# 自动化测试 Automation Test

### 1.项目概述 Project Description

The channel first introduced the Smithsonian Medical R & D Center's implementation plan and related materials for the C9 LVP automated test project.

该频道旨在介绍Smiths Medical 研发中心对于C9 LVP自动化测试项目的实施方案与相关资料。

### 2.环境配置 Environment Configuration

推荐使用pycharm社区版（<https://www.jetbrains.com/pycharm/download/#section=windows>）

因为可免费使用，具体安装过程参考（<https://www.runoob.com/w3cnote/pycharm-windows-install.html>）。

推荐python解释器使用3.6及以上版本（<https://www.python.org/downloads/>）

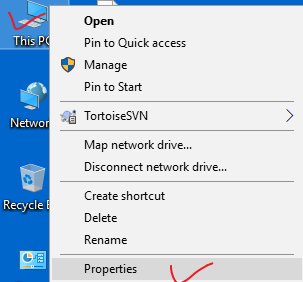
并配置环境变量。（建议安装时选择解释器路径为默认系统盘目录）

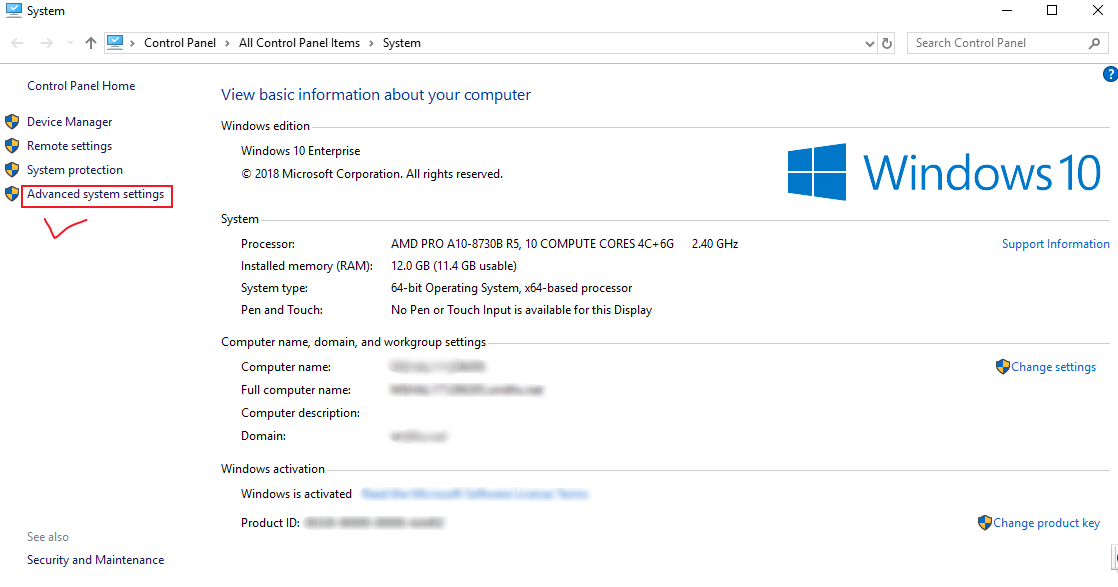
注意：下载python解释器之前需要根据个人计算机操作系统的类型（位数）来选择相对应的版本。

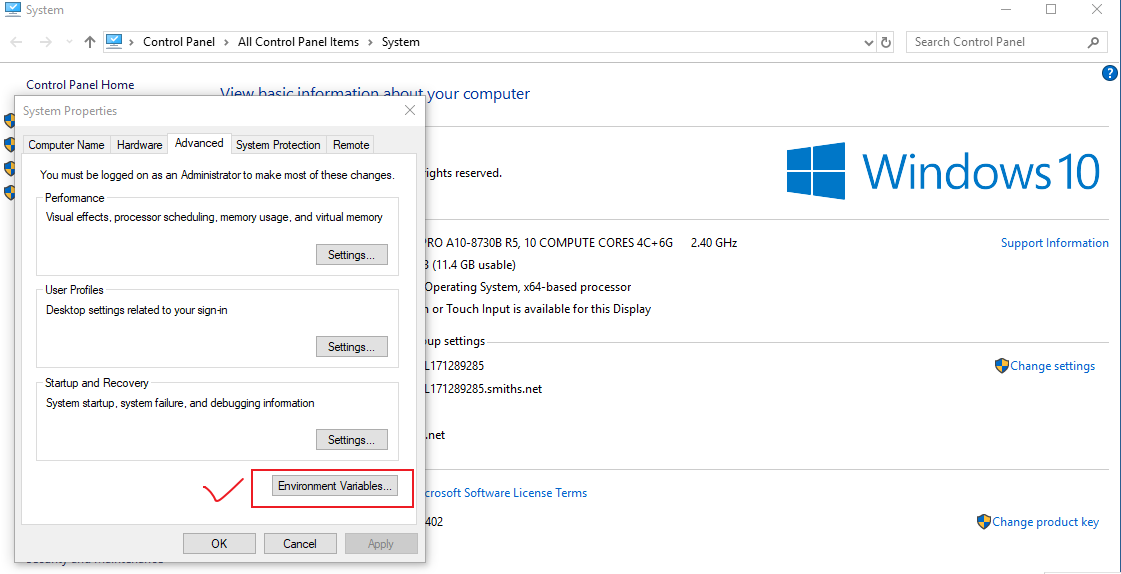
在此介绍两种查看计算机操作系统类型（位数）的方式：

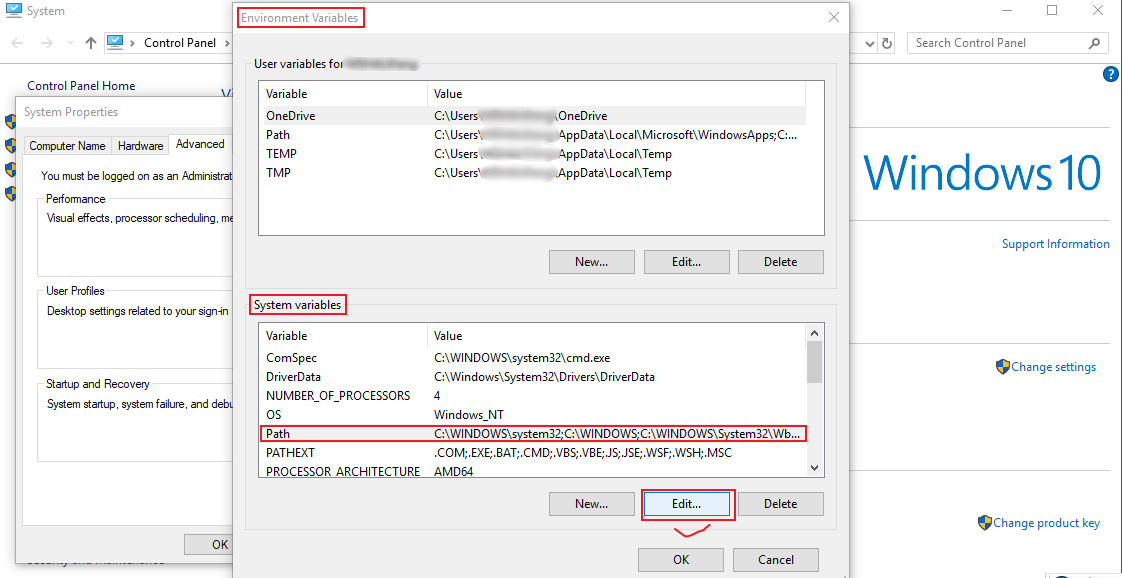
**2.1普通方式：**  
2.1.1打开电脑，进入操作系统，在电脑桌面找到“我的电脑”或者叫“计算机”，右键我的电脑，找到属性。  
2.1.2在弹出来的电脑属性界面中，找到电脑操作系统类型，此处就可以看到电脑位数了。  
**2.2命令行方式：**2.2.1打开电脑，进入电脑系统，在电脑开始菜单中的查询框中输入cmd。找到命令行运行程序。或者也可以通过电脑键盘组合键，快捷键组合为：windows + R，即可直接打开命令行运行程序了。  
2.2.2确定或者敲键盘回车，进入电脑命令行，在电脑命令行中，输入systeminfo命令，等待查询电脑配置信息。  
2.2.3在查询出来的电脑配置信息中，找到电脑系统类型，这就是当前电脑的位数了。

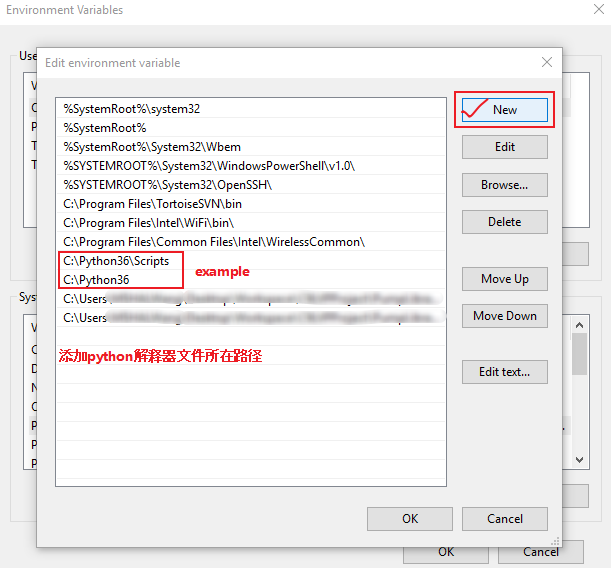
**配置环境变量如图所示：**





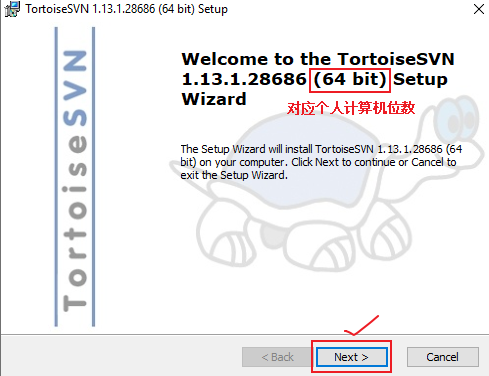


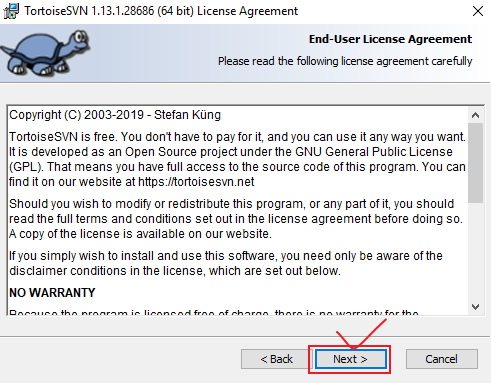


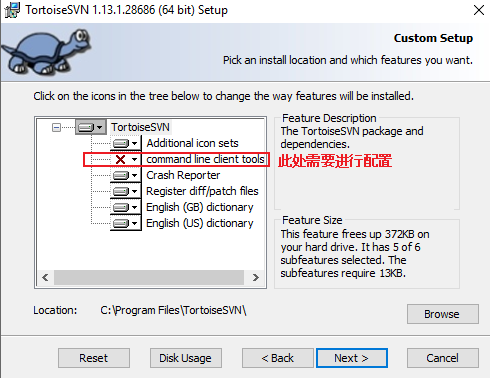


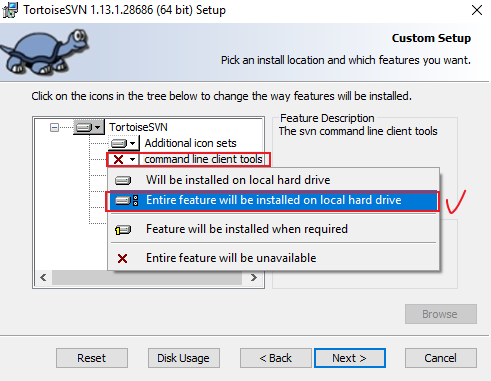
从SVN服务器下载项目文件到本地目录（建议将项目导出到系统盘目录的空文件夹），地址为：

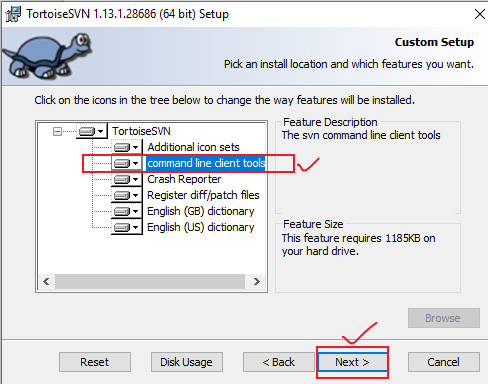
（<https://mshasap04.smiths.net:8443/!/#TestAutomation/view/head/trunk/RobotFramework%20Library/-05-%20LVP%20Python%20Project>），选择下载（<https://tortoisesvn.net/downloads.html>）与操作系统位数（x32/x64）相对应的SVN版本，具体安装SVN如图所示：

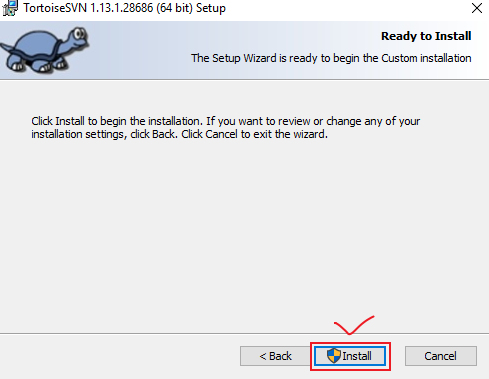




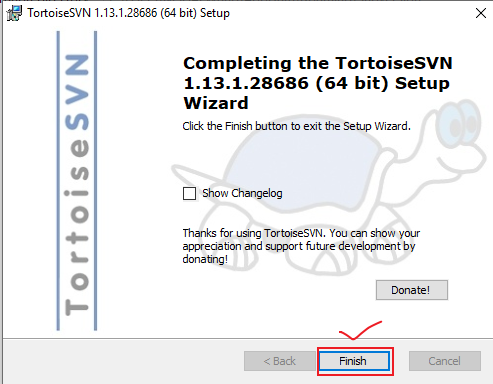




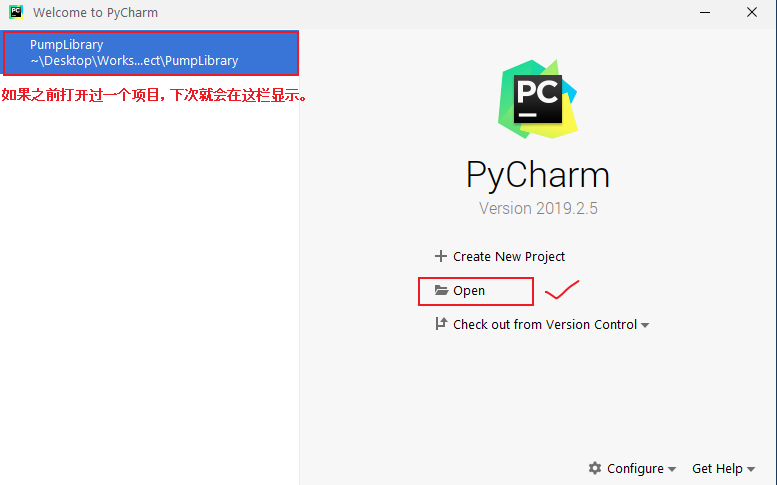


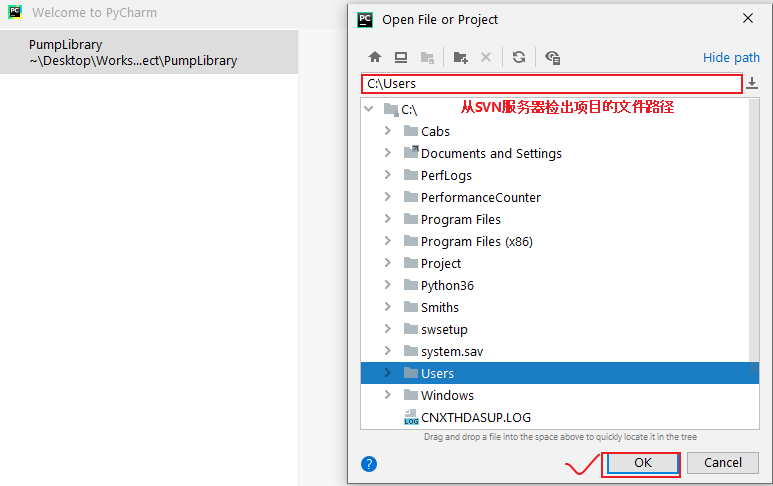


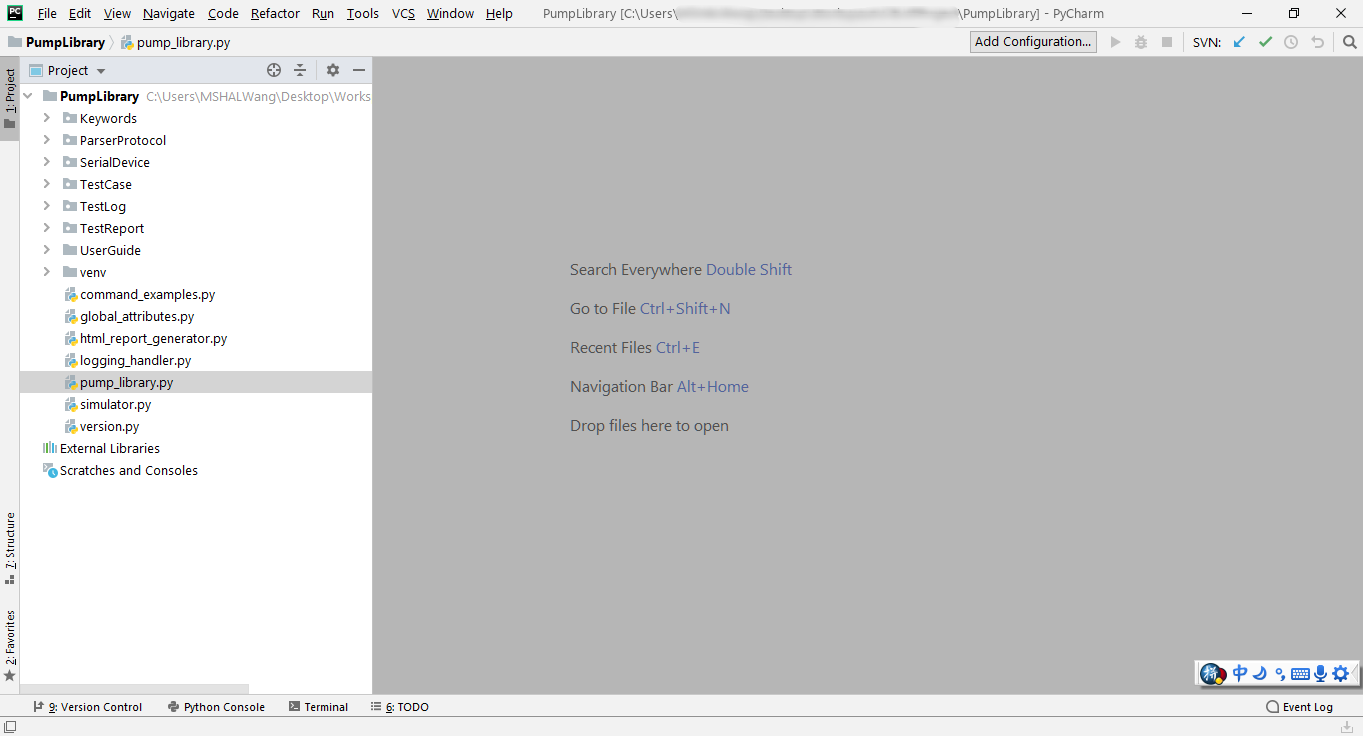
安装过程中会弹出若干窗口，不做选择直接点击OK即可。



随后使用pycharm打开项目文件（PumpLibrary），如图：



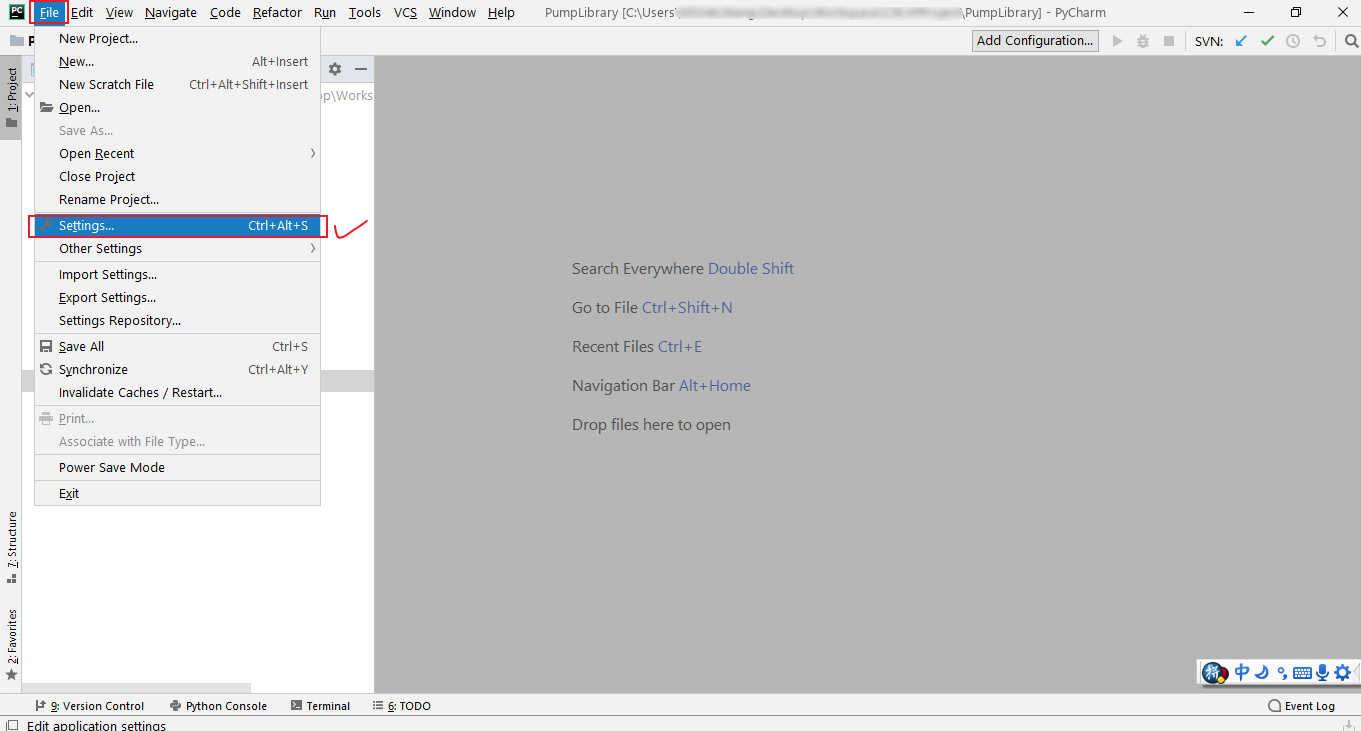


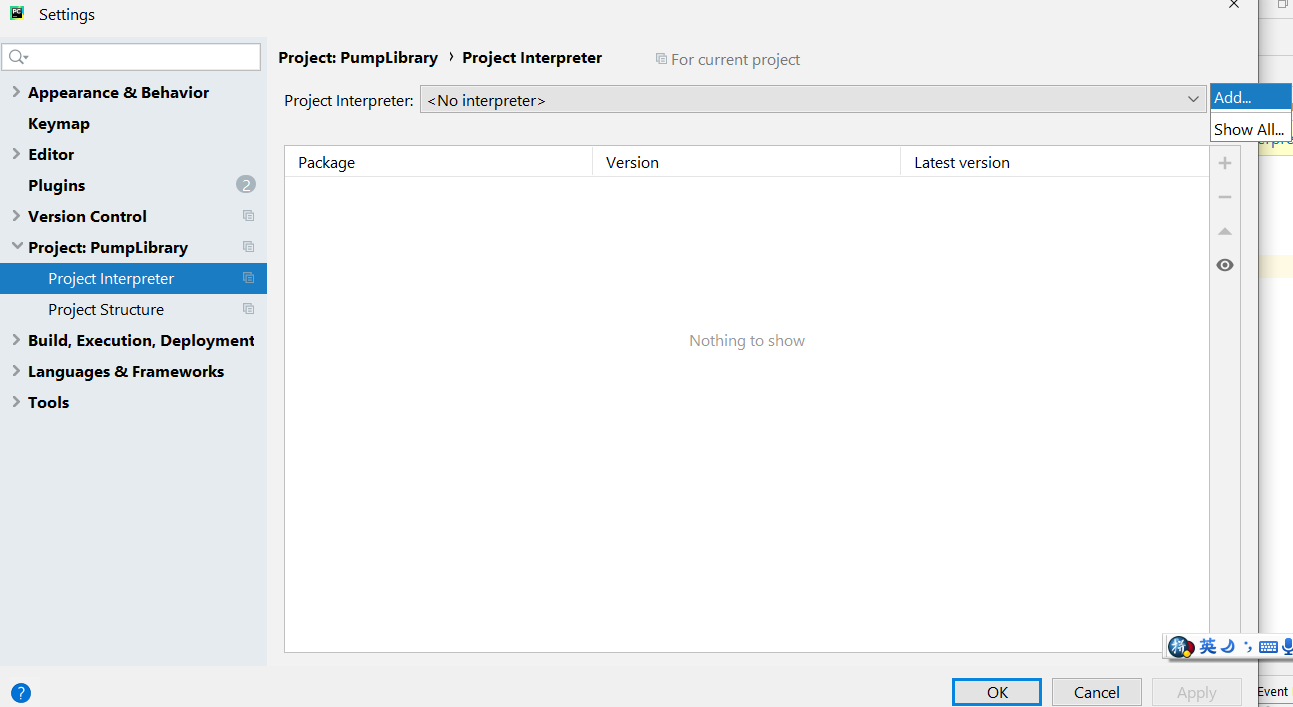


为了方便管理不同项目的依赖包和第三方库，搭建环境时尽量不要选择系统解释器而使用虚拟环境，虚拟环境是用于依赖项目管理和项目隔离的Python工具，允许Python站点包（第三方库）安装在本地特定项目的隔离目录中，而不是全局安装（即作为系统范围内的Python的一部分）。

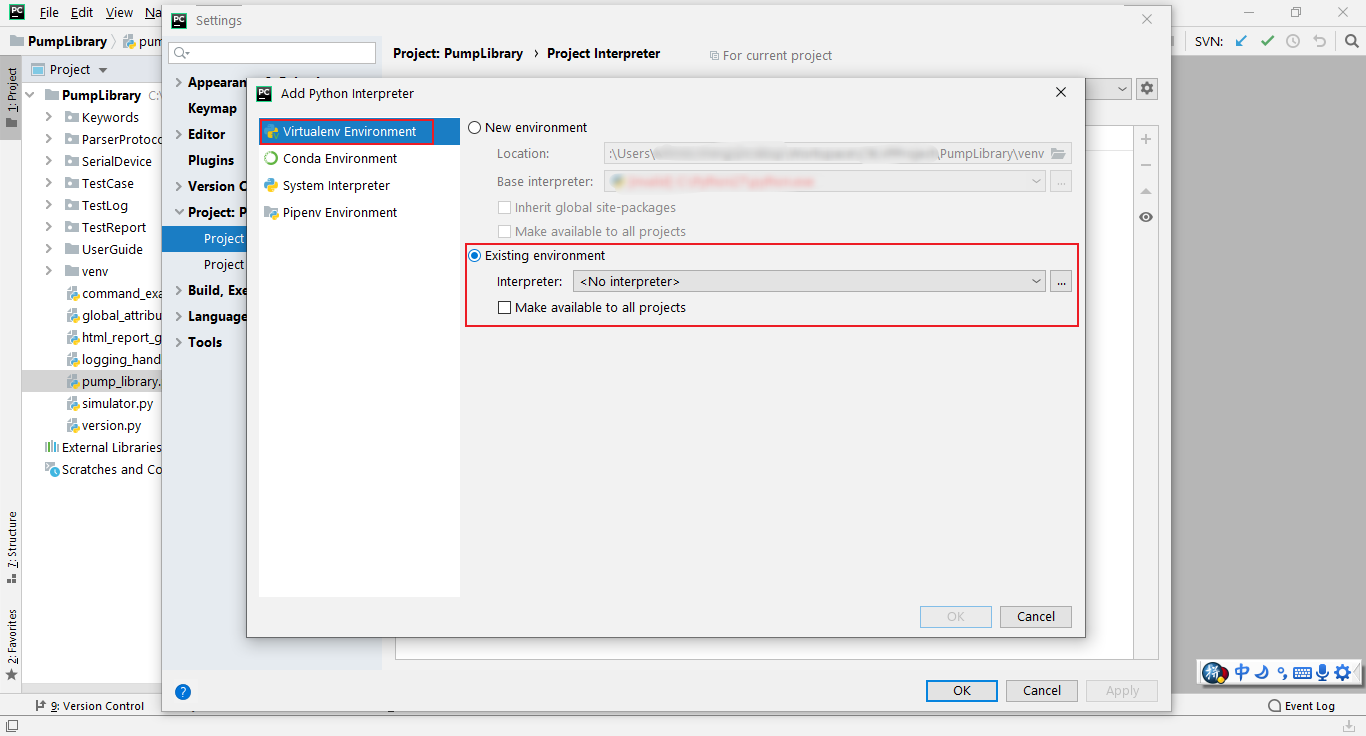
Pycharm中配置python解释器虚拟环境的具体步骤：

1.pycharm中点击设置，然后选择Project Interpreter，点击Add…如图：

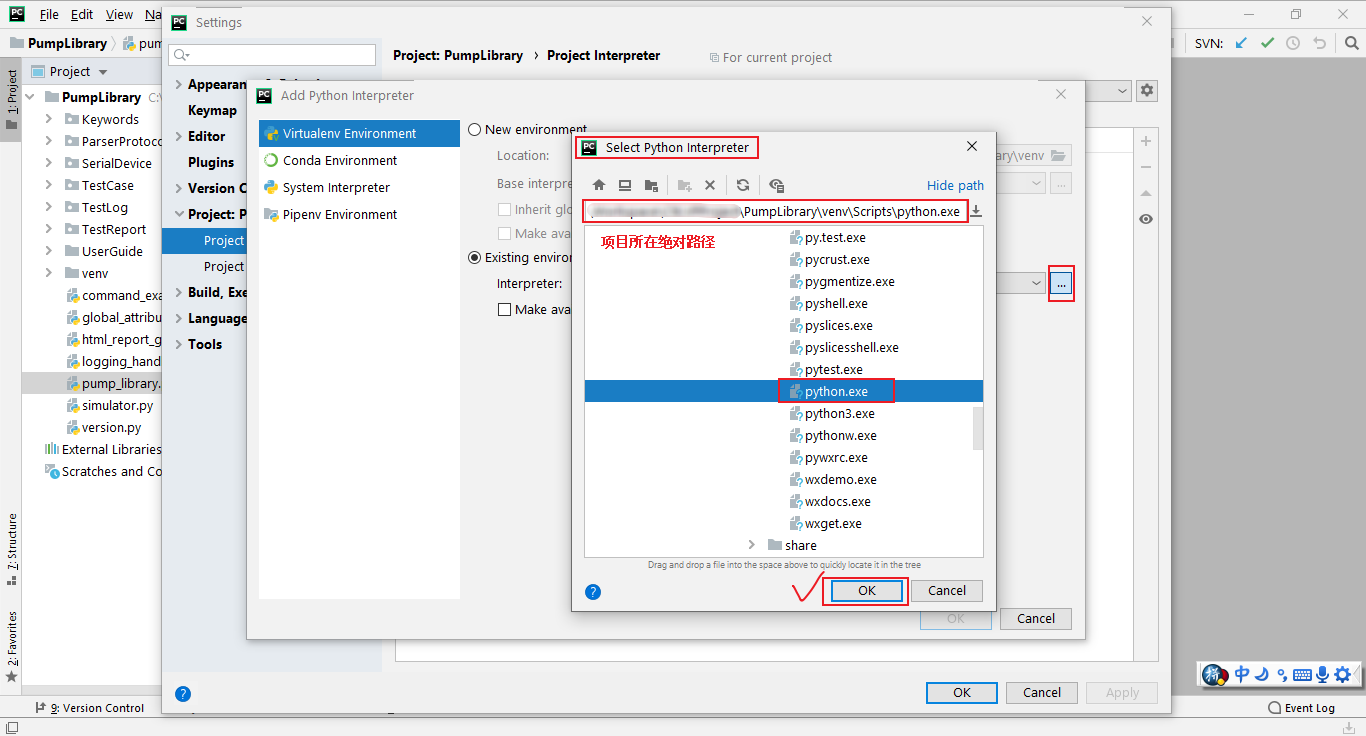


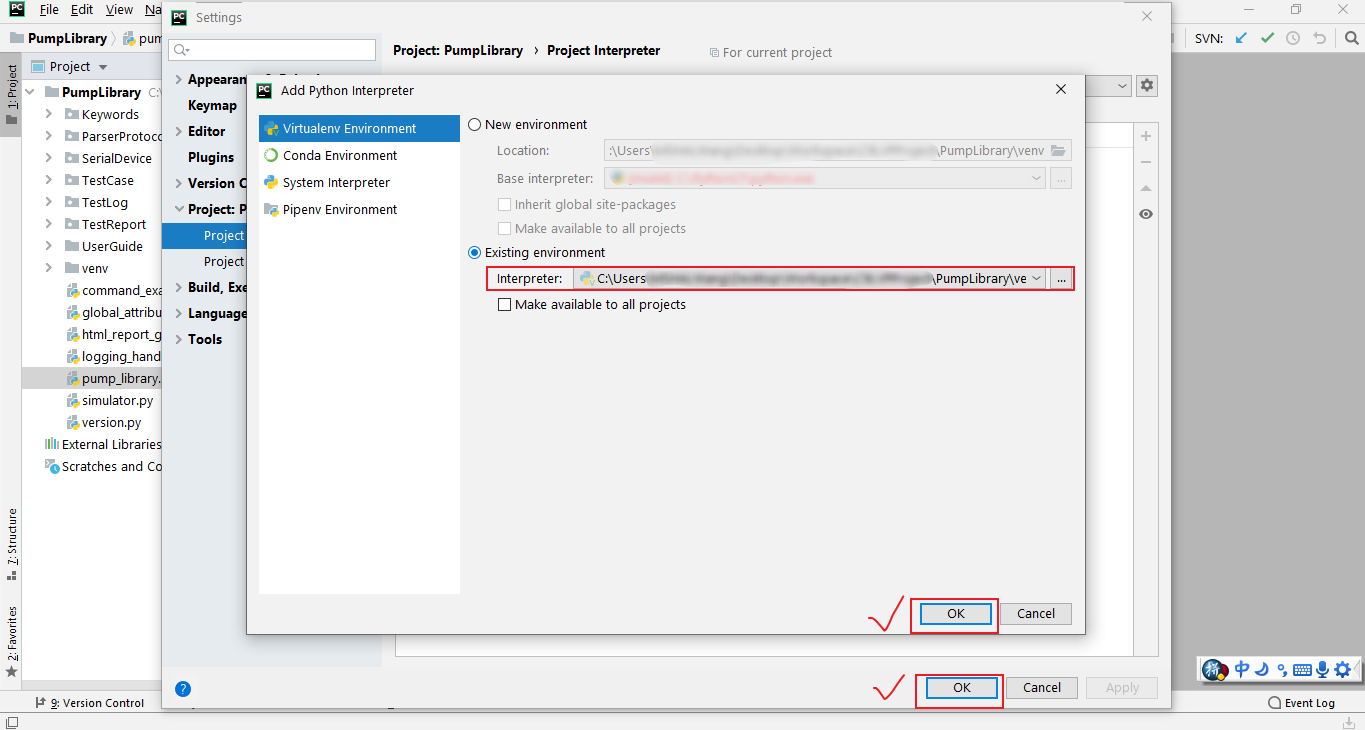


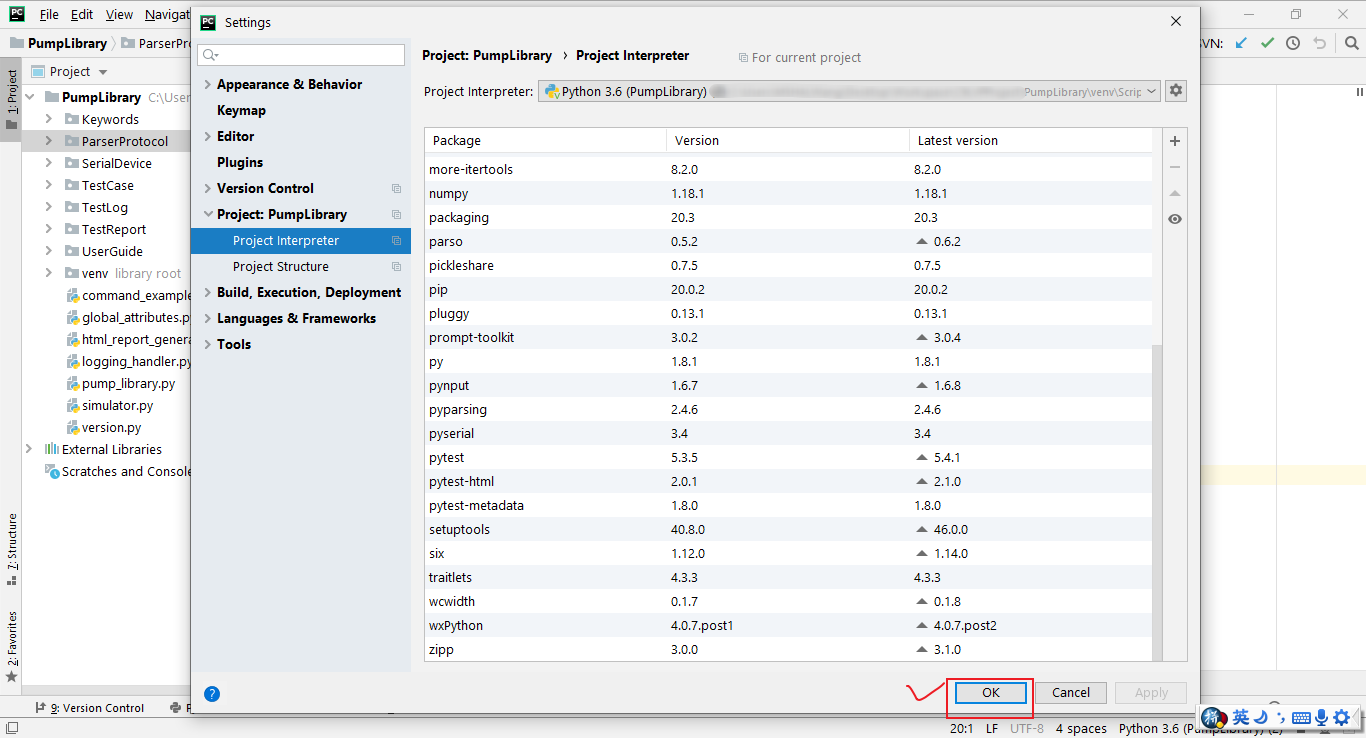
2.选择Virtualenv Environment选项后选定Existing environment，点击 …按钮，如图：



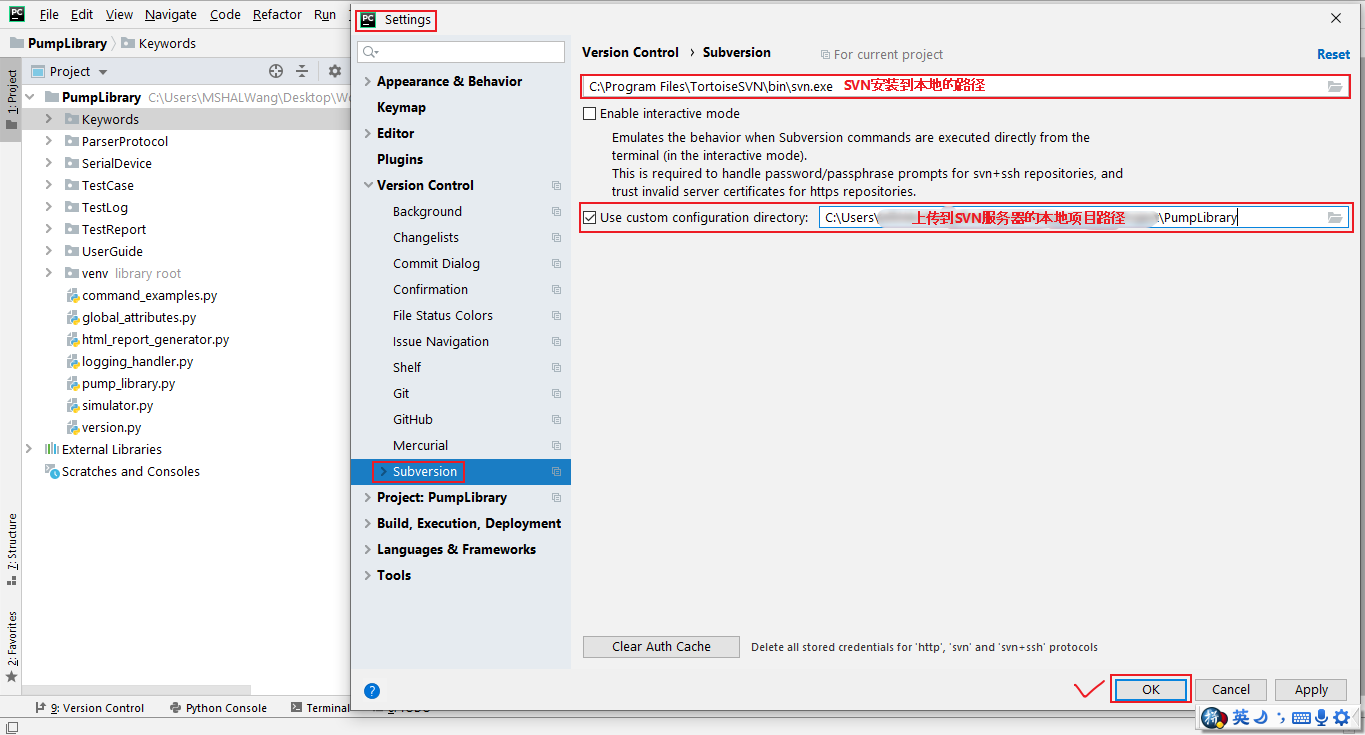
3. 按照图中提示步骤逐步操作，如图：



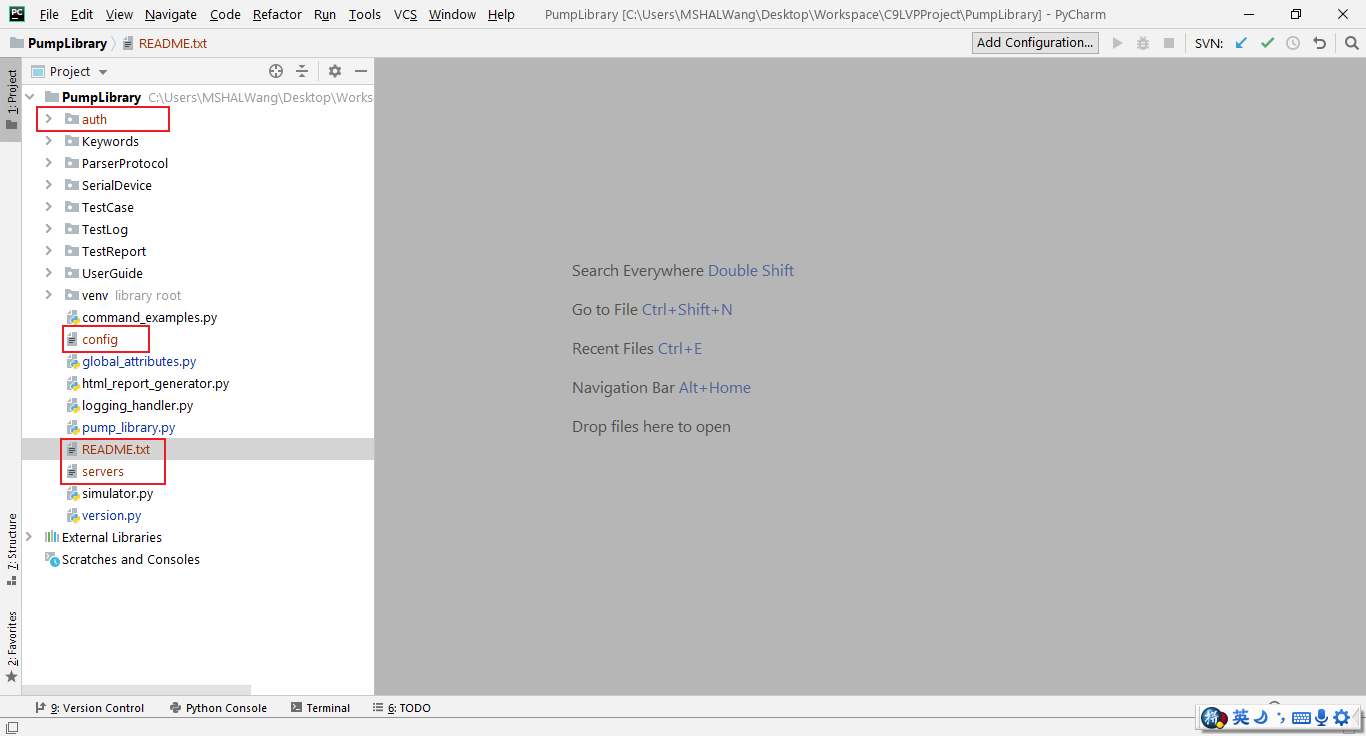


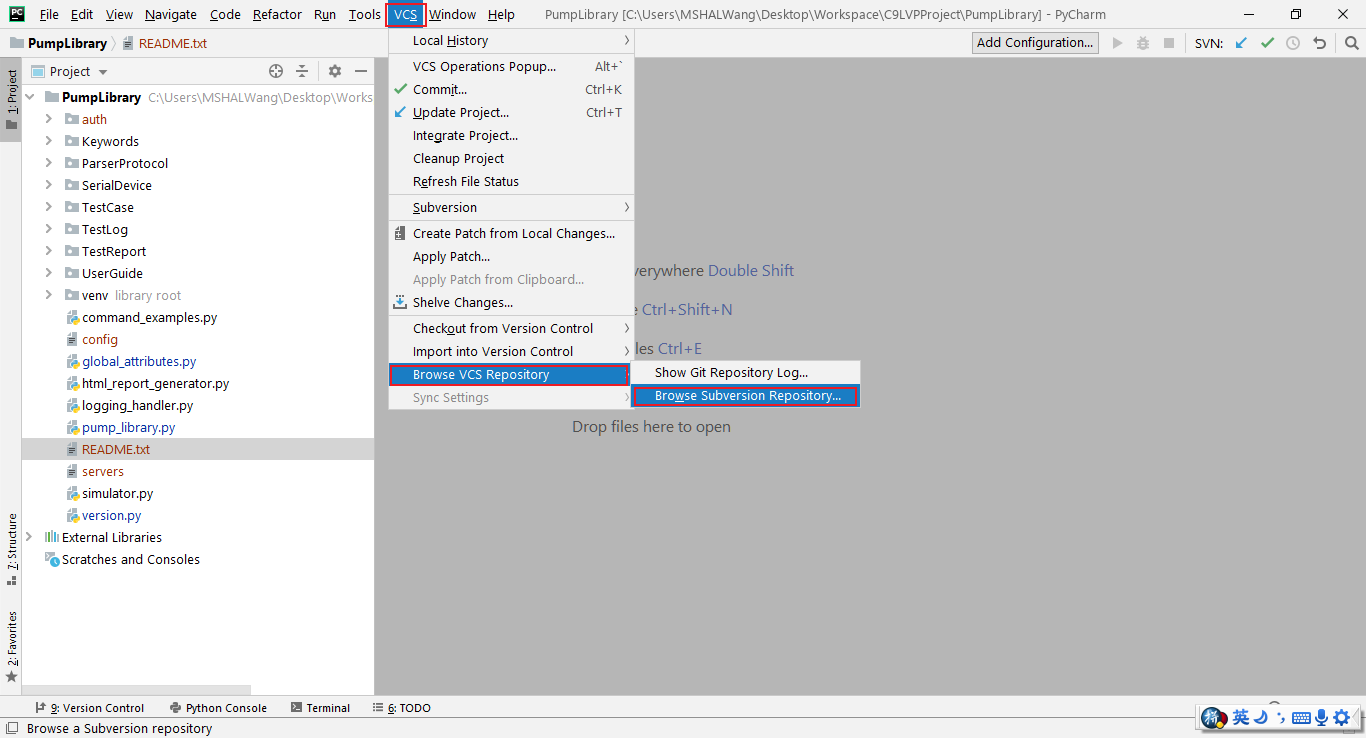


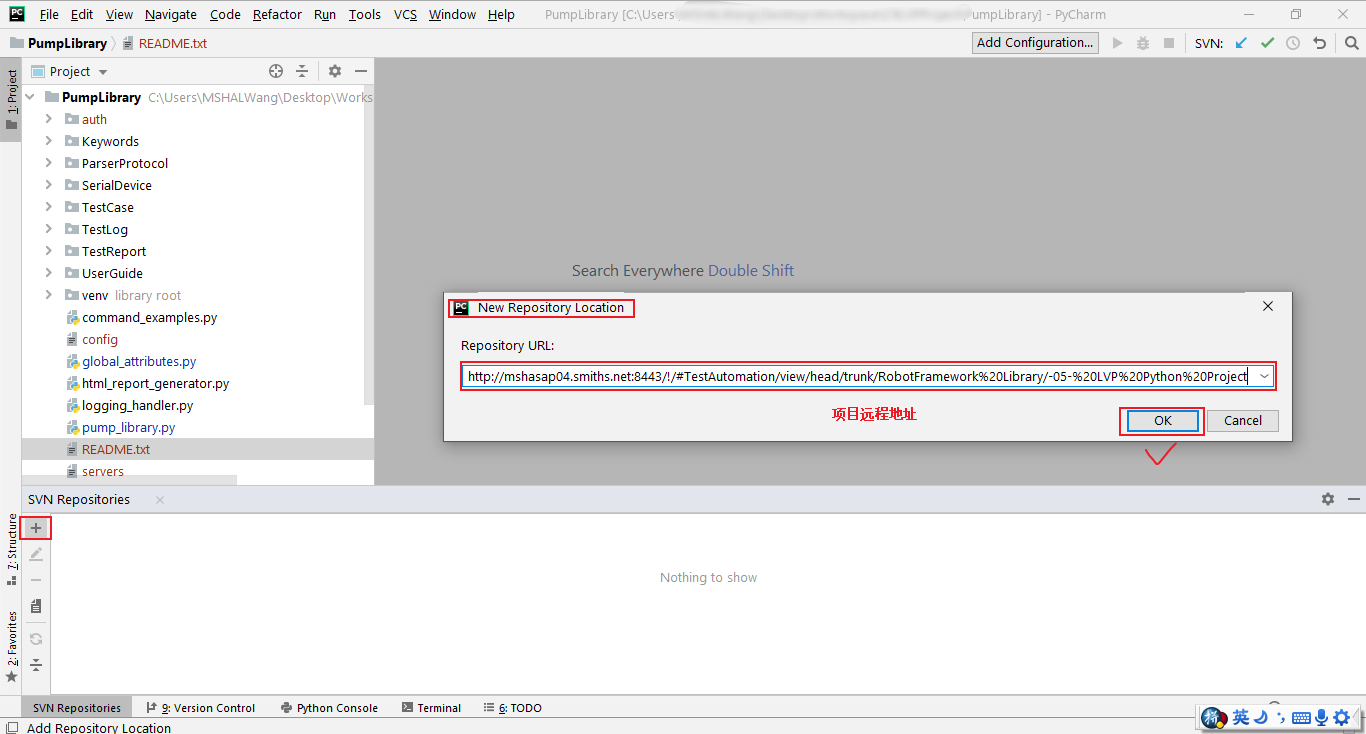
在pycharm中对SVN进行配置以便后期对项目进行更高效快速的版本管理：

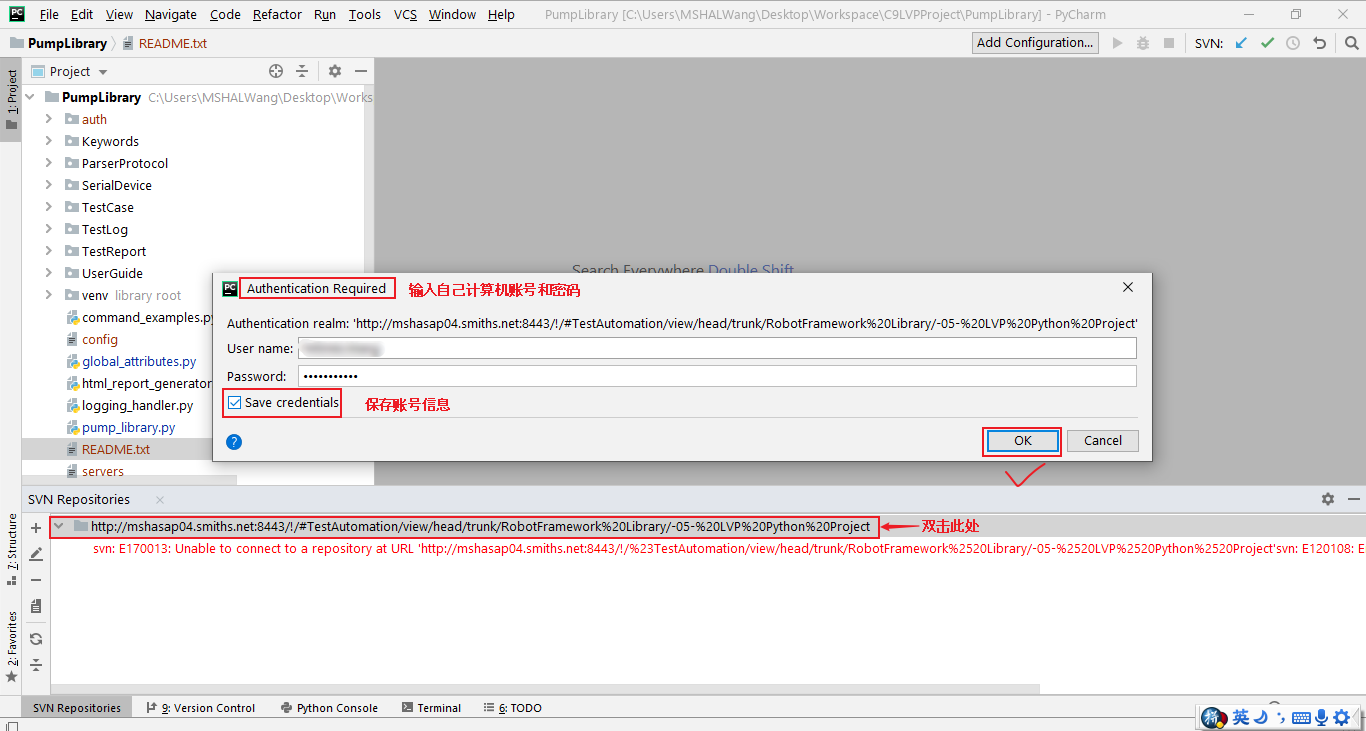


点击确定后可以观察到项目目录文件中增加了几个SVN配置文件。









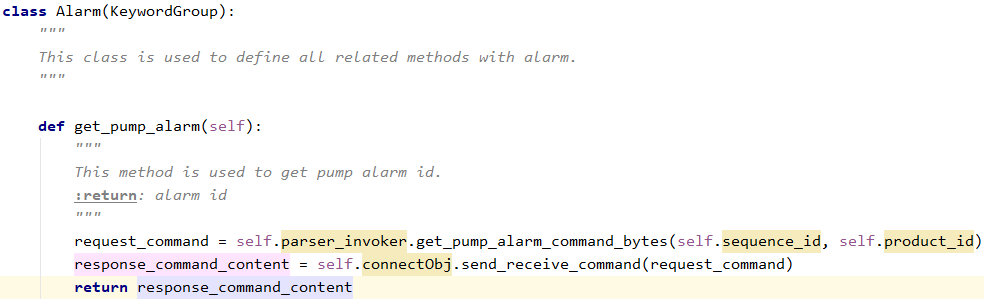
### 3.框架分层 Frame Layering

PumpLibrary项目主要用于处理测试用例在泵端的执行，包括命令的发送与接收的处理。囊括组件层与连接层：

* **组件层：**针对泵的各个功能而制定的关键字命令。
* **连接层：**负责与泵进行串口通讯连接，同时调用GPA的命令来向泵进行发送与接收。

**3.1组件层：**

* 组件层在库中的Keywords文件夹内。它是所有测试用例调用的关键字来源。根据产品的功能模块可以分为五部分：
  + Alarm
  + DeliveryView
  + HardwareControl
  + MenuSettings
  + SensorControl
* 定义的伪代码为：
  + **关键字名（即函数名）**：  
    **请求命令** = 使用解析对象从动态链接库中获取对应的十六进制命令（该命令长度会根据传入的参数不同而有所改变）  
    **响应内容** = 串口连接对象（调用）发送后接收命令并作出处理  
        返回泵端响应的内容
* 实际代码如图：



**3.2连接层：**

* 连接层在库中的位置主要由两部分组成：
  + ParserProtocol：主要用于与ProtocolParser\_x64.dll的交互，解析协议并获取命令，dll文件中是GPA与泵端通讯的一些API接口。
  + SerialDevice：主要用于计算机程序代码与泵端的串口连接工作，对串口进行发送和接收等操作的处理。

### 4.文件结构 File Structure

* Keywords：针对每个模块的创建的关键字，在robot脚本中可以被识别和调用
* ParserProtocol: 对dll文件进行协议解析的模块
* SerialDevice：对RS232串口进行处理的模块，使用pyserial，对串口进行识别和连接。
* TestCase：用于存放测试用例，如临床场景和smoke test等相关用例。
* UserGuide：包括本自动化项目具体操作文件等
* venv：虚拟环境，其中包含了本项目所需的第三方库文件。
* \_\_init\_\_.py：在初始化时创建关键的类对象如：self.parser\_invoker, self.serialobject等
* global\_attributes：定义了全局变量。
* logging\_handler：日志处理器，用于保存测试用例中的日志信息。
* version：用于定义当前的库版本。
* write\_to\_csv：用于将请求命令和响应命令写入csv文件中保存。

### 5.命名规则 Naming Rules

为使项目管理规范化，使用方便化，理解清晰化，特为C9LVP自动化测试项目设计此命名规则，同时它也适用于所有Python项目中：

* **5.1大驼峰命名法：**
* 项目文件夹名（如：PumpLibrary）
* 包名(如：Keywords, ProtocolParser, SerialDevice)
* 类名(如：ProtocolParser包里protocol\_invoker.py中的class ProtocolParserInvoker(BaseInvoker):)
* **5.2小驼峰命名法：**
* 函数名（如：global\_attributes.py中的def setCommandFormat(conn):）
* **5.3 posix命名法：**
* py文件名（如：global\_attributes.py, version.py, write\_to\_csv.py）
* 类方法名（如：ProtocolParser包里protocol\_invoker.py中的class ProtocolParserInvoker类里的def get\_command\_id(self, input\_buffer, length):）
* 类属性名（如：ProtocolParser包里base\_invoker.py中的class BaseInvoker类里的self.path = dll\_path）
* 变量名（如：global\_attributes.py中的command\_format = conn）
* 实例对象名（如：\_\_init\_\_.py中的lvp = PumpLibrary()）
* **5.4. 自定义命名法：**
* 常量名（如：version.py中的VERSION = '0.0.3'）

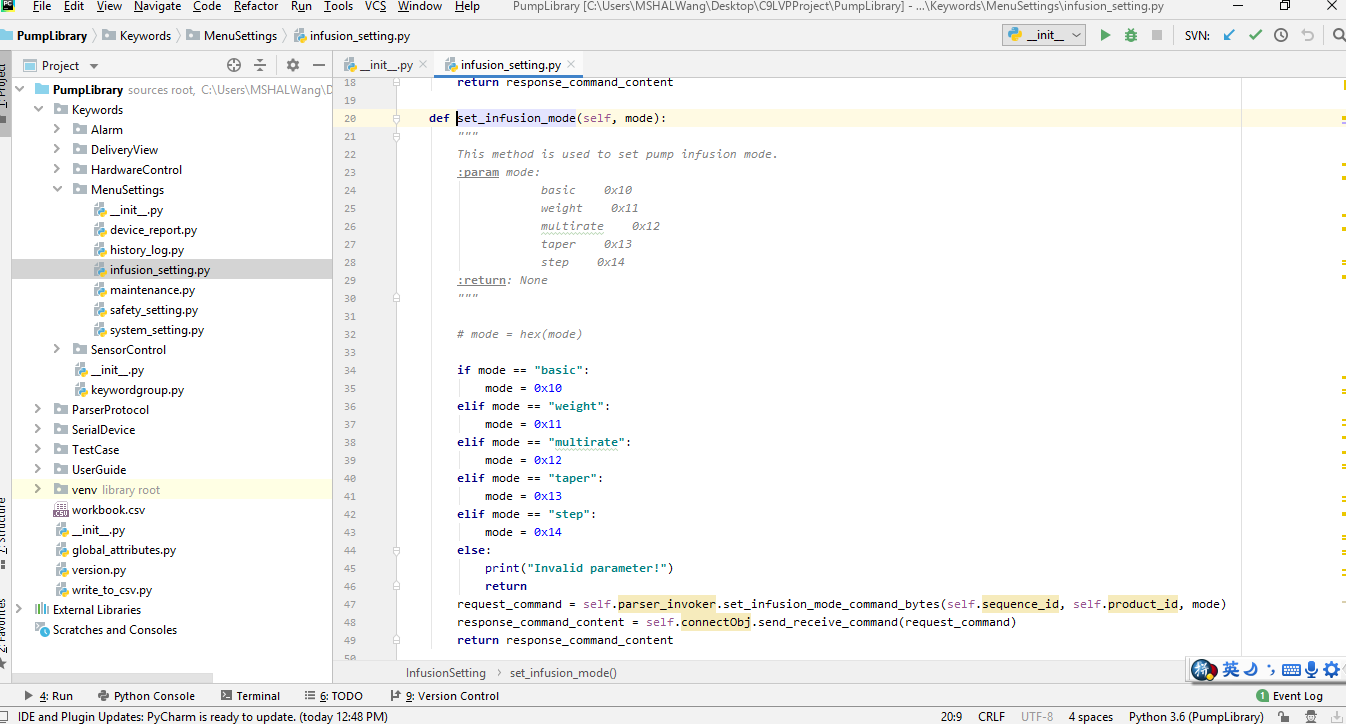
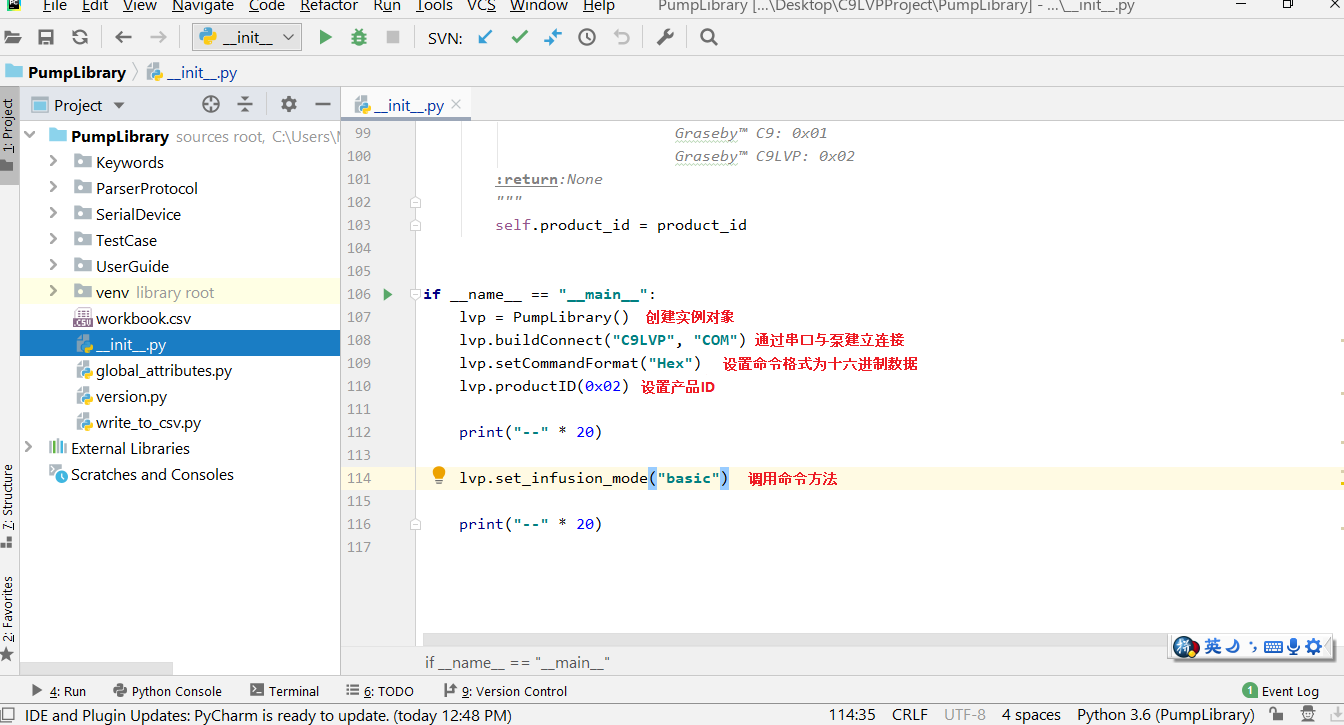
### 6.代码部分 Code Part

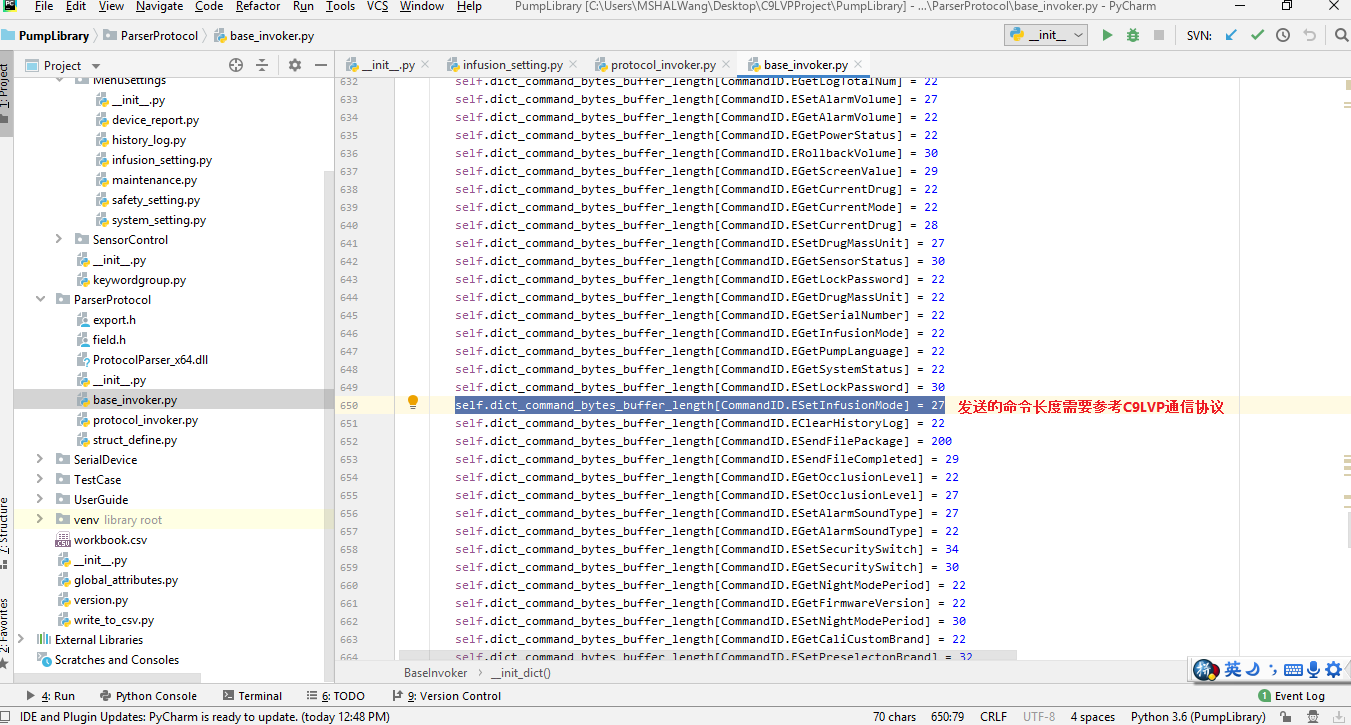
**添加关键字命令方法的具体步骤：**

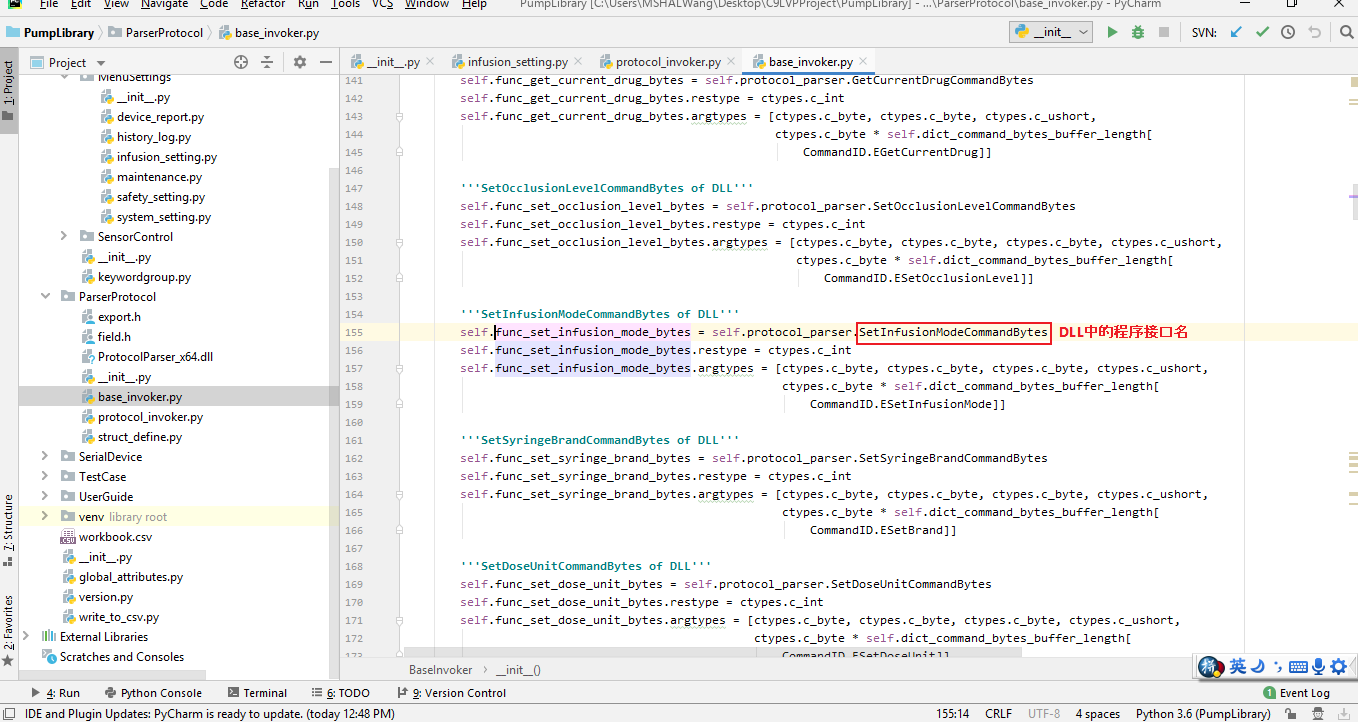
如果新添加的命令方法按照功能分类不属于已经创建成功的模块中，

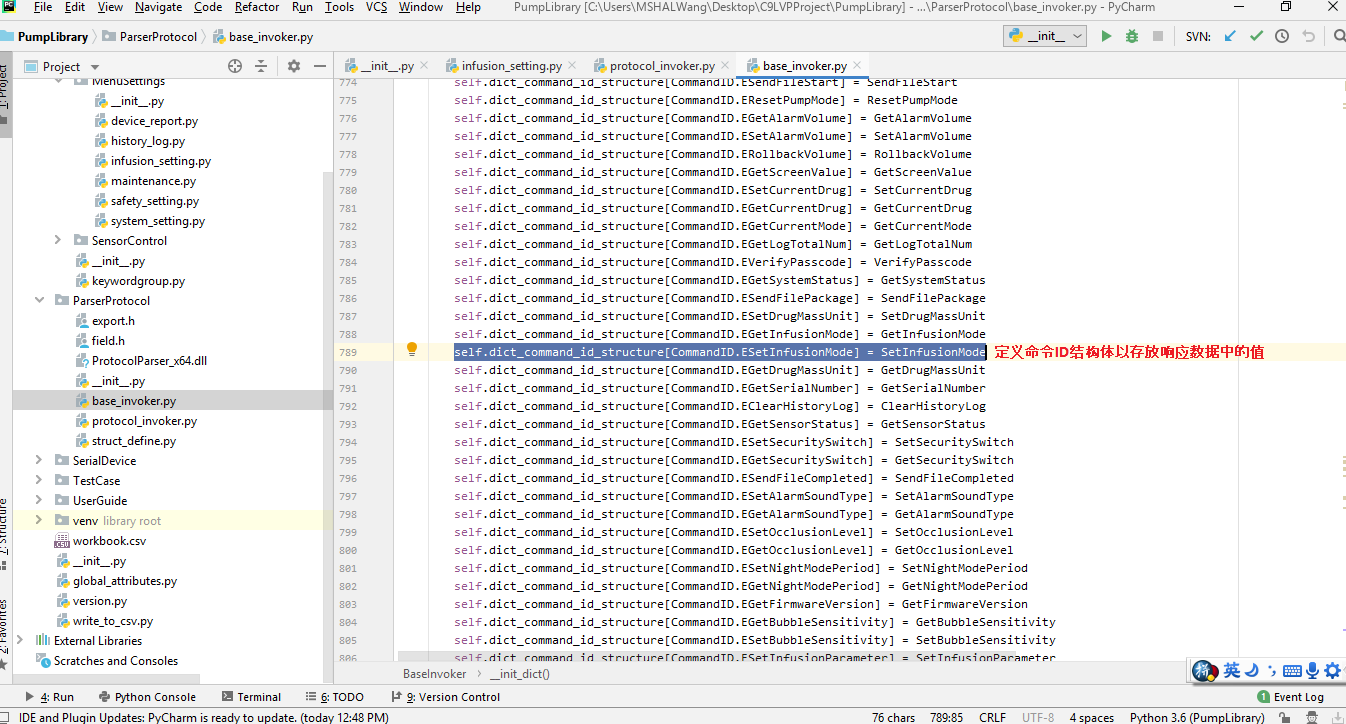
（如：Info, Infusion, InfusionParameter, DeviceReport, HistoryLog, InfusionSetting, Maintenance, SafetySetting, SystemSetting, Alarm, Bolus, Priming, Motor, Sensor）

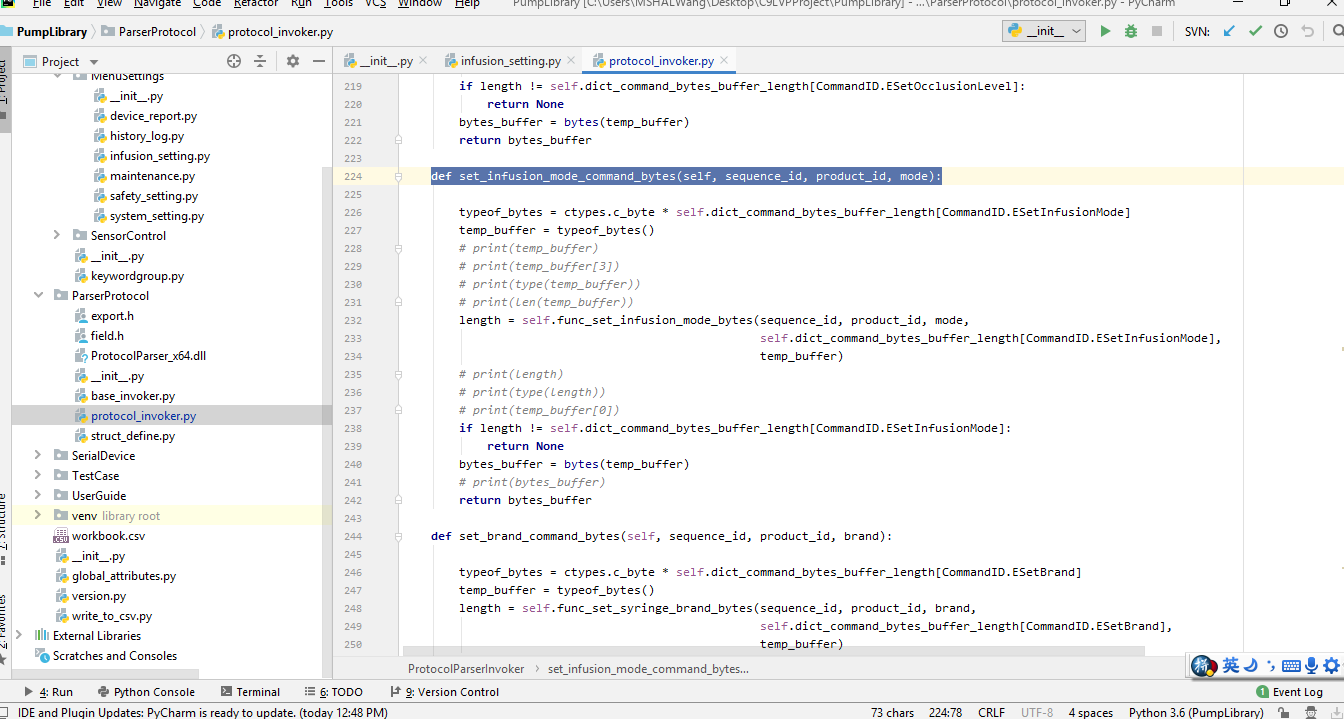
则需在Keywords中创建新的模块，并且在PumpLibrary类中继承所创建父类以使用新的命令方法。

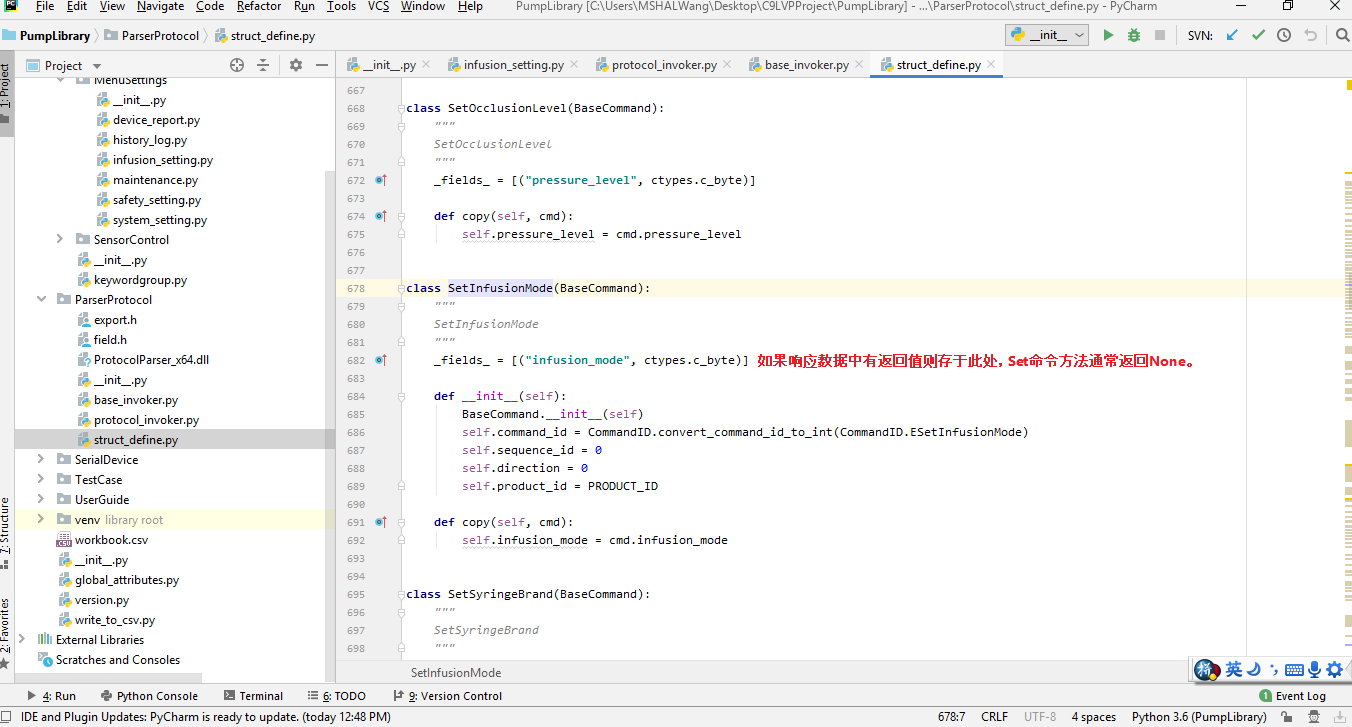


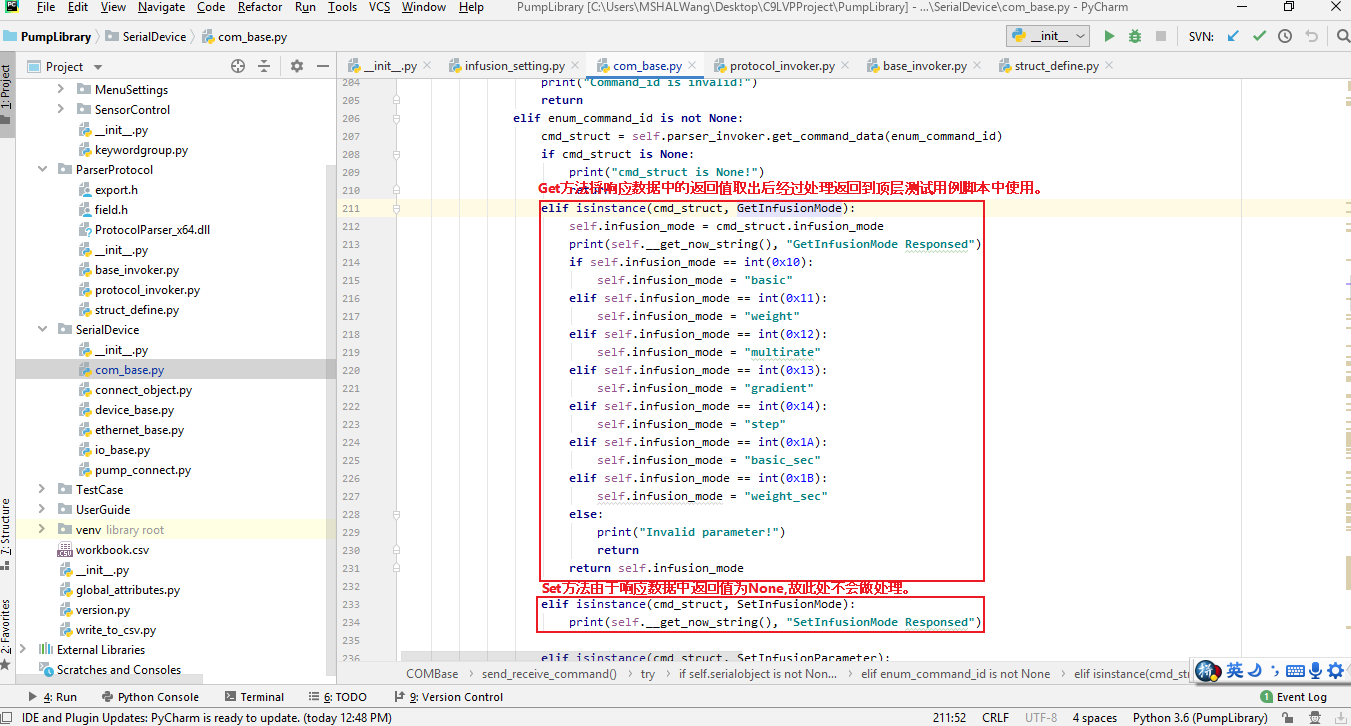












### 7.测试用例 Test Case