次 | 平均 |
| 运行时间 | 10.32s | 9.98s | 10.20s | 10.48s | 9.11s | 10.01s |
| 占用内存 | 690.97MB | 670.88MB | 701.32MB | 705MB | 669.17MB | 687.47MB |

语言易用性对比分析

1. C 语言是一个通用的命令式计算机程序设计语言，属于静态强类型，支持结构化程序设计，是操作系统、数据库和嵌入式等系统级软件开发经常使用的编程语言。C语言能以简易的方式编译、处理低级存储器。C语言是仅产生少量的机器语言以及不需要任何运行环境支持便能运行的高效率程序设计语言。尽管C语言提供了许多低级处理的功能，但仍然保持着跨平台的特性，以一个标准规格写出的C语言程序可在包括类似嵌入式处理器以及超级计算机等作业平台的许多计算机平台上进行编译。C 语言是一个命令式面向过程的编程语言，对应的高级语言程序很容易被翻译到机器语言执行，相对其他语言执行效率较高。
2. 由于C++更接近底层的特性，C++的语法较Java更加复杂。Java和C++虽然都有类的概念，但是Java的功能更加丰富，且不受平台限制。Java 继承了 C 和 C++的很多语法，但是抛弃了较多的低层次的内容，例如指针和内存操作等。Java 中提供了垃圾收集器，编写程序时不需要再手动删除创建的对象。
3. Python有丰富的库可以调用，且支持算法测试，开发速度更快，适用于后端开发、数据科学、机器学习等领域。但是运行时间相对较长，且对解释器的要求较高。
4. Haskell 是静态强类型系统，是一个标准化的纯函数式编程语言。主流实现GHC既支持解释执行又支持翻译执行，因此效率比python略高。作为带类型的函数语言，适用于数据处理, 符号计算，编译器。但是由于其独特的语法和国内较小的使用范围，Haskell对于新手不太友好。

程序规模对比分析

python，c++实现的规模近似，都需要数据的定义，三层for循环实现结果，规模大致在15-20行。在代码量上haskell的优势非常明显。

实验心得体会

编程语言的选择通常依赖于开发速度、执行速度、鲁棒性、受众面等多个因素。不同领域对这些因素的要求不同，适合的语言也不同。对于操作系统、网络、硬件驱动等接近底层的开发，靠近硬件的C比较适合。金融电商公司等对于代码稳定性的要求较高，易于跨平台和代码优化的Java是更好的选择。此外，由于C/C++和Java的对编译环境的要求较低，编译器较为简洁，且语法通俗易懂，故更适合新手学习。在科研领域，需要各种强大的工具包，那么可以选择python。Haskell的复杂是其能很好的控制大型程序，在设计大型程序时，用Haskell语言能较容易的把握核心复杂度。Haskell的复杂实际上是大型程序本身的复杂度在程序语句上的直接体现。

附件

Lab1\_haskell

Lab1\_java

Lab1\_C++

lab1.py