**中国矿业大学**

**计算机科学与技术学院**

**2019级本科生实验报告**

课程名称 软件工程实践

实验题目 实验三 编码

开课学期 2021学年第一学期

报告时间 2021年12月13日

学生姓名 胡钧耀

班 级 2019-4班

专 业 计算机科学与技术

任课教师 刘迎春

# 实验目的与要求

## 实验目的

学习 PHP 、 Java、 Python 集成编程环境，熟悉各语言的编码规范。

## 基本要求

1. 学习 PHP 语言基础，学习 PHP 集成编程环境 PhpStorm，了解常用插件，掌握断点调试功能；
2. 学习 Java 语言基础，学习 Java 集成编程环境 IDEA，了解常用插件，掌握断点调试功能，参考《Java编码规范及实践》，熟悉 Java 的编码规范，深入学习集成开发环境的使用与调试。；
3. 学习 Python 语言基础，学习 Python 集成编程环境 PyCharm，了解常用插件，掌握断点调试功能，使用Opencv库对二维码进行id识别；

# 华为云服务扩展点

服务扩展点是 DevCloud 平台的一种扩展插件，为 DevCloud 平台提供连接第三方服务的能力。

用户典型使用场景：在项目的流水线配置中，如果用户需要远程连接第三方服务，如：连接第三方 GitHub、码云的 Git 仓库获取项目源码，连接第三方 Jenkins 服务执行 Jenkins 任务，连接 Kubernetes 集群进行部署等，均可以使用服务扩展点实现。

本次实验中使用连接第三方码云仓库进行代码检查。

## 添加码云服务扩展点步骤

连接到 Github 帐号后可以获取该帐号下的仓库、分支等信息。

进入项目，在顶部导航栏单击【设置 > 通用设置 > 服务扩展点管理】。

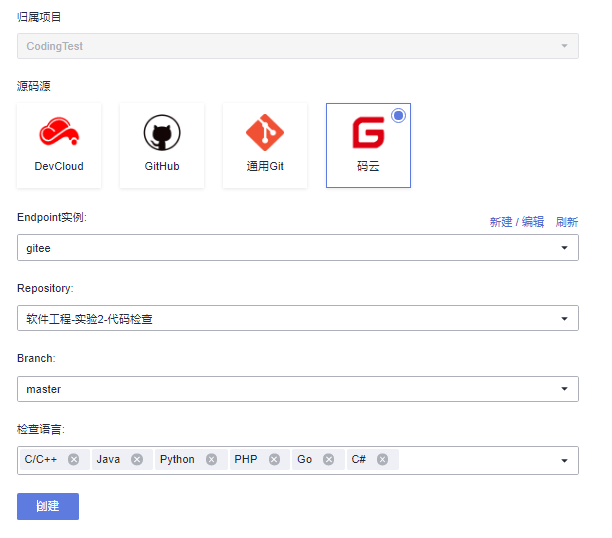
进入“服务扩展点管理”页面，单击“新建服务扩展点”，选择“码云”。

弹出“新建服务扩展点：码云”对话框，根据需要选择对应验证方式（如使用 OAuth 认证方法），并填写相应参数。在码云进行授权。

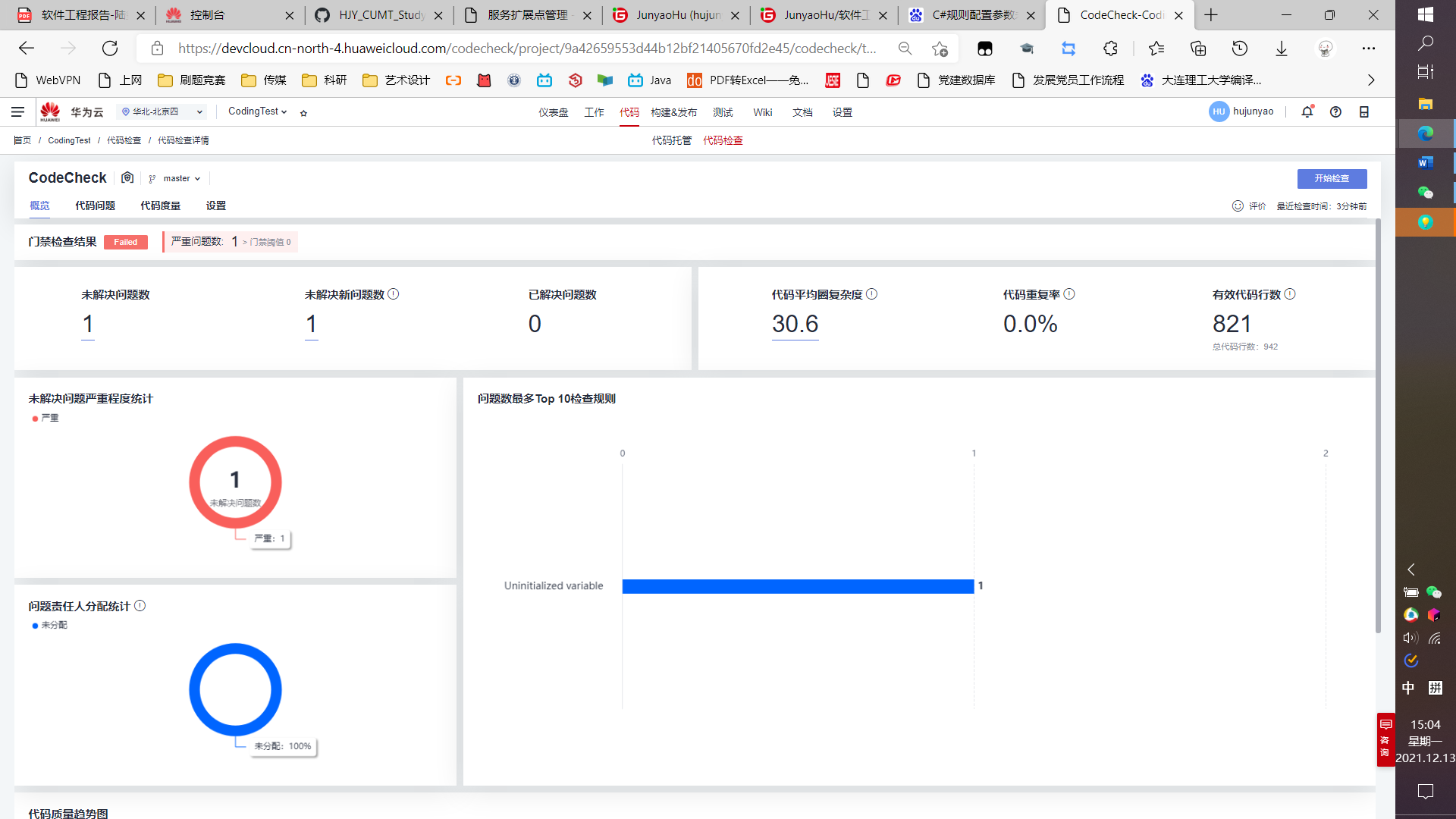


## 进行DevCloud代码检查

在华为云 DevCloud 中新建工作项目，选择【代码 > 代码检查】，创建新任务，选择需要检查的码云仓库。



点击代码检查，可以查看当前仓库的相关检查结果。



特别的，在 DevCloud 中，C# 与其他语言暂不支持同时执行检查，如果选择了 C# 和其他语言，当前后台仅支持检查 C# 文件。



# PHP

## 集成编程环境 PhpStrom

PhpStorm 是 JetBrains 公司开发的一款商业的 PHP 集成开发工具，旨在提高用户效率，可深刻理解用户的编码，提供智能代码补全，快速导航以及即时错误检查。

## PHP 编码规范及实践

任何语言的标准书写都需有一套规范来共同遵守，从而便于相互之交换阅读源代码。根据相关规范范例[1]，整理了部分在平常书写代码过程中，较为高频使用的部分规范，如下表所示。

**PSR-1 基本代码规范**

PHP代码文件必须以 <?php 或 <?= 标签开始；

PHP代码文件必须不带BOM的 UTF-8 编码；

PHP代码中应该只定义类、函数、常量等声明，或其他会产生从属效应 的操作（如：生成文件输出以及修改.ini配置文件等），二者只能选其一；

命名空间以及类必须符合 PSR 的自动加载规范：PSR-0 或 PSR-4 中的一个；

类的命名必须遵循 StudlyCaps 大写开头的驼峰命名规范；

类中的常量所有字母都必须大写，单词间用下划线分隔；

方法名称必须符合 camelCase 式的小写开头驼峰命名规范。

**PSR-2代码风格规范**

代码必须遵循 PSR-1 中的编码规范。

代码必须使用4个空格符而不是 tab键进行缩进。

每个 namespace 命名空间声明语句和 use 声明语句块后面，必须插入一个空白行。

类、方法的开始花括号({)必须写在函数声明后自成一行，结束花括号(})也必须写在函数主体后自成一行。

类的属性和方法必须添加访问修饰符（private、protected 以及 public）， abstract 以及 final必须声明在访问修饰符之前，而 static必须声明在访问修饰符之后。

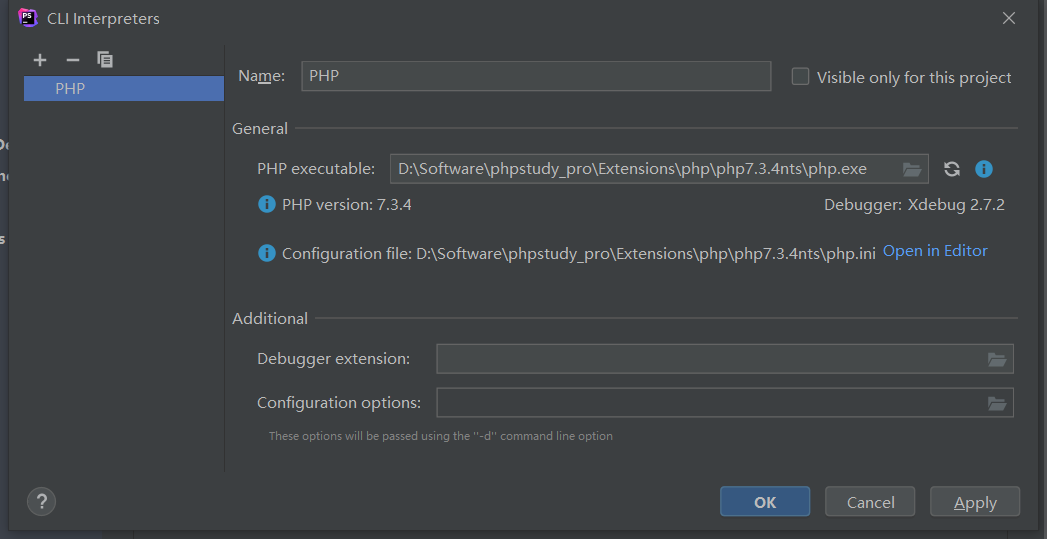
控制结构的关键字后必须要有一个空格符，而调用方法或函数时则一定不能有。

控制结构的开始花括号({)必须写在声明的同一行，而结束花括号(})必须写在主体后自成一行。

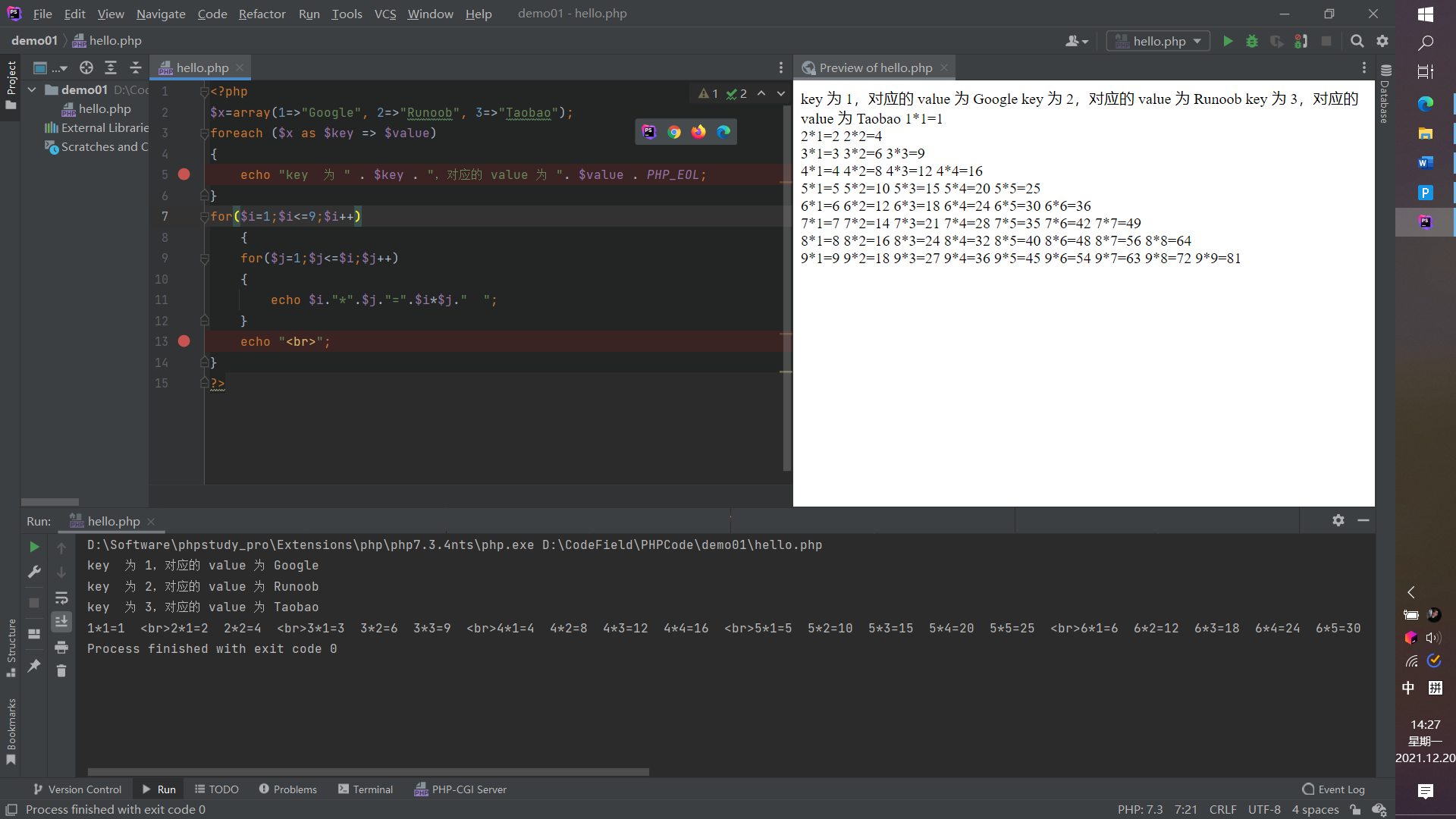
控制结构的开始左括号后和结束右括号前，都一定不能有空格符。

## PhpStrom 环境基本使用方法

打开 PhpStrom，对环境进行配置，具体配置如下。在配置前，还需要安装 PhpStudy提供的PHP编译器，也需要安装 Xdebug 调试软件。



下图是PhpStrom运行界面，实现了键值对的输出与九九乘法表的输出，对字典和for循环进行了学习。



## 华为云PHP代码检查

在华为云 DevCloud 中新建工作项目，选择【代码 > 代码检查】，创建新任务，选择需要检查的码云仓库【软件工程-实验2-代码检查】。将代码上传至码云Gitee，同时华为云 DevCloud 链接到Gitee，对代码进行代码检查。具体见“代码检查”一章。

# Java

## 集成编程环境 IDEA

IDEA 全称 IntelliJ IDEA，是java编程语言开发的集成环境。IntelliJ在业界被公认为最好的java开发工具，尤其在智能代码助手、代码自动提示、重构、JavaEE支持、各类版本工具(git、svn等)、JUnit、CVS整合、代码分析、创新的GUI设计等方面的功能可以说是超常的。

## Java编码规范及实践

根据阿里巴巴的Java开发手册[2]，整理了部分在平常书写代码过程中的注意事项，摘录如下表所示。

**1 编程归约**

（一）命名规约

1. 【强制】 代码中的命名均不能以下划线或美元符号开始，也不能以下划线或美元符号结束。

2. 【强制】 代码中的命名严禁使用拼音与英文混合的方式，更不允许直接使用中文的方式。

3. 【强制】类名使用 UpperCamelCase 风格，必须遵从驼峰形式，但以下情形例外：（领域模型的相关命名）DO / BO / DTO / VO 等。

4. 【强制】方法名、参数名、成员变量、局部变量都统一使用 lowerCamelCase 风格，必须遵从驼峰形式。

5. 【强制】常量命名全部大写，单词间用下划线隔开，力求语义表达完整清楚，不要嫌名字长。

**2 异常日志**

（一）异常处理

1. 【强制】不要捕获 Java 类库中定义的继承自 RuntimeException 的运行时异常类，如：IndexOutOfBoundsException / NullPointerException，这类异常由程序员预检查来规避，保证程序健壮性。

2. 【强制】异常不要用来做流程控制，条件控制，因为异常的处理效率比条件分支低。

**3 MySQL归约**

（一）建表归约

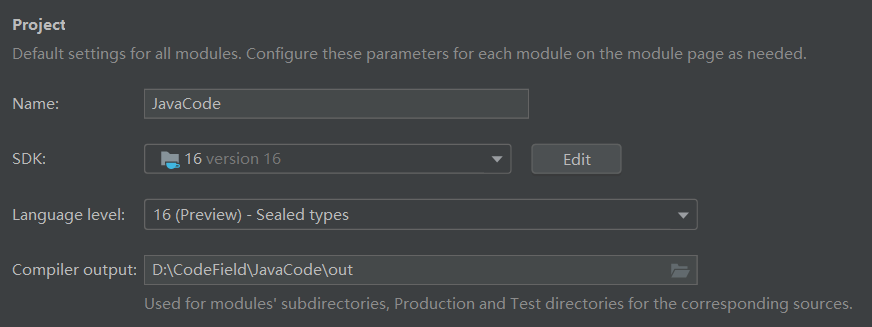
1. 【强制】表达是与否概念的字段，必须使用 is\_xxx 的方式命名，数据类型是 unsigned tinyint。

2. 【强制】表名、字段名必须使用小写字母或数字；禁止出现数字开头，禁止两个下划线中间只 出现数字。

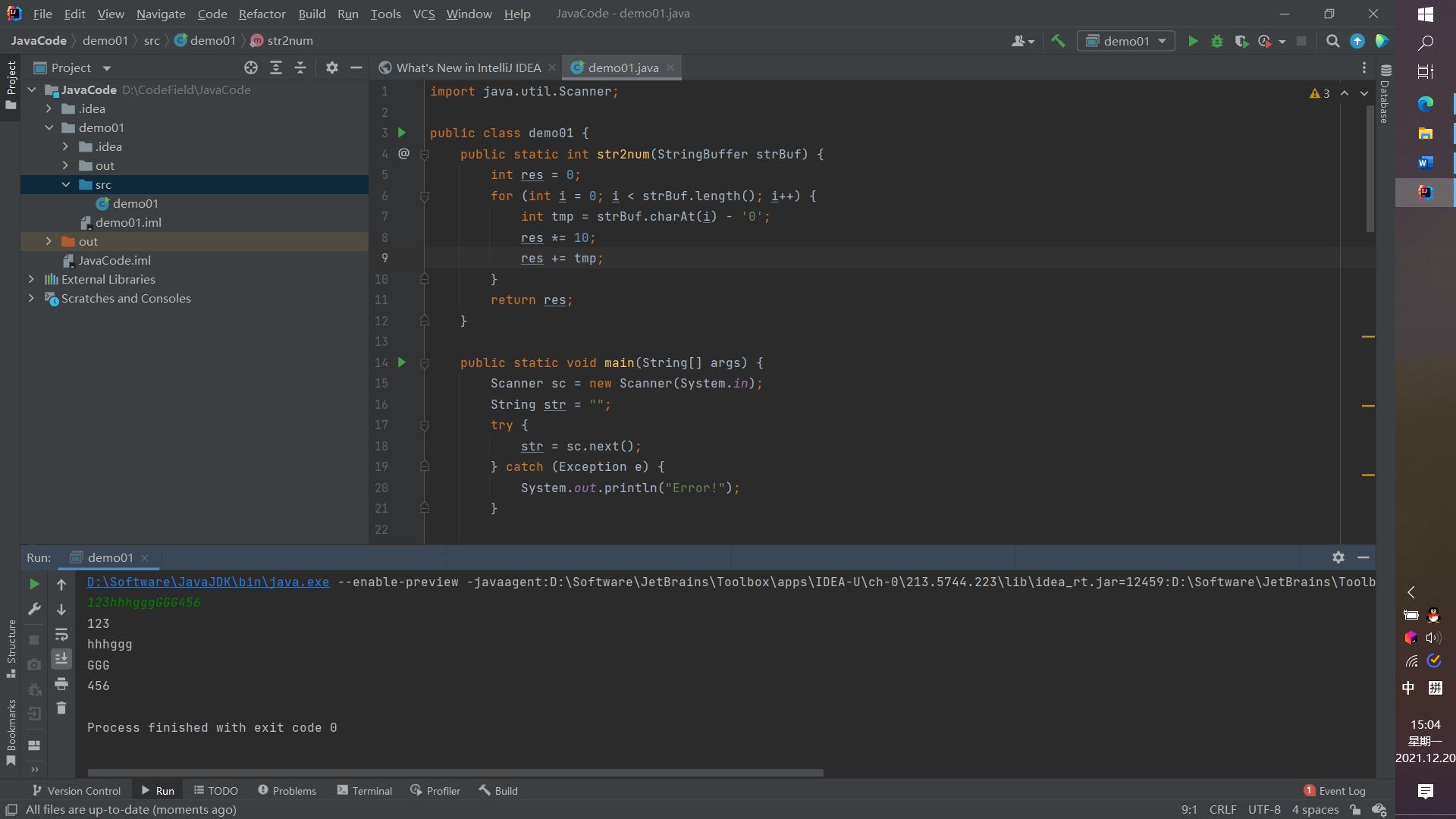
3. 【强制】表名不使用复数名词。

## IDEA 环境基本使用方法

SDK设置，需要提前下载好JDK，并且设置好环境变量。

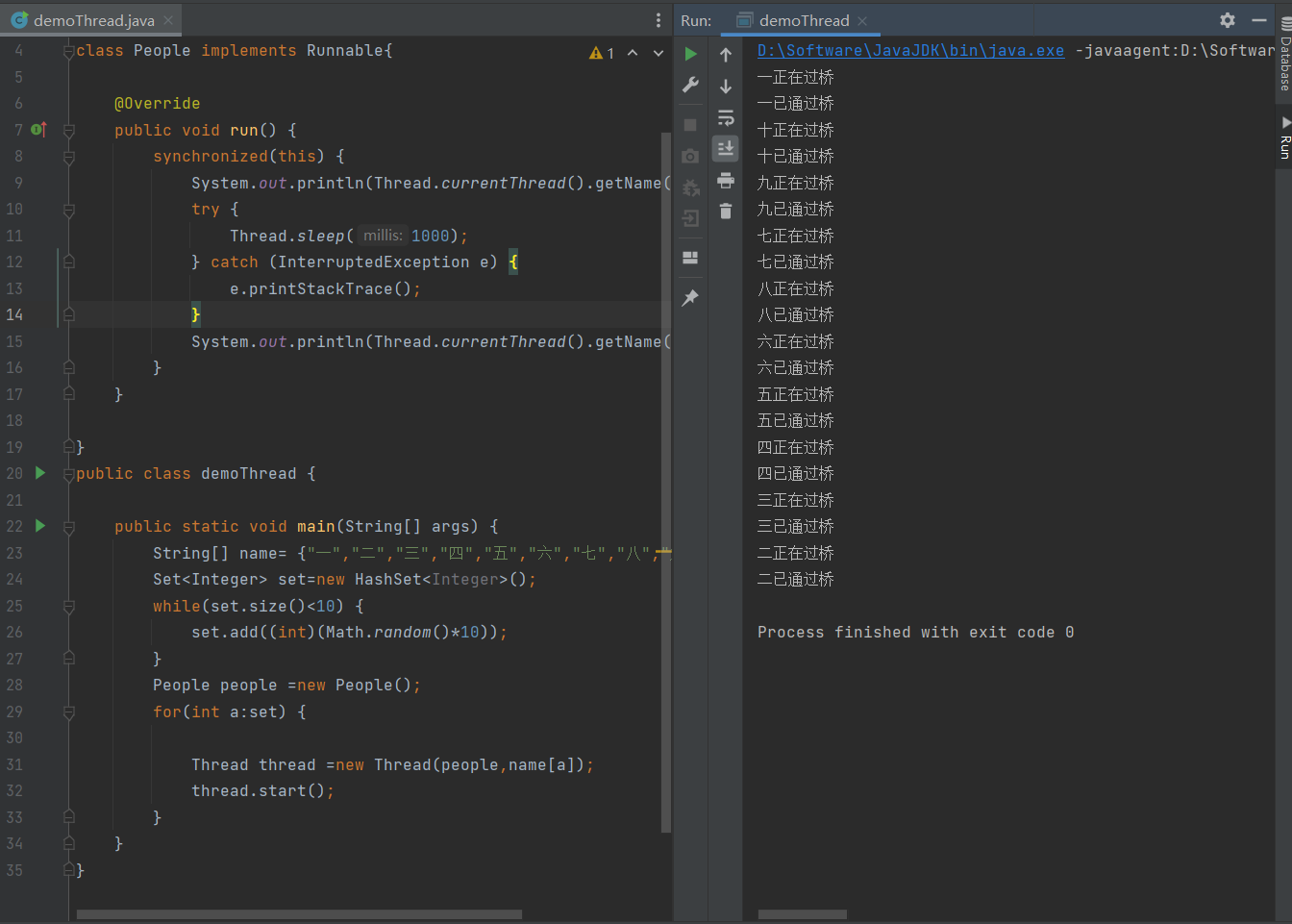


主界面如下。



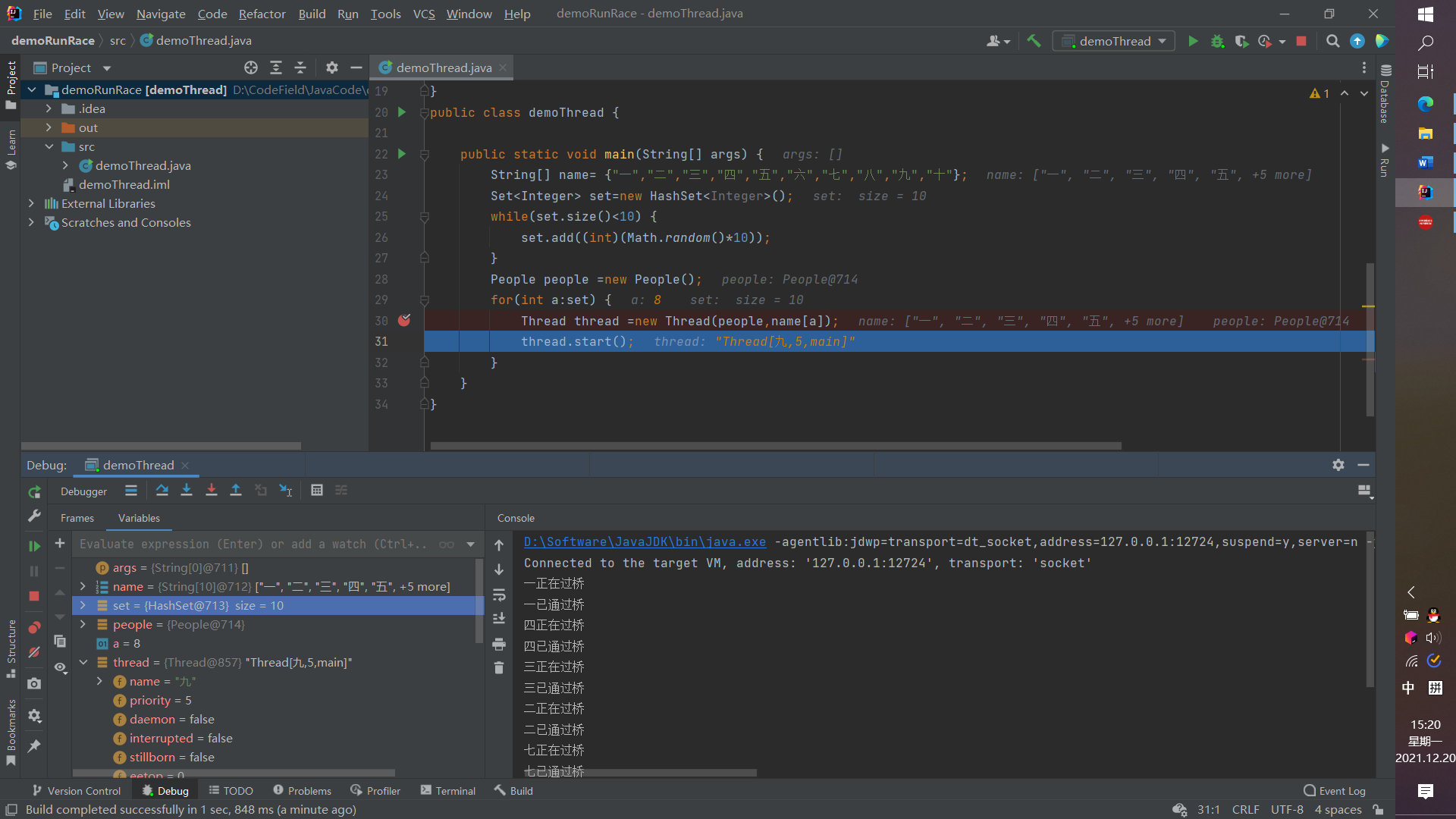
## 进程、线程并发编程

使用Runnable接口和Thread方法编写多线程应用程序，模拟多人过独木桥的模拟。独木桥每次只能通过一个人，每个人通过木桥的时间为1秒，随机生成10个人过此独木桥，显示一下每次通过独木桥人的姓名。



## 调试方法

点击右上角虫子图标进行调试，单击行数右侧可打断点，程序将在断点处停滞，左下角是运行中相关变量记录，右下角是输出控制台，可通过变量值与输出结果对程序正确性进行判断。



## 华为云Java代码检查

在华为云 DevCloud 中新建工作项目，选择【代码 > 代码检查】，创建新任务，选择需要检查的码云仓库【软件工程-实验2-代码检查】。将代码上传至码云Gitee，同时华为云 DevCloud 链接到Gitee，对代码进行代码检查。具体见“代码检查”一章。

# Python

## 集成编程环境 PyCharm

PyCharm 是一种 Python IDE，带有一整套可以帮助用户在使用 Python 语言开发时提高其效率的工具，比如调试、语法高亮、项目管理、代码跳转、智能提示、自动完成、单元测试、版本控制。此外，该 IDE 提供了一些高级功能，以用于支持 Django 框架下的专业 Web 开发。

## Python 编码规范及实践

Python 采用 PEP 8 作为编码规范[3]，其中 PEP 是 Python Enhancement Proposal的缩写，8 代表的是 Python 代码的样式指南。以下列出 PEP 8 中初学者应严格遵守的一些编码规则：如下表所示。

**空格的使用**

1. 使用空格来表示缩进而不要用制表符（Tab）。

2. 和语法相关的每一层缩进都用4个空格来表示。

3. 每行的字符数不要超过79个字符，如果表达式因太长而占据了多行，除了首行之外的其余各行都应该在正常的缩进宽度上再加上4个空格。

4. 函数和类的定义，代码前后都要用两个空行进行分隔。

5. 在同一个类中，各个方法之间应该用一个空行进行分隔。

6. 二元运算符的左右两侧应该保留一个空格，而且只要一个空格就好。

**表达式和语句**

1. 采用内联形式的否定词，而不要把否定词放在整个表达式的前面。例如if a is not b就比if not a is b更容易让人理解。

2. 不要用检查长度的方式来判断字符串、列表等是否为None或者没有元素，应该用if not x这样的写法来检查它。

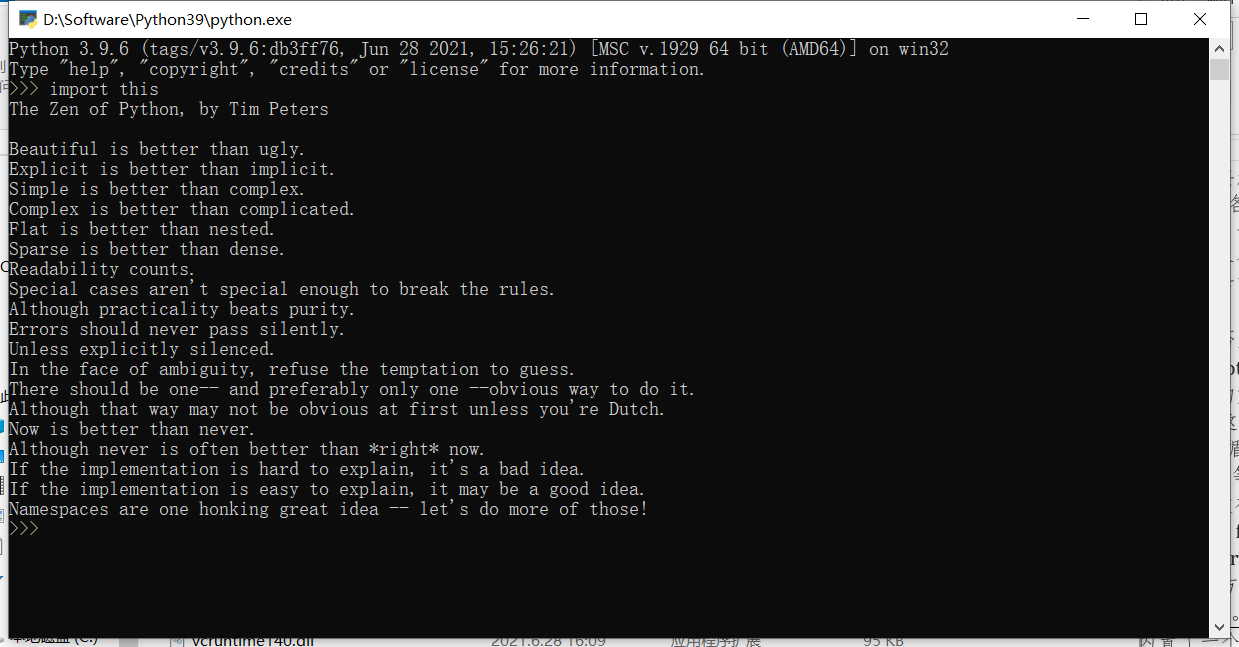
3. 就算if分支、for循环、except异常捕获等中只有一行代码，也不要将代码和if、for、except等写在一起，分开写才会让代码更清晰。

4. import语句总是放在文件开头的地方。

5. 引入模块的时候，from math import sqrt比import math更好。

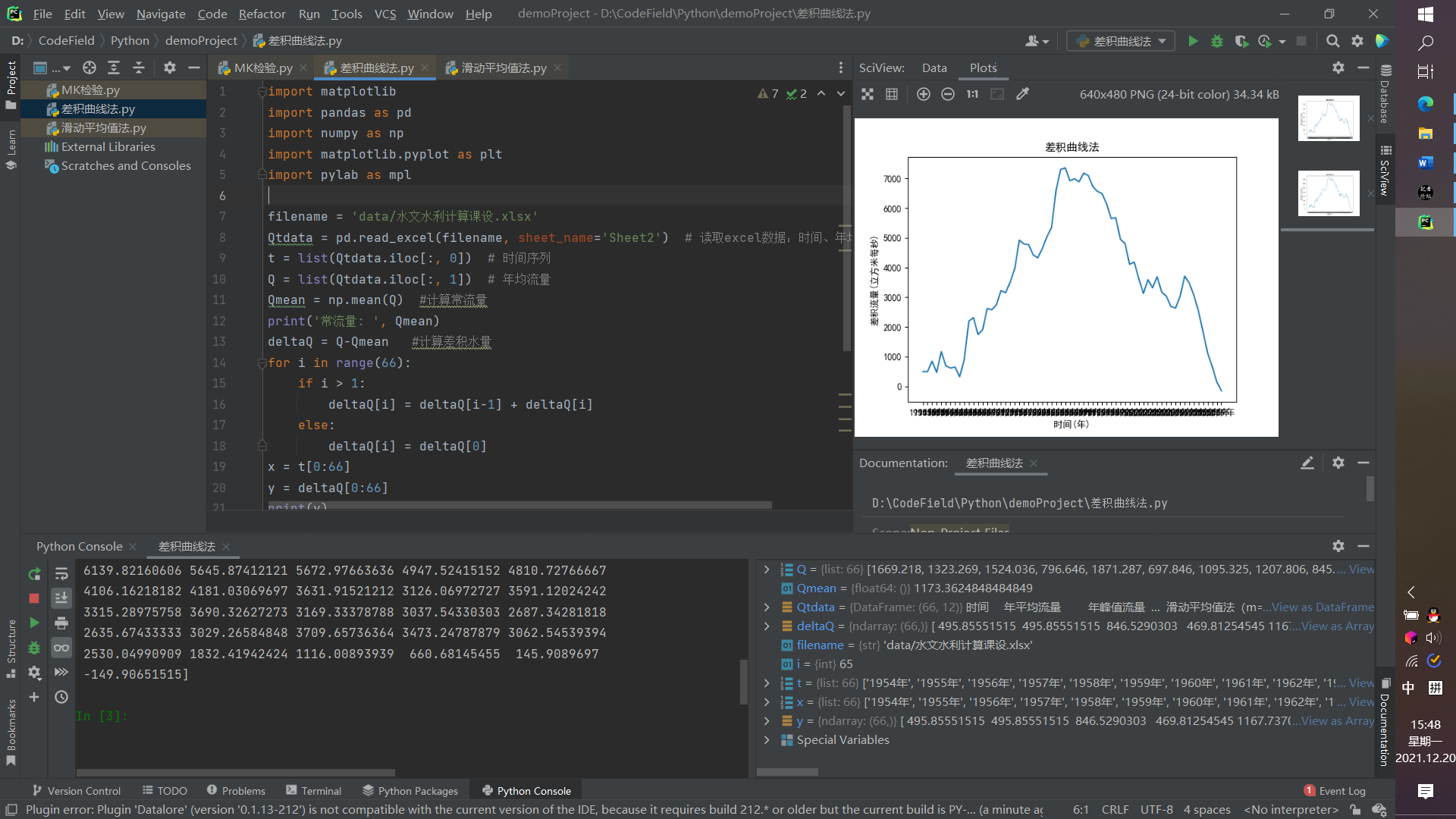
6. 如果有多个import语句，应该将其分为三部分，从上到下分别是Python标准模块、第三方模块和自定义模块，每个部分内部应该按照模块名称的字母表顺序来排列。

Python 中内置了一个彩蛋，简单明了地说明了 Python 的哲学，只要输入 import this，然后按下回车就能看到原文。

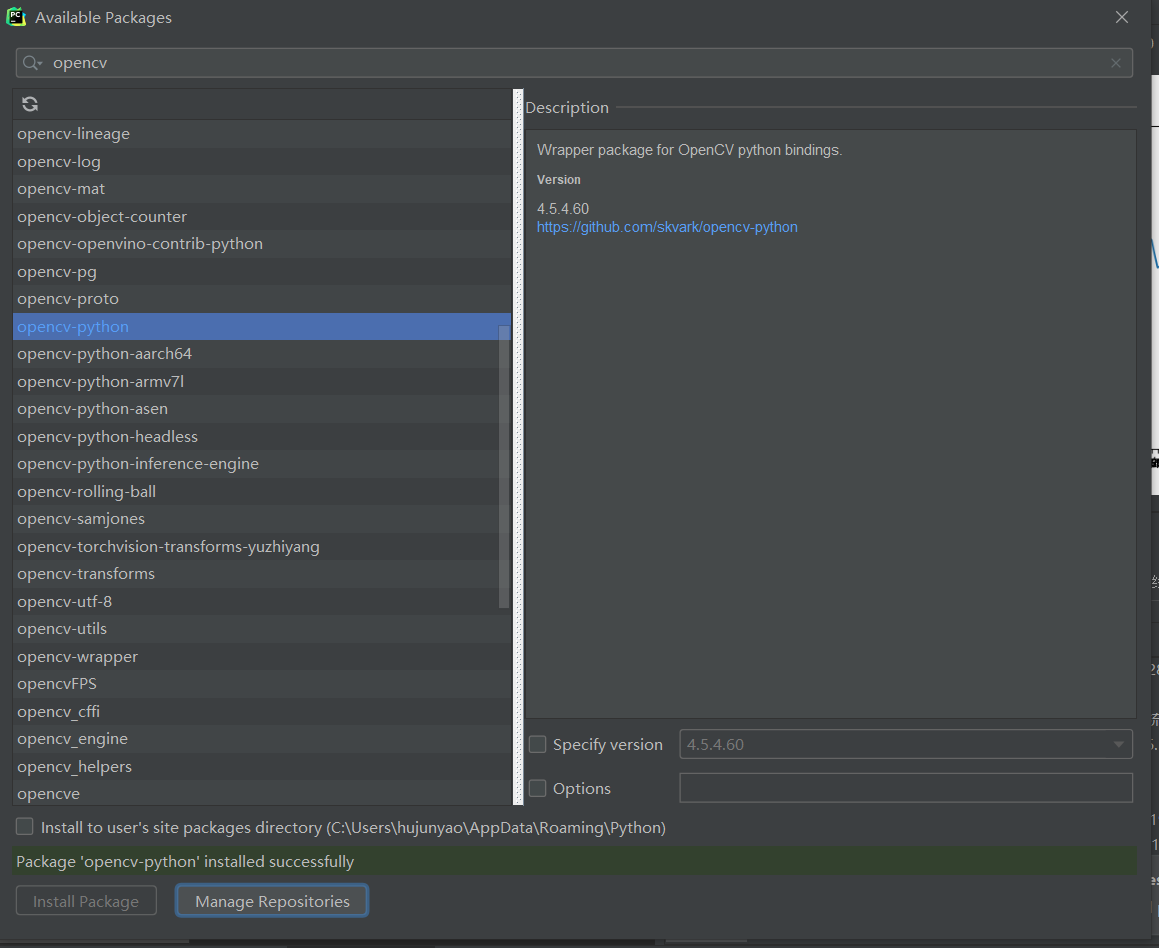


## PyCharm 环境基本使用方法

主界面。



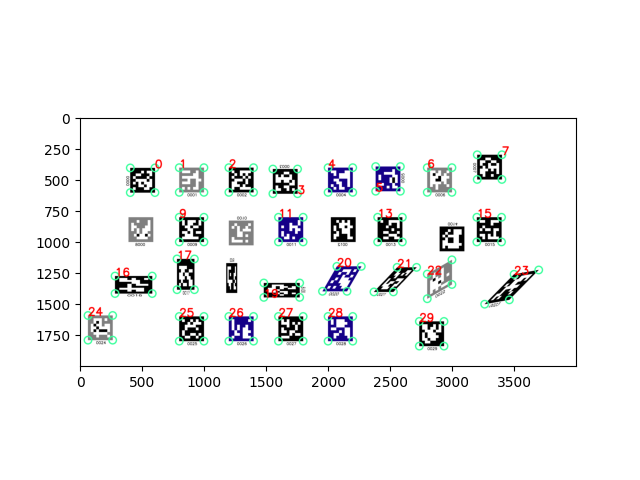
利用PyCharm安装需要的库。



## 基于OpenCV等库的AcuroMaker二维码图像识别项目

本项目基于opencv，对编码规则为DICT\_7X7\_1000的 AcuroMaker 二维码进行图像信息识别，不同的二维码所代表的数字不同，为验证opencv的有效性，将对二维码进行旋转、翻转、修改颜色等处理。该项目需要添加opencv-python进行图像识别，需要numpy库保存图像通道信息和颜色信息，使用matplotlib库对结果进行可视化。

以下是测试样本的测试结果，识别率24/30，测试效果良好。

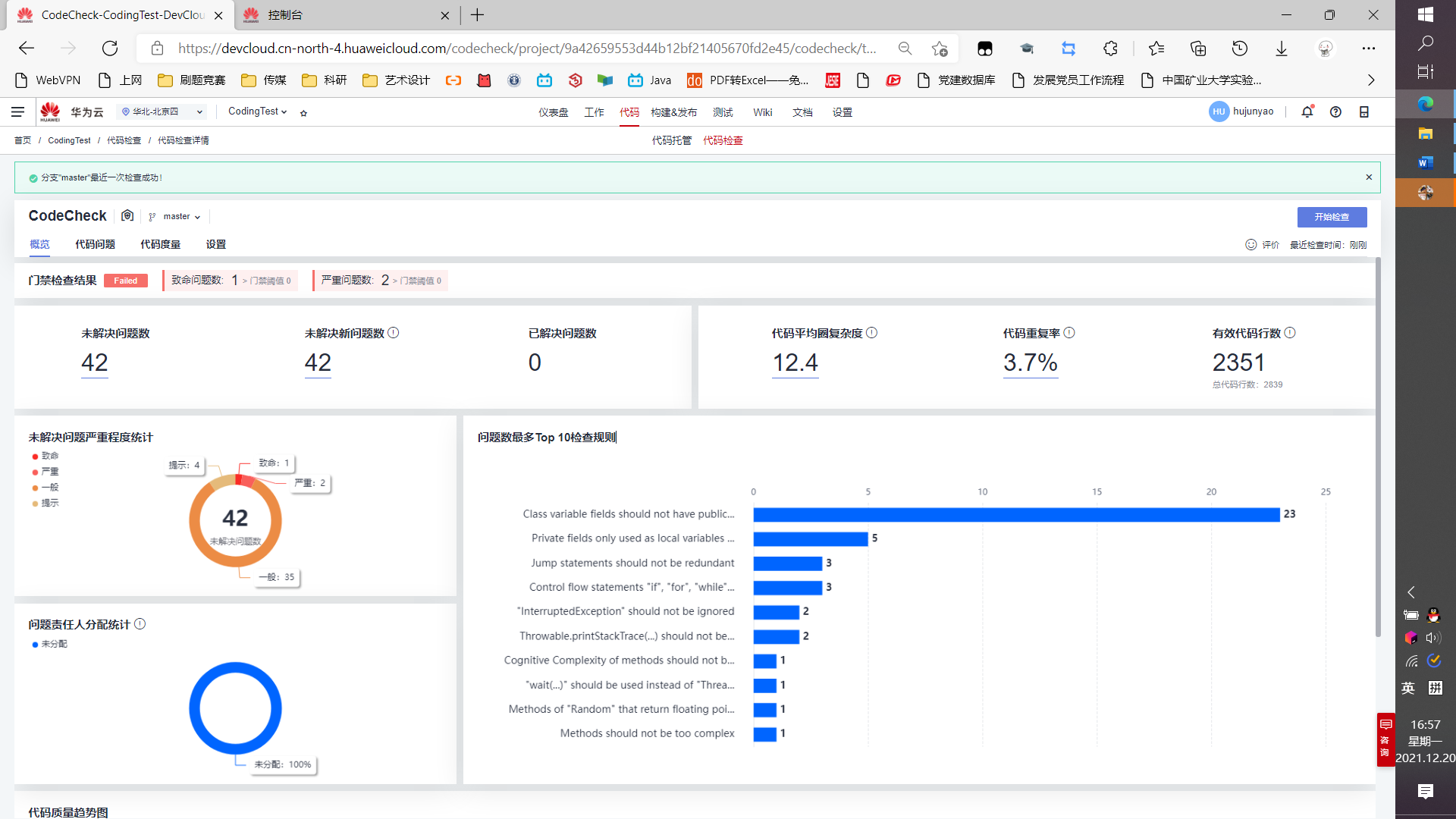


## 华为云Python代码检查

在华为云 DevCloud 中新建工作项目，选择【代码 > 代码检查】，创建新任务，选择需要检查的码云仓库【软件工程-实验2-代码检查】。将代码上传至码云Gitee，同时华为云 DevCloud 链接到Gitee，对代码进行代码检查。具体见“代码检查”一章。

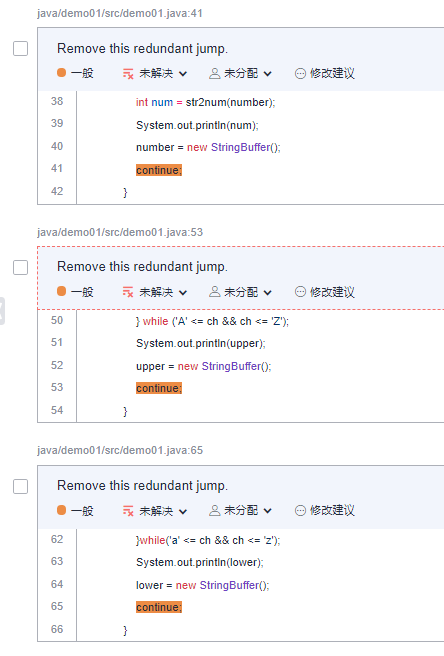
# 代码检查

除了上述代码，在Gitee中也添加了部分曾经写过的工程文件，一同在华为云DevCloud工作区中进行代码检查，检查结果如下。



其中，出现此时最多的问题是：是类变量字段不应该具有公共可访问性（Class variable fields should not have public accessibility，23次）。其他问题还包括：仅在方法中作为局部变量使用的私有字段应该成为局部变量（Private fields only used as local variables in methods should become local variables，5次）等等问题。

在发现的42个问题中，100%都是Java相关的问题，包括异常处理、线程、变量命名等问题。选择【修改建议】，华为云会给出该问题相关的解决方法，方便程序员进行修改。



# 实验感悟

这次实验，学习 PHP 、 Java、 Python 集成编程环境，熟悉各语言的编码规范。学习了 JetBrains 全家桶中的 PhpStrom 、 IDEA 、 PyCharm，结合着课堂上所学，更进一步的明白了代码编写不能仅凭个人志趣，要符合规范。此外，实验中在阅读代码规范的过程中，发现了许多自己以前不注意的点。其中，很大一部分的原因在于以往所编写的代码太小不够完整。对比规范，让我切身的感受到了编写易懂又便于维护的代码是多么的重要又不容易，让我不再想当然的觉得软件开发只是敲代码，只是一件凭灵感肆意运行的事情。未来将学以致用，将软件面向对象的分析方法真正的在实际项目中加以运用，同时在平常学习中，逐步培养起规范编写代码的习惯，更好的开发软件。此外，我觉得 JetBrains 系列软件的界面风格非常统一，学会了一个开发IDE就能很快迁移到另外的IDE上面，同时界面交互非常友好，这也是我们设计软件UI的时候需要注意的点。

# 参考文献

1. Introduction PHP编码规范（中文版） 看云https://www.kancloud.cn/thinkphp/php-fig-psr/3139
2. 阿里巴巴Java开发手册（嵩山版）https://github.com/alibaba/p3c/blob/master/Java%E5%BC%80%E5%8F%91%E6%89%8B%E5%86%8C%EF%BC%88%E5%B5%A9%E5%B1%B1%E7%89%88%EF%BC%89.pdf
3. PEP 8 -- Style Guide for Python Code https://www.python.org/dev/peps/pep-0008/

# 附件

附件1：

PHP面向过程代码：

键值对输出与九九乘法表的打印

<?php  
$x=array(1=>"Google", 2=>"Runoob", 3=>"Taobao");  
foreach ($x as $key => $value)  
{  
 echo "key 为 " . $key . "，对应的 value 为 ". $value . *PHP\_EOL*;  
}  
  
$j = 1;  
while($j<=9){  
 $i = 1;  
 while($i<=$j){  
 echo "{$i}x{$j}=".($i\*$j)." ";  
 $i++;  
 }  
 echo "<br />";  
 $j++;  
}

附件2：

Java 过独木桥多线程编程代码：

import java.util.HashSet;  
import java.util.Set;  
  
class People implements Runnable{  
  
 @Override  
 public void run() {  
 synchronized(this) {  
 System.out.println(Thread.currentThread().getName()+"正在过桥");  
 try {  
 Thread.sleep(1000);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 System.out.println(Thread.currentThread().getName()+"已通过桥");  
 }  
 }  
  
}  
public class demoThread {  
  
 public static void main(String[] args) {  
 String[] name= {"一","二","三","四","五","六","七","八","九","十"};  
 Set<Integer> set=new HashSet<Integer>();  
 while(set.size()<10) {  
 set.add((int)(Math.random()\*10));  
 }  
 People people =new People();  
 for(int a:set) {  
 Thread thread =new Thread(people,name[a]);  
 thread.start();  
 }  
 }  
}

附件2：

Python二维码识别代码：检测图片中的编码点，并在图片上画出结果

# -\*- coding:utf-8 \_\*-  
import cv2  
import cv2.aruco as aruco  
import numpy as np  
import matplotlib.pyplot as plt  
  
'''检测图片中的编码点，并在图片上画出结果'''  
  
dict1 = aruco.getPredefinedDictionary(aruco.DICT\_7X7\_1000) # 编码点的类型,与生成的时候对应  
font = cv2.FONT\_HERSHEY\_SIMPLEX # 使用默认字体  
  
'''载入图片'''  
Img = cv2.imread('testBigImg\_new.png')  
h = Img.shape[0] # 图片的尺寸  
w = Img.shape[1]  
  
'''检测编码点，并在图片上显示检测结果'''  
corners, ids, rejectedImgPoints = aruco.detectMarkers(Img, dict1)  
c = np.array(corners) # 检测出来的marker  
id1 = np.array(ids).flatten() # 对应marker的编码信息  
ImgDetection = Img  
for i in range(0, c.shape[0]): # 遍历每一个marker  
 for j in range(0, c[i][0].shape[0]): # 遍历marker的4个角点  
 ImgDetection = cv2.circle(Img, (int(c[i][0][j][0]), int(c[i][0][j][1])), 30, (55, 255, 155), 10) # 圈出角点  
 ImgDetection = cv2.putText(ImgDetection, str(id1[i]),  
 (int(c[i][0][0][0]),  
 int(c[i][0][0][1])),  
 font, 3, (255, 0, 0), 10) # 写上id信息  
plt.imshow(ImgDetection, 'gray')  
plt.show()  
cv2.imwrite('BigImg\_detect.png', ImgDetection)