

# internal软工一系统

- [门户系统](#)
  - [地址](#)
  - [注册/登录](#)
  - [选课](#)
  - [完成任务](#)
- [Coder系统](#)
  - [地址](#)
  - [完成任务](#)
- [Git答题参考流程](#)
- [可能遇到的问题](#)
- [一些FAQ](#)

## 门户系统

### 地址

[p.internal-paas.seec.seecoder.cn](http://p.internal-paas.seec.seecoder.cn)

门户包括用户注册、登录、选课、参与测试等功能，且与Coder数据同步。

### 注册/登录

注意：密码应该不少于8位且不能过于简单，否则将无法通过注册。

### 选课

在课程界面找到相应课程，点击确认选课，输入正确的选课码即可完成选课，在门户完成选课  
后，也会自动完成Coder系统的选课。



### 软件工程与计算 I 2024

进行中

教师: liuqin  
季度: 2024春  
时间: 2024.02.26至2024.06.30

编程基础、面向对象编程

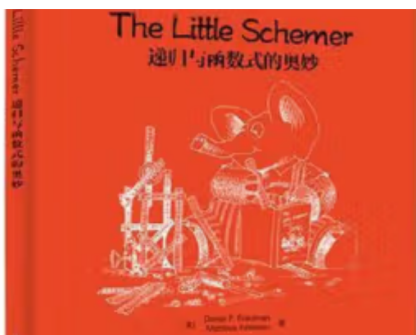


### 软件工程与计算 II 2024

进行中

教师: liuqin  
季度: 2024春  
时间: 2024.02.26至2024.06.30

软件工程基础



## 软件工程与计算 I 2024

教师: liuqin  
学期: 2024春  
时间: 2024-02-26 00:00至2024-06-30 00:00

MOOC链接: [点击此处](#)

确认选课

## 完成任务

点击下方的任务进入考试，点击参加考试即可参与考试。

任务

课件

通知

评论

项目

团队

刷新



未参加 00-HelloWorld

2024-02-26 00:00 — 2024-03-04 00:00

### 00-HelloWorld

参加考试

生成自测试卷

编号: 207

教师: liuqin

课程: 软件工程与计算 I 2024

状态: 进行中

创建时间: 2024/02/26 17:47

起止时间: 2024/02/26 00:00 至 2024/03/04 00:00

参与完成后即可看到考试的具体内容，包括Git地址与提交记录等，可以选择使用Git作答或在线作答（目前门户在线作答正在上线，可以暂时使用Coder系统的在线作答），Git作答或在线作答方式尽量只使用一种方式作答以避免冲突，推荐使用Git+本地IDE的方式进行答题。

编号: 207

教师: liuqin

课程: 软件工程与计算 I 2024

状态: 进行中

创建时间: 2024/02/26 17:47

起止时间: 2024/02/26 00:00 至 2024/03/04 00:00

试卷地址: <https://gitlab.internal-paas.seec.seecoder.cn/testUser/826196c2723b41478f7e2a57b30229b5> 复制

## 题目列表

HelloWorld Python &gt;

## 提交记录

#	提交时间	状态	成绩	操作
---	------	----	----	----

No Data

提交完成后可以查看提交信息，点击详情可以查看错误用例以及控制台输出等。

HelloWorld Python &gt;

## 提交记录

#	提交时间	状态	成绩	操作
64	2024/03/01 16:29	运行结束	100	∨

题目	状态	分数	操作
HelloWorld	所有用例通过	100	<a href="#">详情</a>   <a href="#">停止</a>

## Coder系统

## 地址

[coder.internal-paas.seec.seecoder.cn](https://coder.internal-paas.seec.seecoder.cn)

## 完成任务

Coder系统的操作与门户系统类似，可以参加考试、查看考试具体内容、使用Git做题或在线做题。

Coder中点击在线做题进入在线做题页面。

# 00-HelloWorld



在线做题

编号: 207

创建者: kenny67nju

创建时间: 2024/02/26 17:47

成绩: 0

状态: 正在进行

起止时间: 2024/02/26 00:00 至 2024/03/04 00:00

试卷地址: <https://gitlab.internal-paas.seec.seecoder.cn/testUser/826196c2723b41478f7e2a57b30229b5>

复制

提交记录

题目列表

#	提交时间	状态	成绩	操作
暂无数据				

进入在线做题页面，修改/src下的源文件即可进行作答，/test为本次考试的一部分测试用例文件夹，/readme.md为本次考试的文档。完成修改后即可点击保存并测试完成提交。

题目1: 26-helloworld

离结束还有: 2天07小时33分27秒

退出

src

\_\_init\_\_.py

hello\_world.py

test

requirements.txt

README.md

1

"""write your code in method hello\_world"""

2

def hello\_world(self):

3

return

编程完毕后，点击下方“保存并测试”按钮运行当前题目的代码；在考试详情页的“提交记录”中查看代码运行情况

保存并测试

提交完成可以查看提交记录，点击详情可以查看错误用例以及控制台输出等。

提交记录

题目列表

#	提交时间	状态	成绩	操作
64	2024/03/01 16:29	等待或运行中	0	<a href="#">详情</a>

提交结果包括如下结果。

### 查看在线测试结果

作答提交后将会被放入运行队列中依次运行测试，当测试完成后可以查看结果，测试结果可能有以下几种状态：

等待运行

表明该题正在运行队列中等待进行测试

正在运行

表明该题正在进行测试

所有用例通过

表明该题的所有用例均通过测试或测试/变异覆盖率达到100%

有用例未通过

表明该题的部分/所有用例没有通过测试或测试/变异覆盖率达到100%

编译或运行失败

表明该题所提交的代码不能正常进行编译或运行

运行超时

表明该题所提交的代码能正常进行编译或运行但进行测试时超过了时间限制

对于“有用例未通过”状态，可以查看详情中各未通过用例的具体报错；对于“编译或运行失败”状态，可以查看详情中的控制台输出分析问题；对于“运行超时”状态，可以考虑死循环等代码问题

## Git答题参考流程

### 基本流程

参加考试 → Clone试卷 → 本地作答/测试 → Push试卷 → 查看在线测试结果 → 修改作答/本地测试 → Push试卷 → .....

### 参加考试

在考试列表页面找到所要参加的考试，点击“参加考试”，在弹出的对话框中确认信息后点击“确定”，等待系统创建试卷完成即表明成功参加考试

### Clone试卷

在本地命令行/Git Bash中使用 `git clone [试卷地址]` 命令或使用IDE集成的Git插件将试卷clone到本地，注意试卷地址只会在考试开始后发放，在考试列表及考试详情页面均可查看

### 本地作答

在本地使用任意IDE/编辑器对试卷进行作答，请务必在每次修改后进行保存；试卷中包含部分测试用例的题目可以使用JUnit进行本地测试，未包含测试用例的题目请构造main方法进行手工测试

### Push试卷

在本地命令行/Git Bash中使用以下命令或使用IDE集成的Git插件将试卷push到服务器

```
git add .
git commit -m "[该次提交的Commit Message]"
git push
```

## 可能遇到的问题

## 可能遇到的问题

### 命令行中执行Clone和Push的时候明明密码正确但提示认证失败

方法一：当出现Git Credential Manager要求输入账号密码的对话框时关闭该对话框并在命令行中输入用户名以及在随后的弹框中输入密码

方法二：执行命令 `git config --system --unset credential.helper`

### 使用Eclipse内置的Git插件进行Clone时提示Git地址不正确

在试卷地址后增加 `.git` 后缀后作为Git地址

### 运行 `git commit` 命令时提示没有可以提交的内容

检查修改的文件是否已保存，检查导入到IDE中的项目的修改是否正确反映到原始文件中

### 提交后显示“编译或运行失败”并且控制台输出中出现 `\u0000` 字样

使用Notepad++或其他工具将源代码中编码为 `UTF-8 BOM` 的文件转换为 `UTF-8` 编码

## 一些FAQ

### FAQ

#### 考试中提交次数是否有限制？

考试中不对提交次数进行任何限制，但恶意刷提交数量占用系统资源者后果自负

#### 如何界定考试临近结束时的提交是否有效？

提交是否有效按照提交时间为准，只要考试结束前提交被系统记录则为有效，即结束前提交但排队到结束后运行测试的仍视为有效提交

#### 考试中服务端进行测试的环境？

服务端进行测试的环境为Linux & OpenJDK/JRE11 & Python3.10，资源限制为单核 & 1GB内存，单次运行限制1分钟时长

#### 能否使用第三方库和依赖？

除在试题中（Java为 `pom.xml` 中，Python为 `requirements.txt` 中）已经声明好的库和依赖外，不允许使用任何其他库和依赖

#### 试卷中的那些文件可以修改？

默认情况下只允许修改源代码文件（普通题目）或测试代码文件（测试覆盖率题目或变异覆盖题目）

#### 含有多道题的考试中一次提交会运行哪些题目？

系统会检测该次提交和上次提交之间的修改，并运行所有修改过的题目，如果没有检测到有对题目的修改（例如只修改了试卷根目录下的内容）则会运行所有题目

#### 一场考试、一次提交以及对一道题目的测试的分数是如何计算的？

一场考试的分数由分数最高的提交决定

一次提交的分数由所有题目的平均分决定，其中该次提交运行过的题目取该次提交的分数，该次提交没有运行过的题目取上次运行的分数（非历史最高分数），从未运行过的题目取0分

一道题目的测试分数为通过用例数在所有用例中的比例（普通题目）或测试覆盖率（测试覆盖率题目）或变异覆盖率（变异覆盖题目）

以上满分均为100分，所有计算采用向下取整策略

#### 考试结束后想要继续研究题目时如何获取试题？

考试结束后试卷地址仍有效，但只能进行clone而不能进行push