



元智能平台

版本 1.07

作者姓名: 杨国梁

2026 年 2 月

目录

第 1 章	引言	3
1.1	平台简介	3
1.2	联系	3
第 2 章	配置环境	4
2.1	检验是否安装	4
2.2	安装 jdk-11	4
2.2.1	下载	4
2.2.2	安装	5
2.2.3	配置环境变量	6
第 3 章	快速入门	9
3.1	运行平台	9
3.1.1	安装 yuan-common	9
3.1.2	运行 yuan-platform	10
3.1.3	打包 yuan-platform	11
3.2	运行 Maven 项目	11
3.3	运行 Python	12
3.4	小结	13

第1章 引言

元智能系统由北京力控元通科技有限公司山东分公司开源，并维护到 1.0.6 版本。系统分为平台和算法库两部分，二者既可以拆分使用，也可以灵活结合。Yuan 平台是元智能系统中的平台部分，简称为 Yuan。

Yuan 当前包括 yuan-common 和 yuan-platform 两部分。yuan-common 提供通用的接口和工具等内容，可以被扩展插件和 yuan-platform 使用。yuan-common 基于 Apache 2.0 开源。yuan-platform 是平台主体部分，主要基于 Apache 2.0 开源，其文本编辑 rsyntaxtextarea 插件，使用的是 BSD-3-Clause 协议。需要注意的是使用的停靠窗口 docking-frames-common 和 docking-frames-core 基于 LGPL-2.1 协议。

Yuan 支持 Java 和 Python 编辑和运行，当前只在 Windows 操作系统测试过。支持 Java 标准的 Maven 项目的编辑、编译、运行、打包、安装到仓库，添加自定义插件等基本功能。支持 Python 的编辑、运行等功能。当前 Yuan 还比较简单，不太适合于做复杂的开发，主要与 AI 算法库结合使用，形成零代码可扩展的 AI 智能平台。

1.1 平台简介

第 2 章介绍了 Yuan 的环境配置。

第 3 章介绍了怎么快速入门使用 Yuan，并使用 Yuan 安装了 yuan-common 到仓库，运行了 yuan-platform 源文件。

1.2 联系

系统技术支持，请联系杨国梁：Guoliang-Yang@hotmail.com

v1.06 以前版本业务合作，请联系尤从国(北京力控元通科技有限公司山东分公司)：youcongguo@126.com

平台的开源主页：<https://github.com/YuanIdea/yuan>

本文最新修改时间为 2026 年 2 月 11 日。

第2章 配置环境

Yuan 当前测试了 Window 操作系统的使用。Yuan 使用了 apache-maven-3.9.0 和 jdk-11，发布文件中内置了 apache-maven-3.9.0。通过 2.1 节检验安装情况，运行 Java 时，如果发布文件中没有提供内置的 jdk-11，则需要用户自己安装，并配置环境变量，具体可参见 2.2 节。

2.1 检验是否安装

Win+R 键输入 cmd，如图 2-1 所示。进入到 Windows 控制台，然后输入：`java -version`，出现：`'java'` 不是内部或外部命令，也不是可运行的程序或批处理文件，说明未安装。

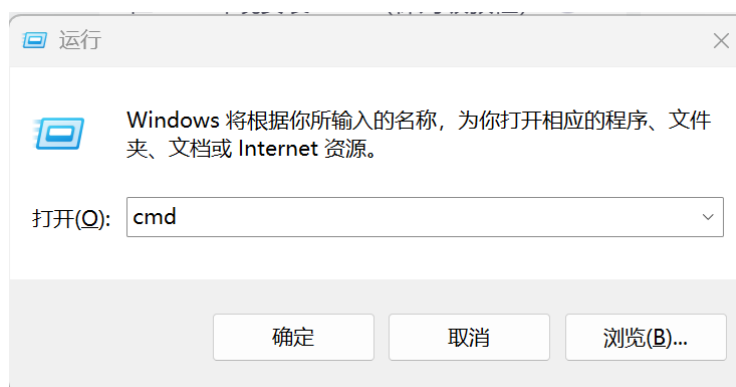


图 2-1 打开命令行

如果出现了类似图 2-10 的提示信息，说明安装成功，可以跳过 2.2 节，直接使用 Yuan。

2.2 安装 jdk-11

本节基于 jdk-11，演示下载安装和配置过程。如果已经安装可以跳过本节。

2.2.1 下载

官网地址[<https://www.oracle.com/java/technologies/downloads/>]，借鉴图 2-2 寻找 jdk-11 的版本，并下载安装。需要自己注册一个 Oracle 账号，下载的时候需要登录。本文选择的版本是 `jdk-11.0.25_windows-x64_bin.exe`。jdk-11.0.2 版本在编译 Yuan 时出现了英文代码注释乱码问题，建议使用 jdk-11.0.25。

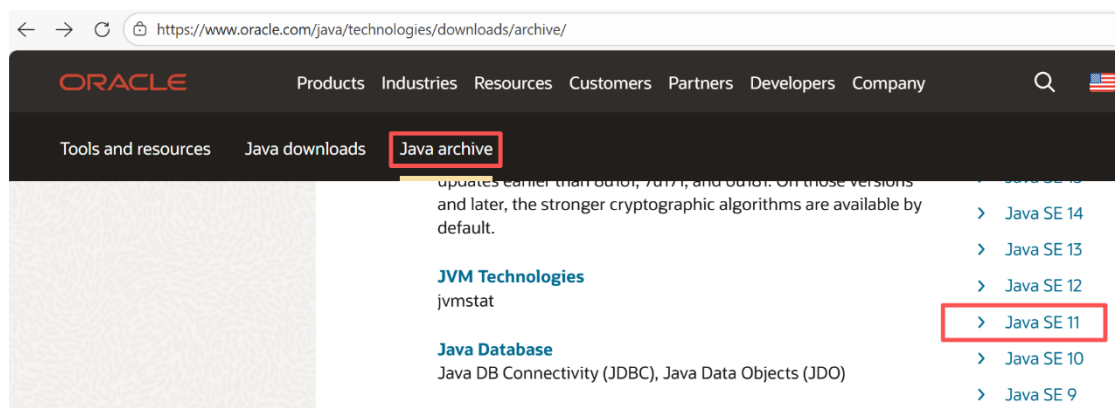


图 2-2 找到 jdk-11

2.2.2 安装

双击打开下载好的文件，出现图 2-3，点击下一步。



图 2-3 进入安装界面。

在图 2-4 中，选择合适的安装目录。



图 2-4 选择安装目录

2.2.3 配置环境变量

本文以 Win11 为例演示添加 jdk 环境变量。不同系统的环境变量设置方式，可通过网络查询具体方法。



图 2-5 电脑属性

右击“此电脑”选择属性，如图 2-5 所示，在弹出的如图 2-6 设置窗口中搜索“环境变量”，选择“编辑系统环境变量”。

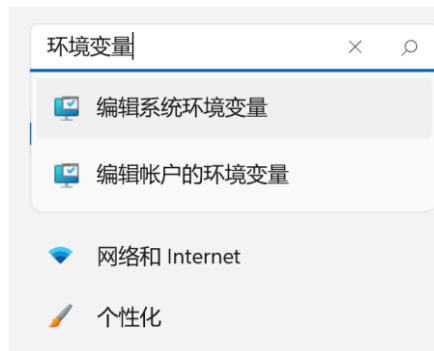


图 2-6 选择系统环境变量

在弹出的图 2-7 的“系统属性”中，点击环境变量按钮。



图 2-7 选择环境变量

2.2.3.1 添加 JAVA_HOME

在系统变量中点击新建，设置变量名为 JAVA_HOME，变量值填入刚刚安装的 JDK 的路径，如图 2-8。

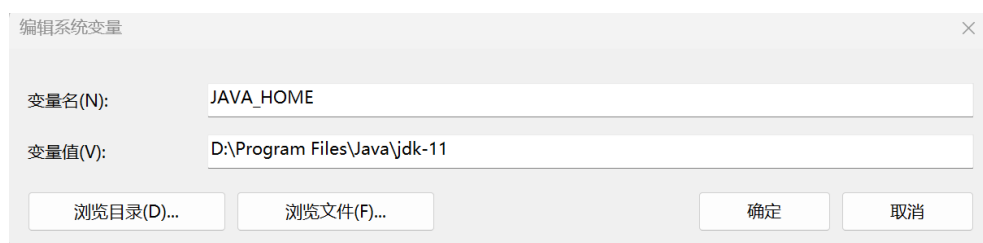


图 2-8 添加 JAVA_HOME

2.2.3.2 配置 Path

Path 变量中包含了系统可以执行的文件路径。当你在命令行中输入 `java` 或 `javac` 命令时，操作系统会根据 Path 环境变量的配置查找可执行文件。

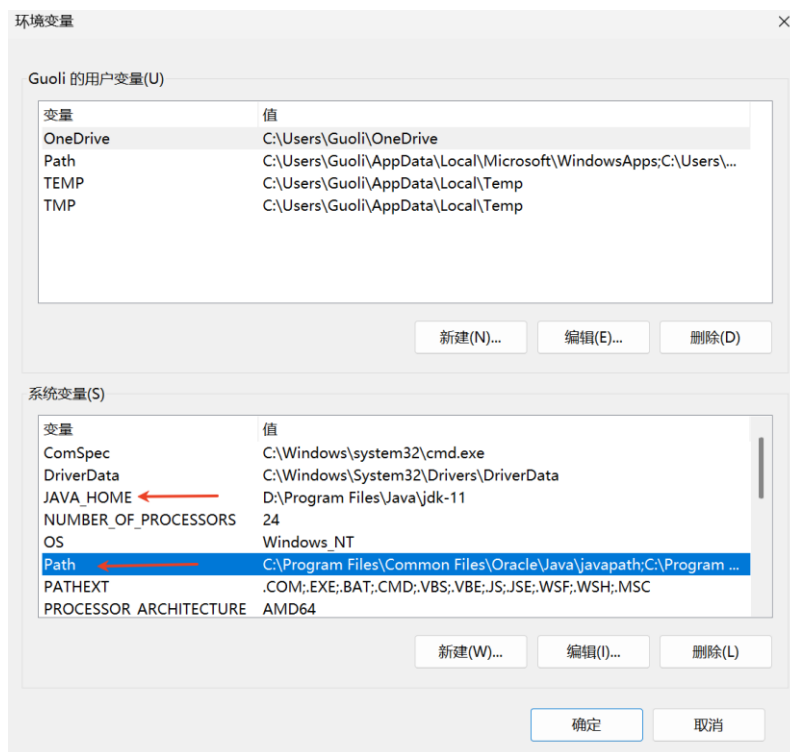


图 2-9 环境变量

在图 2-9“环境变量”对话框中，找到并选中“系统变量”区域中的 Path 变量，然后点击“编辑”，添加“`%JAVA_HOME%\bin`”到 Path 中。

```
C:\Users\Guoli>java -version
java version "11.0.25" 2024-10-15 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.25+9-LTS-256)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.25+9-LTS-256, mixed mode)
```

图 2-10 安装 jdk-11 成功

按照 2.1 节检验安装，如果出现了类似图 2-10 的提示信息，说明安装成功，可以跳过 2.2 节，直接使用 `yuan`。

第3章 快速入门

按第 2 章配置了平台运行环境后，在[\[https://github.com/YuanIdea/yuan\]](https://github.com/YuanIdea/yuan)下载 Yuan 源码，找到 `yuan\yuan-platform\target\artifacts\yuan\yuan.exe`，双击后出现图 3-1 平台界面，不同的版本会略有差异。

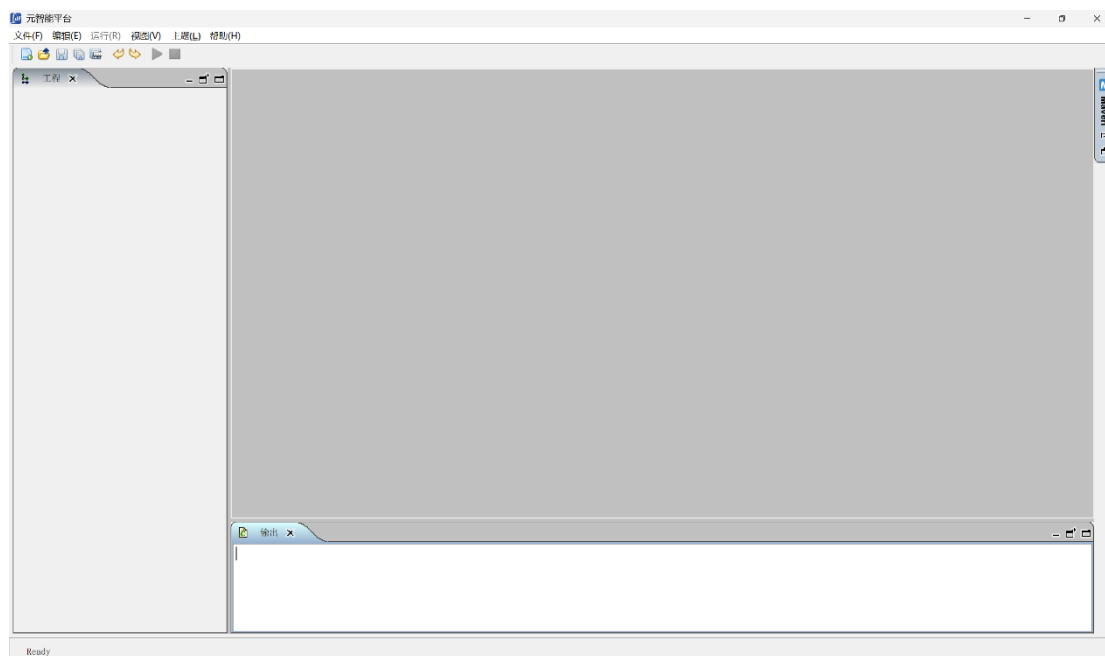


图 3-1 平台界面

3.1 运行平台

本节介绍在 Yuan 平台内编译，安装依赖和打包 Yuan 平台。

3.1.1 安装 yuan-common

选择平台中的“文件”→“打开目录”菜单栏。选择 `yuan-common` 的根目录。在右边的 maven 停靠窗口的“生存期”中，双击“`compile`”命令，运行效果如图 3-2 所示。在底部的输出停靠窗口中，会出现“`✓ compile succeeded!`”等编译信息。

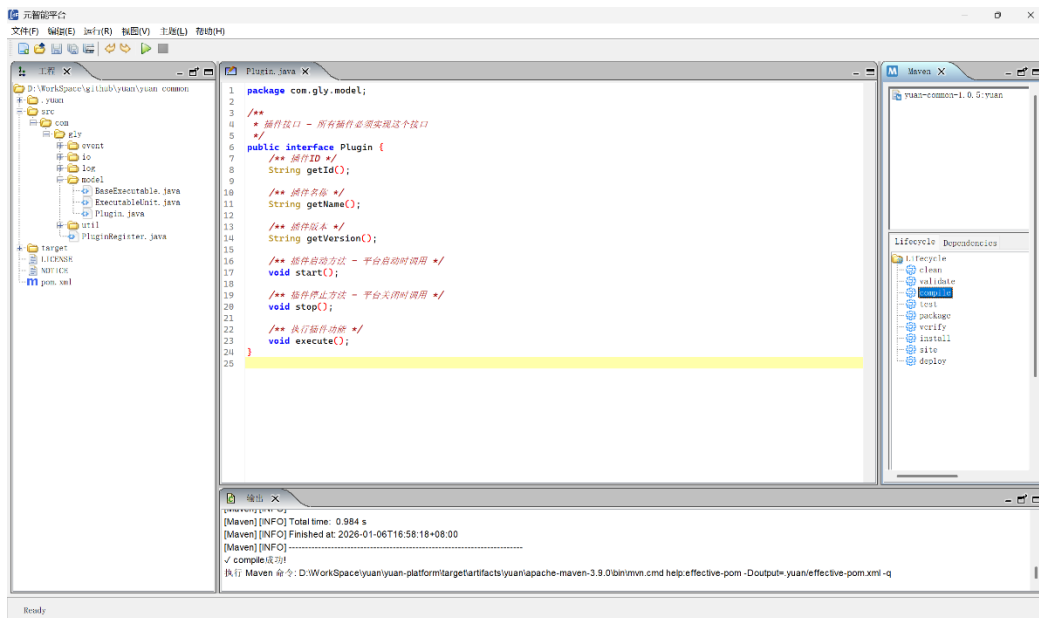


图 3-2 编译 yuan-common

点击“install”命令，将 yuan-common 安装到仓库，成功后输出面板如图 3-3 所示。

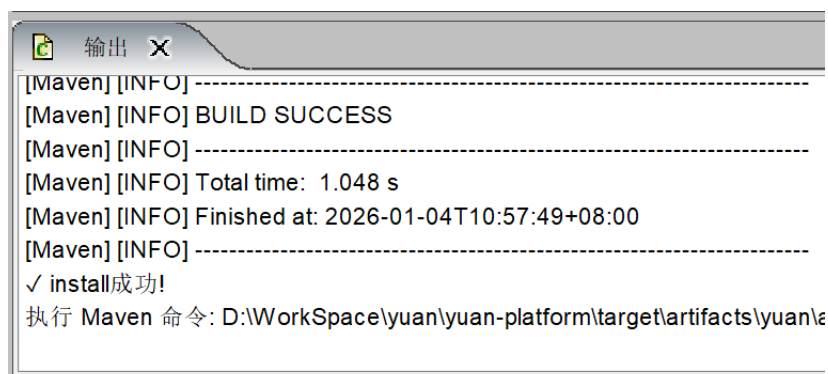


图 3-3 安装到仓库

3.1.2 运行 yuan-platform

安装完 yuan-platform 后，就可以运行 yuan-platform 了。同样，可以使用工具栏的“打开目录”按钮，打开 yuan-platform 的根目录。点击工具栏的绿色三角“运行”按钮，可以看到编译和运行 yuan-platform 源码后的效果为图 3-4。

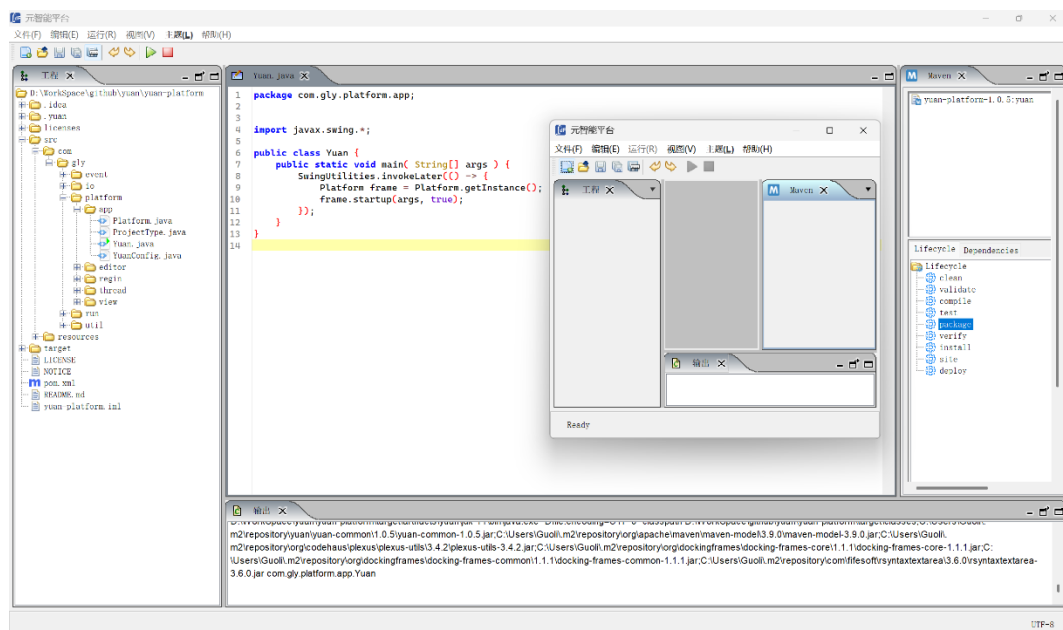


图 3-4 运行 yuan-platform

3.1.3 打包 yuan-platform

右边的 maven 停靠窗口的 Lifecycle 中，双击“package”命令后，将 yuan-platform 打包到了 yuan-platform/target/artifacts/yuan/yuan-platform-1.0.7.jar 中，在命令行中切换到 yuan-platform/target/artifacts/yuan 目录后使用 java -jar yuan-platform-1.0.7.jar 命令，同样可以打开图 3-1 所示界面。

3.2 运行 Maven 项目

Yuan 平台支持对标准类型 Maven 项目，以 yuan 平台使用的第 3 方 docking-frames 为例，在当前要运行的“工程”资源管理中添加配置文件.yuan/project.xml，见配置 3-1。

配置 3-1 project.xml

1. <?xml version='1.0'?>
2. <project>
3. <type>maven</type>
4. <jdk>D:\Program Files\Java\jdk-1.8</jdk>
5. </project>

其中，type 指定了是 maven 类型的项目，jdk 指定了当前工程中使用的 jdk 存储路径，docking-frames 选择 jdk-1.8 版本进行编译，如果没有设置 jdk 标签，默认使用的是 JAVA_HOME 中指定的 JDK，见 2.2.3.1 节。

在右边的 Maven 停靠面板中选择 docking-frames-demo-chess，将需要的依赖添加到仓库后，点击运行按钮出现图 3-5。

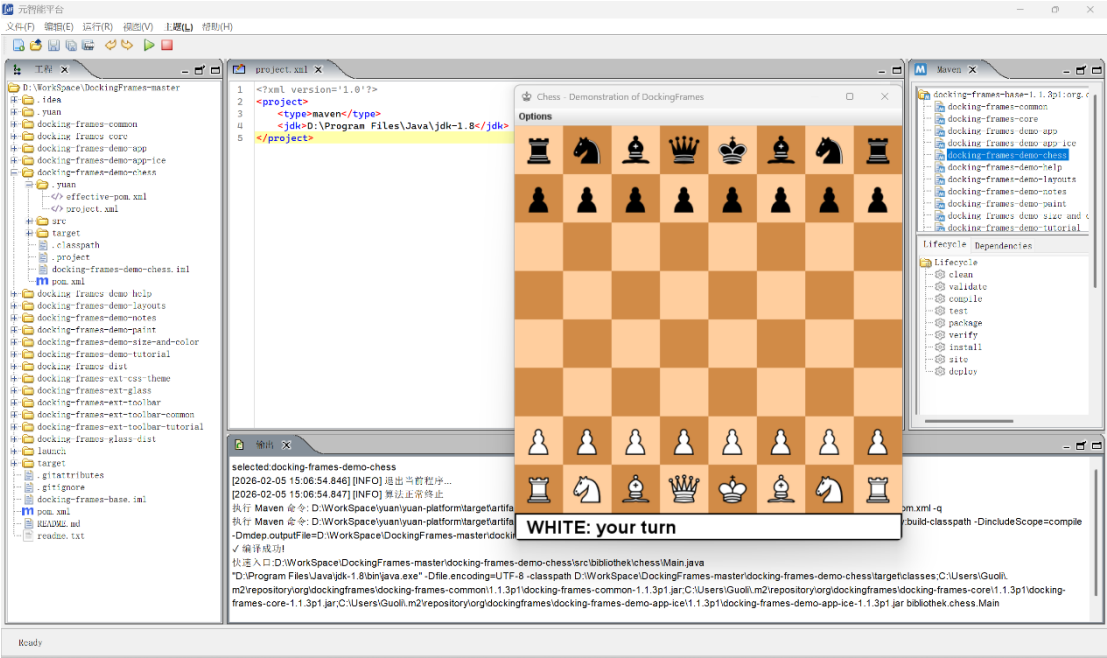


图 3-5 运行第 3 方开源 Java 工程。

注意，在运行工程前，需要将所有的本地依赖，使用 Maven 中的 install 暗装到仓库。通常，在使用 yuan 平台打开包含 pom.xml 的根目录时，平台会默认创建一个.yuan/project.xml 的配置文件，用户可以根据需要修改配置内容。

3.3 运行 Python

使用 Yuan 平台运行 Python 代码前，需要自行安装 python 环境，Yuan 只提供了编辑和运行 Python 的功能。安装 Python 成功后，在命令行中输入 python 命令，如果类似现出图 3-6 所示 Python 版本信息，则安装成功。

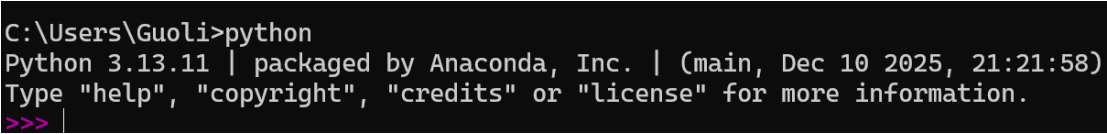


图 3-6 python 已安装效果

在 yuan 平台打开项目后，添加配置文件.yuan/project.xml，见配置 3-2。

配置 3-2 project.xml

1. `<?xml version='1.0'?>`
2. `<project>`

3. `<type>python</type>`
4. `<pythonHome>D:/ProgramData/anaconda3/envs/python310</pythonHome>`
5. `</project>`

其中，type 指定了是 Python 类型的项目，pythonHome 指定了当前工程中使用的 python.exe 存储路径。

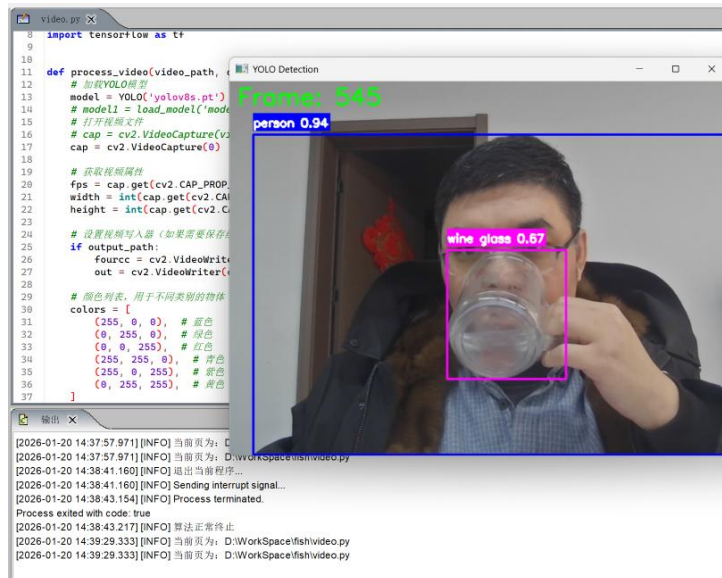


图 3-7 图像识别

图 3-7 演示了就是 Yuan 平台中，通过 Python 调用 yolov 模型，进行图像识别的过程。

3.4 小结

本节介绍了 Yuan 的使用，并在 Yuan 内编译安装了 yuan-common 到仓库。并在 Yuan 中编译运行了 yuan-platform，演示了 Yuan 的具体使用。当前 Yuan 只支持标准的 Maven 项目和简单的 Python 运行。