

Java开发利器：IDEA的安装与使用（上）

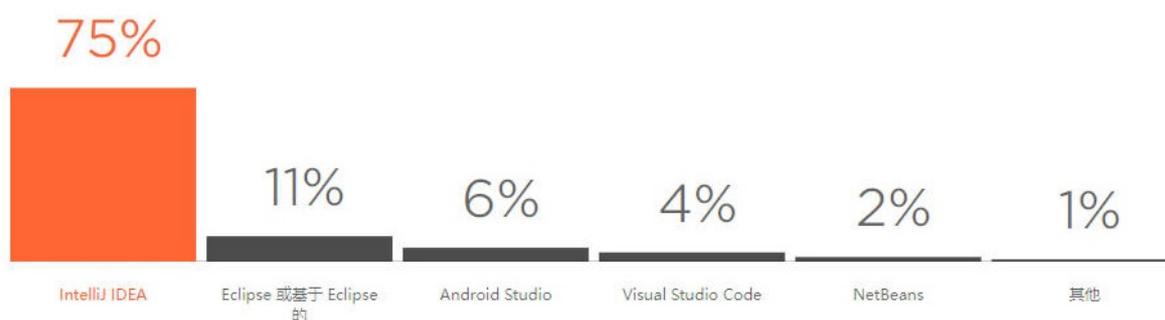
【本套课程适用人群】

- 初学Java语言，熟悉了记事本、EditPlus、NotePad++或Sublime Text3等简易开发工具的Java初学者
- 熟练使用其他Java集成开发环境（IDE），需要转向IDEA工具的Java工程师们
- 关注IDEA各方面特性的Java程序发烧友、爱好者

【Why IDEA ?】

四分之三的 Java 开发者选择 IntelliJ IDEA

您最常使用哪种IDE/编辑器进行Java开发？



【注】JetBrains官方说明：

尽管我们采取了多种措施确保受访者的代表性，但结果可能会略微偏向 JetBrains 产品的用户，因为这些用户更有可能参加调查。

此外，2022年，某美国软件开发商在对近千名专业的Java开发者调研后，发布了《2022年Java开发者生产力报告》。报告提到：JetBrains 的 IntelliJ IDEA是最受欢迎的 Java IDE，占 48%，其次是 Eclipse，占 24%，Visual Studio Code 占 18%。

【Why IDEA 2022.x】

在Java 17正式发布之前，Java开发框架Spring率先在官博宣布，Spring Framework 6和Spring Boot 3计划在2022年第四季度实现总体可用性的高端基线：

- Java 17+(来自 Spring Framework 5.3.x 线中的 Java 8-17)
- Jakarta EE 9+ (来自Spring框架5.3.x 线中的 Java EE 7-8)

Spring 官方说明：<https://spring.io/blog/2022/01/20/spring-boot-3-0-0-m1-is-now-available>

Spring Boot 3.0.0-M1 is now available

RELEASES | PHIL WEBB | JANUARY 20, 2022 0 COMMENT

On behalf of the team and everyone who has contributed, I'm happy to announce that Spring Boot [3.0.0-M1](#) has been released and is now available from <https://repo.spring.io/milestone>.

This milestone starts our exciting journey to the next generation of the Spring Framework and raises our baseline from Java 8 to Java 17. We are planning to release a new milestone of Spring Boot 3.0 every two months. M2 should arrive on March 24 and we are planning on a GA release in late November.

意味着：springboot3.0 是需要用java17和spring6.0为基础建设。如果从企业选型最新springboot3.0 作为架构来说，它搭配jdk17肯定是标配了。

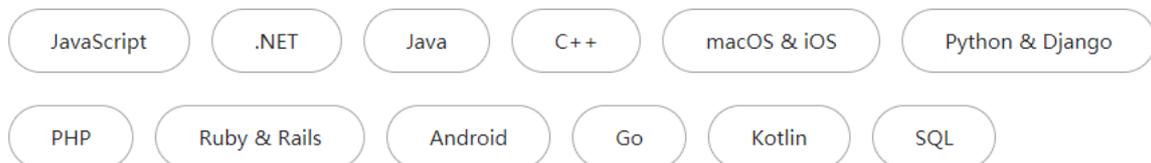
本着"工欲善其事必先利其器"的精神，本套课程从IDEA的介绍、安装、设置入手，讲解IDEA中项目的创建、快捷键与模板的使用、断点调试、常用插件等。

1. 认识IntelliJ IDEA

1.1 JetBrains 公司介绍

IDEA, 是 JetBrains (<https://www.jetbrains.com/>)公司的产品，该公司成立于2000年，总部位于捷克的布拉格，致力于为开发者打造最高效智能的开发工具。

Whichever technologies you use, there's a JetBrains tool to match.

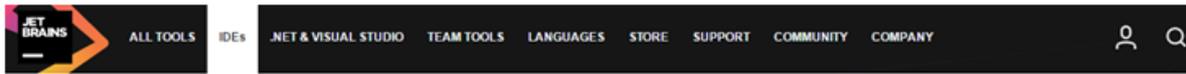


公司旗下还有其它产品，比如：

- WebStorm：用于开发 JavaScript、HTML5、CSS3 等前端技术
- PyCharm：用于开发 python
- PhpStorm：用于开发 PHP
- RubyMine：用于开发 Ruby/Rails
- AppCode：用于开发 Objective - C/Swift
- CLion：用于开发 C/C++
- DataGrip：用于开发数据库和 SQL
- Rider：用于开发.NET

- GoLand: 用于开发 Go

用于开发 Android的Android Studio, 也是Google 基于 IDEA 社区版进行迭代的。



Check out our IDEs

IntelliJ IDEA The most intelligent Java IDE	PyCharm Python IDE for professional developers	WebStorm The smartest JavaScript IDE	PhpStorm Lightning-smart PHP IDE	CLion A smart cross-platform IDE for C and C++
Rider Cross-platform .NET IDE	DataGrip Many databases, one tool	RubyMine The most intelligent Ruby IDE	AppCode Smart IDE for iOS/macOS development	Goland Up and coming Go IDE

1.2 IntelliJ IDEA 介绍

IDEA, 全称 **IntelliJ IDEA**, 是 Java 语言的集成开发环境, 目前已经 (基本) **代替** 了Eclipse的使用。IDEA 在业界被公认为是最好的 Java 开发工具 (之一), 因其 **功能强悍**、**设置人性化**, 而深受 Java、大数据、移动端程序员的喜爱。尤其在智能代码助手、代码自动提示、重构、J2EE支持、Ant、JUnit、CVS 整合、代码审查、创新的 GUI 设计等方面的功能可以说是超常的。

IntelliJ IDEA 在 2015 年的官网上这样介绍自己:

Excel at enterprise, mobile and web development with Java, Scala and Groovy,with all the latest modern technologies and frameworks available out of thebox.



1.3 IDEA的主要优势: (vs Eclipse)

功能强大:

① 强大的整合能力。比如: Git、Maven、Spring等

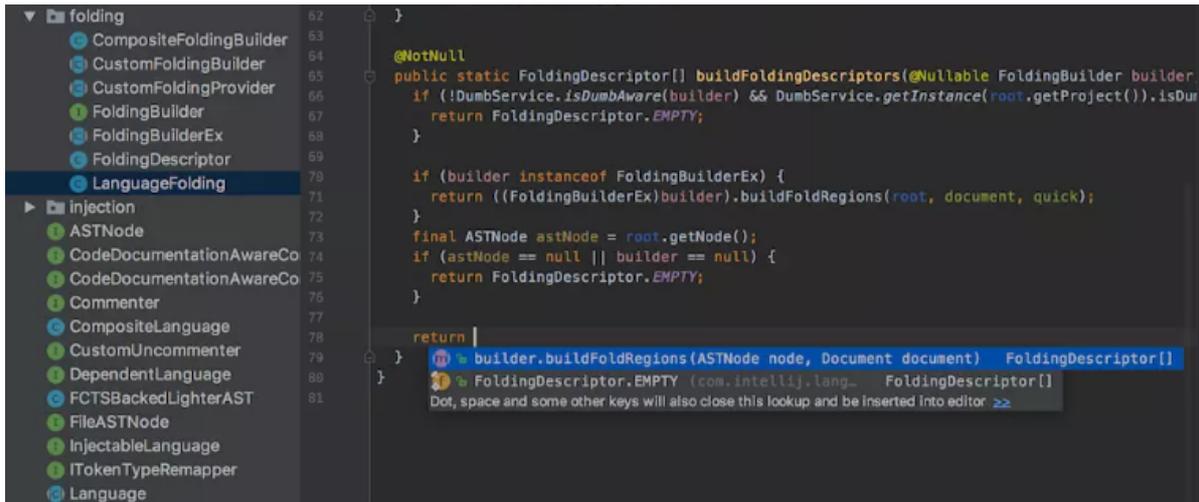
Built-in tools and supported frameworks

 git				
Built-in tools	JVM languages	Enterprise frameworks	Mobile development	Web development
Build tools Version control Decompiler Coverage Database tools/SQL	Java Scala Groovy Kotlin	Spring Java EE JBoss Play Grails App Servers/Clouds	Android React Native Cordova Ionic	JavaScript TypeScript HTML and CSS Node.js Angular React Vue.js

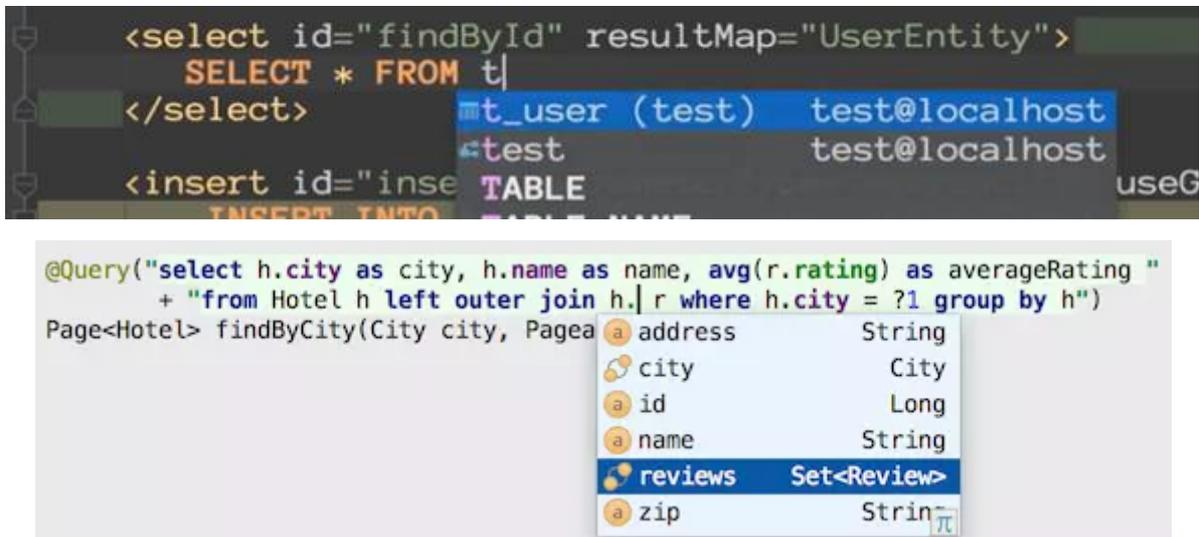
② 开箱即用的体验 (集成版本控制系统、多语言支持的框架随时可用, 无需额外安装插件)

符合人体工程学：

- ① 高度智能（快速的智能代码补全、实时代码分析、可靠的重构工具）



- ② 提示功能的快速、便捷、范围广



- ③ 好用的快捷键和代码模板

- ④ 精准搜索

1.4 IDEA 的下载

- 下载网址：<https://www.jetbrains.com/idea/download/#section=windows>
- IDEA 分为两个版本：**旗舰版(Ultimate)** 和 **社区版(Community)**。
- IDEA的大版本每年迭代一次，大版本下的小版本（如：2022.x）迭代时间不固定，一般每年3个小版本。



Version: 2022.1.2
Build: 221.5787.30
1 June 2022

[Release notes](#)

[System requirements](#)

[Installation instructions](#)

[Other versions](#)

[Third-party software](#)

Download IntelliJ IDEA

[Windows](#) [macOS](#) [Linux](#)

Ultimate

For web and enterprise development

[Download](#) [.exe](#)

Free 30-day trial available

Community

For JVM and Android development

[Download](#) [.exe](#)

Free, built on open source

	IntelliJ IDEA Ultimate	IntelliJ IDEA Community Edition ⓘ
Java, Kotlin, Groovy, Scala	✓	✓
Maven, Gradle, sbt	✓	✓
Git, GitHub, SVN, Mercurial, Perforce	✓	✓
Debugger	✓	✓
Docker	✓	✓
Profiling tools ⓘ	✓	
Spring, Jakarta EE, Java EE, Micronaut, Quarkus, Helidon, and more ⓘ	✓	

两个不同版本的详细对比, 可以参照官网: https://www.jetbrains.com/idea/features/editions_comparison_matrix.html

官网提供的详细使用文档: <https://www.jetbrains.com/help/idea/meet-intellij-idea.html>

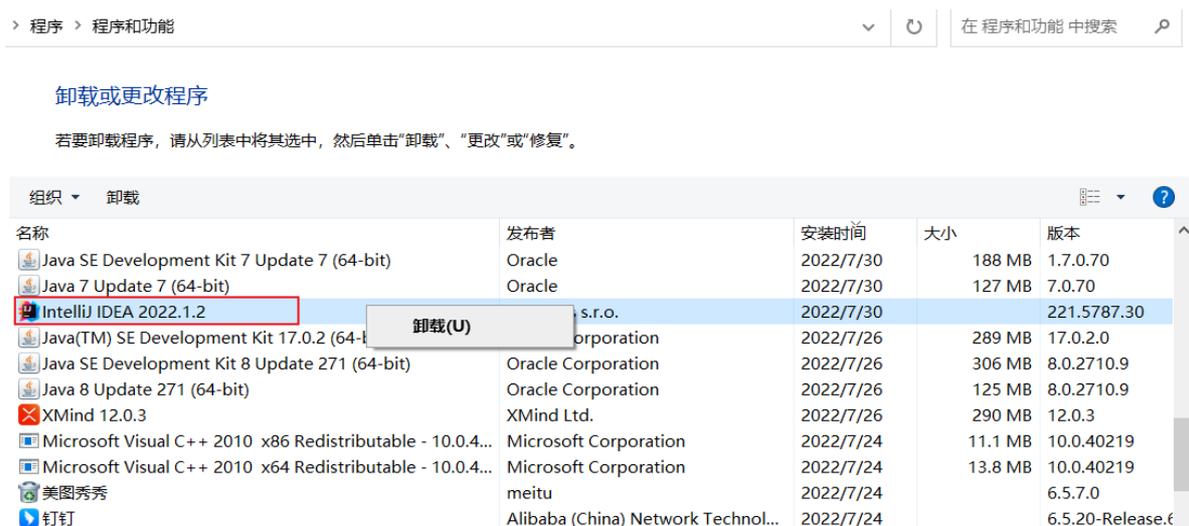
2. 卸载与安装

2.1 卸载过程

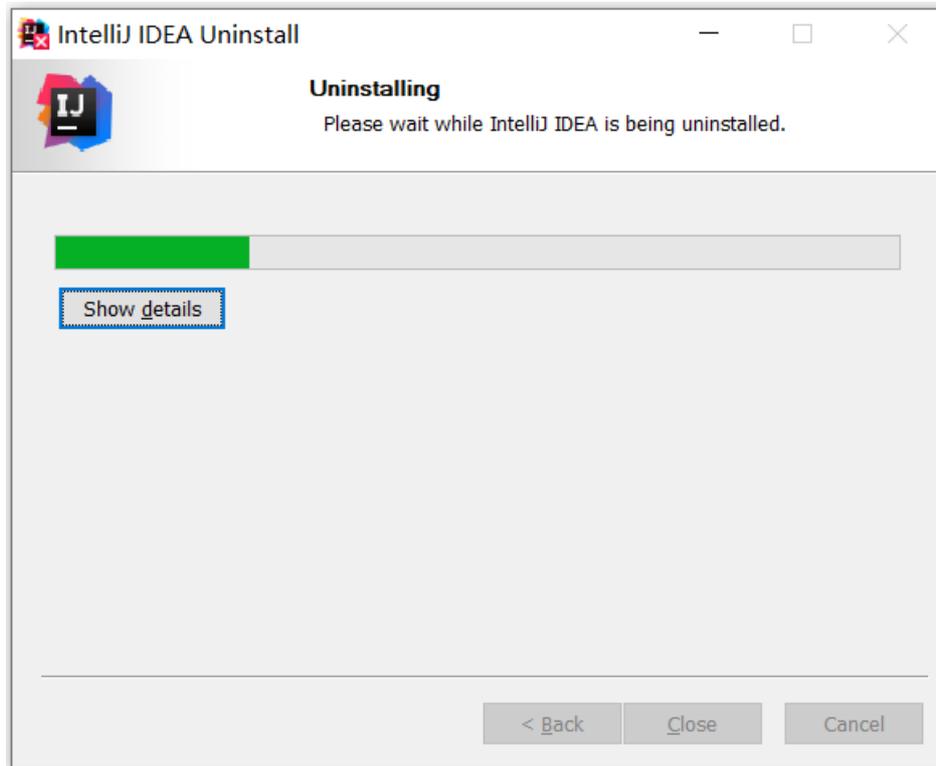
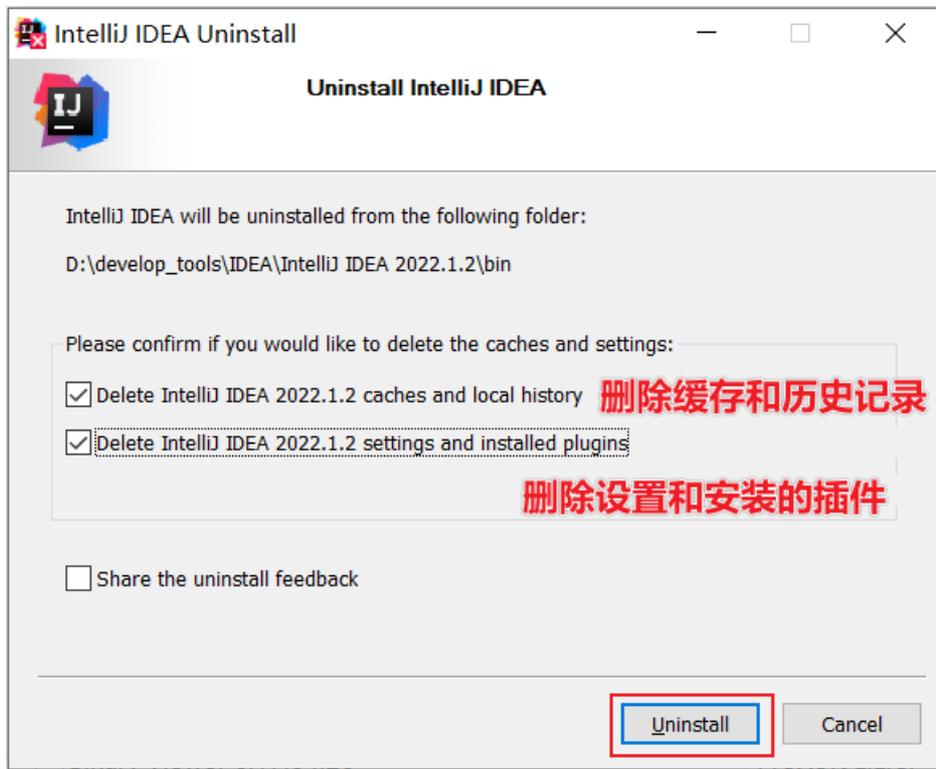
这里以卸载2022.1.2版本为例说明。在【控制面板】找到【卸载程序】

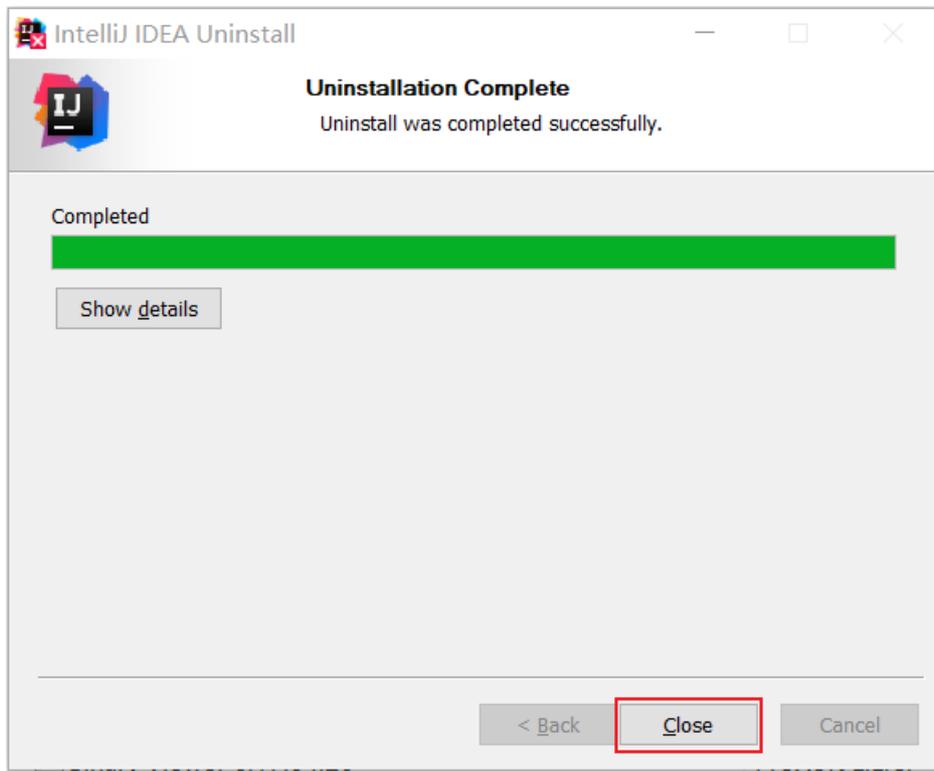


右键点击或左键双击IntelliJ IDEA 2022.1.2进行卸载:



如果需要保留下述数据, 就不要打√。如果想彻底删除IDEA所有数据, 那就打上√。





软件卸载完以后，还需要删除其它几个位置的残留：

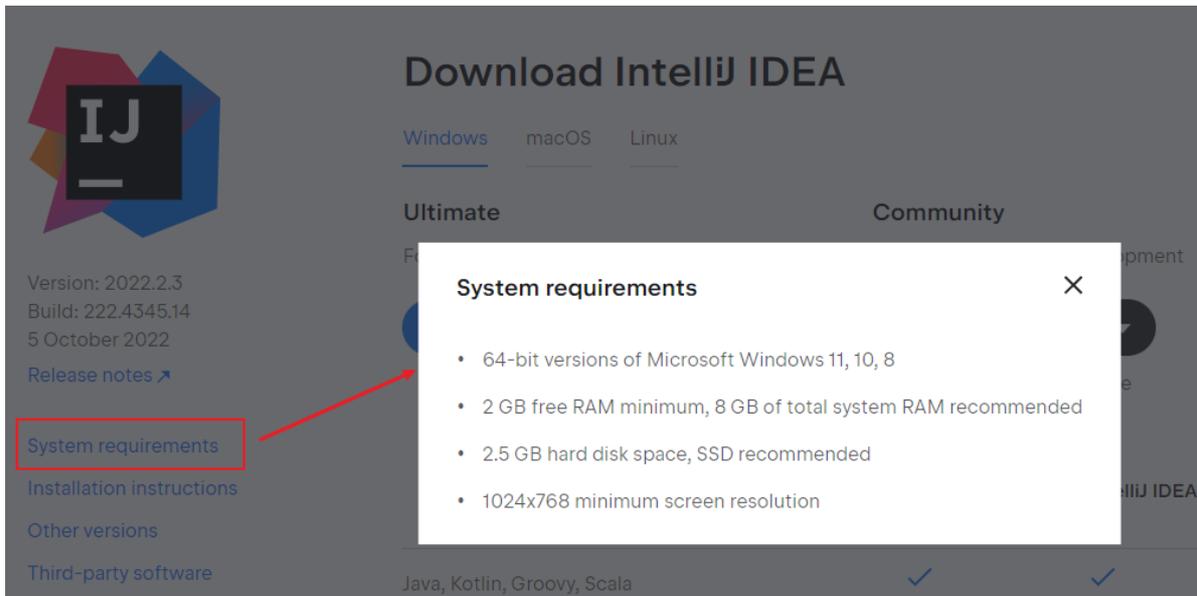
此电脑 > system (C:) > 用户 > shkstart > AppData > Local >

名称	修改日期	类型	大小
EVVIP1001	2022/8/2 16:21	文件夹	
EVWhiteBoard	2022/7/25 10:36	文件夹	
Google	2022/7/19 9:50	文件夹	
JetBrains	2022/10/19 12:09	文件夹	
kingsoft	2022/7/24 21:15	文件夹	
kwmusic	2022/7/24 19:04	文件夹	
main.kts.compiled.cache	2022/7/23 10:39	文件夹	
Meitu	2022/7/24 21:35	文件夹	

> system (C:) > 用户 > shkstart > AppData > Roaming >

名称	修改日期	类型	大小
install_com_aaa	2022/8/16 20:45	文件夹	
IQIYI Video	2022/10/4 11:23	文件夹	
JdReader	2022/8/11 23:49	文件夹	
JetBrains	2022/10/19 12:23	文件夹	
kingsoft	2022/7/24 21:16	文件夹	
lds	2022/7/20 9:32	文件夹	
Macromedia	2022/7/19 10:01	文件夹	
McumRepairer	2022/10/6 13:54	文件夹	
MEISHENG90	2022/8/13 18:03	文件夹	
Microsoft	2022/7/29 17:35	文件夹	

2.2 安装前的准备

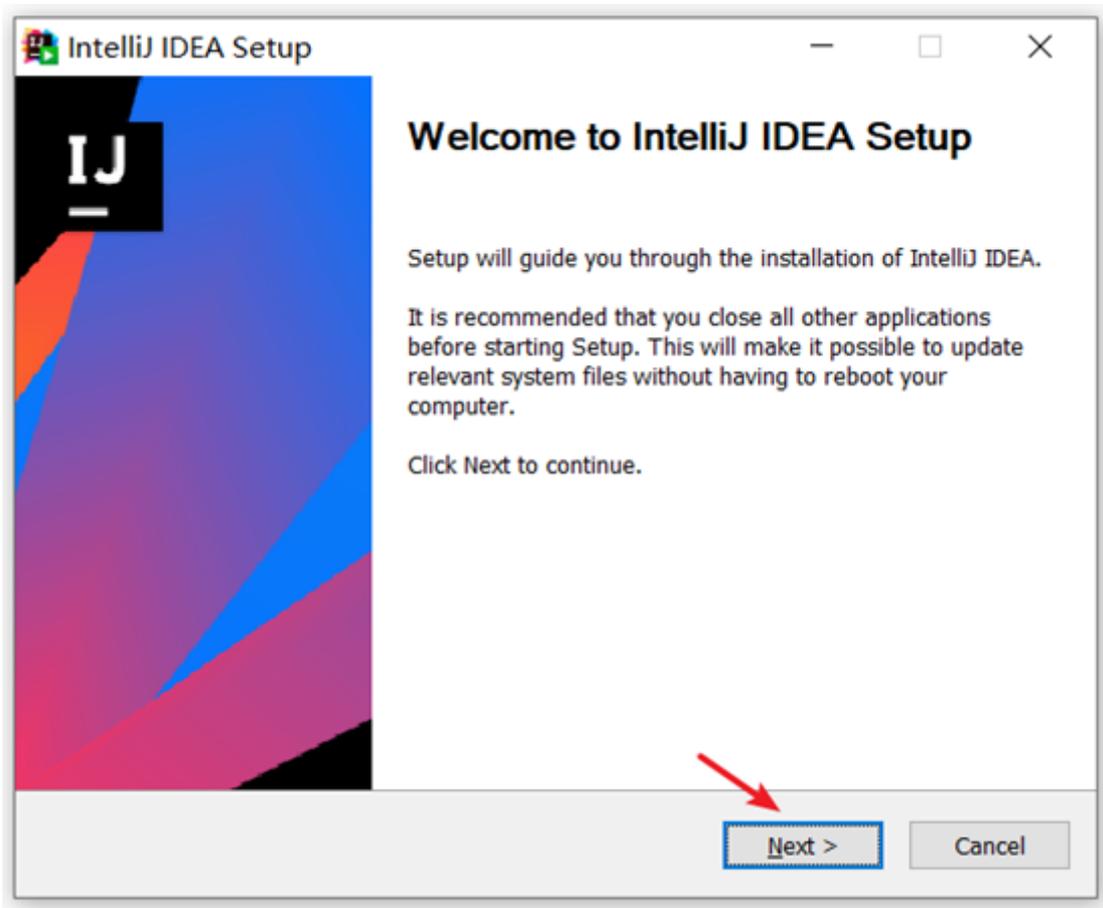


- 64 位 Microsoft Windows 11、10、8
- 最低 2 GB 可用 RAM，推荐 8 GB 系统总 RAM
- 2.5 GB 硬盘空间，推荐 SSD
- 最低屏幕分辨率 1024x768

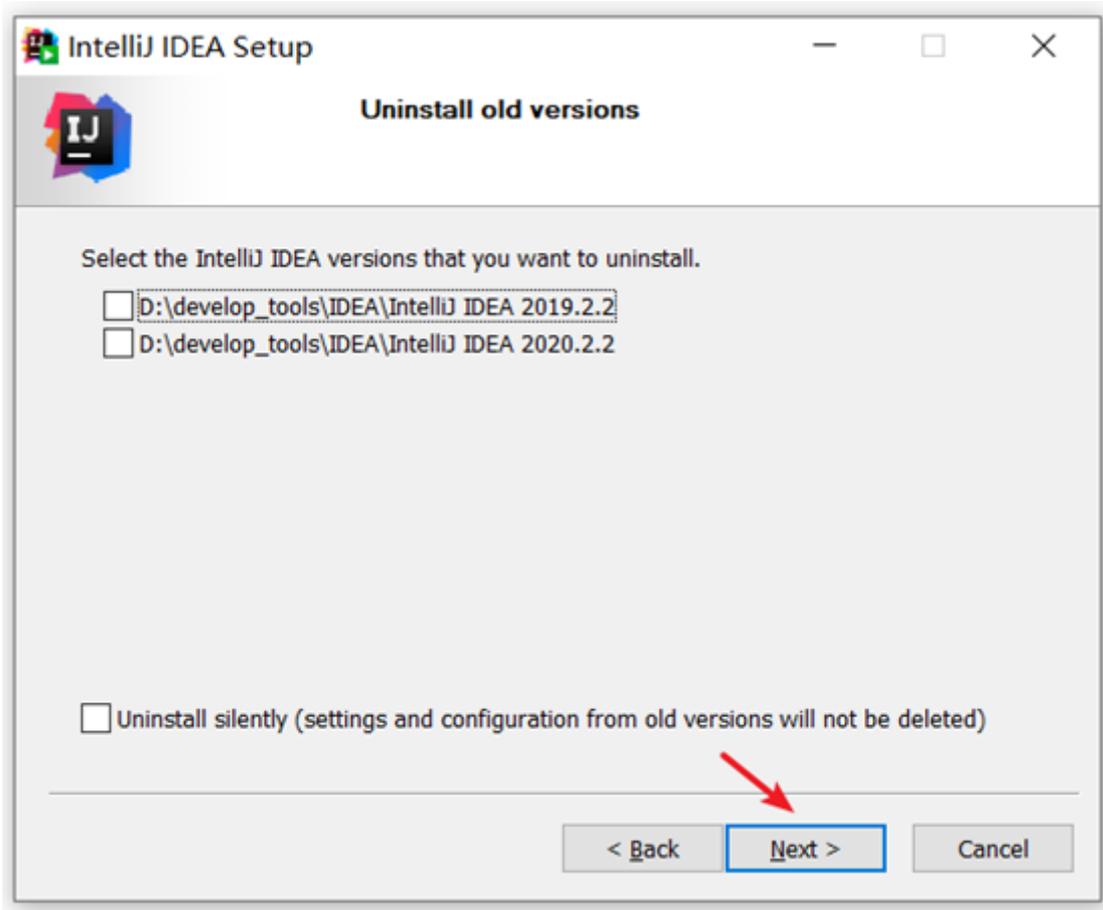
从安装上来看，IntelliJ IDEA 对硬件的要求 **似乎不是很高**。可是在实际开发中并不是这样的，因为 IntelliJ IDEA 执行时会有大量的缓存、索引文件，所以如果你正在使用 Eclipse / MyEclipse，想通过 IntelliJ IDEA 来解决计算机的卡、慢等问题，这基本上是不可能的，本质上你应该对自己的硬件设备进行升级。

2.3 安装过程

- 1、下载完安装包，双击直接安装  `ideaIU-2022.1.2.exe`
- 2、欢迎安装

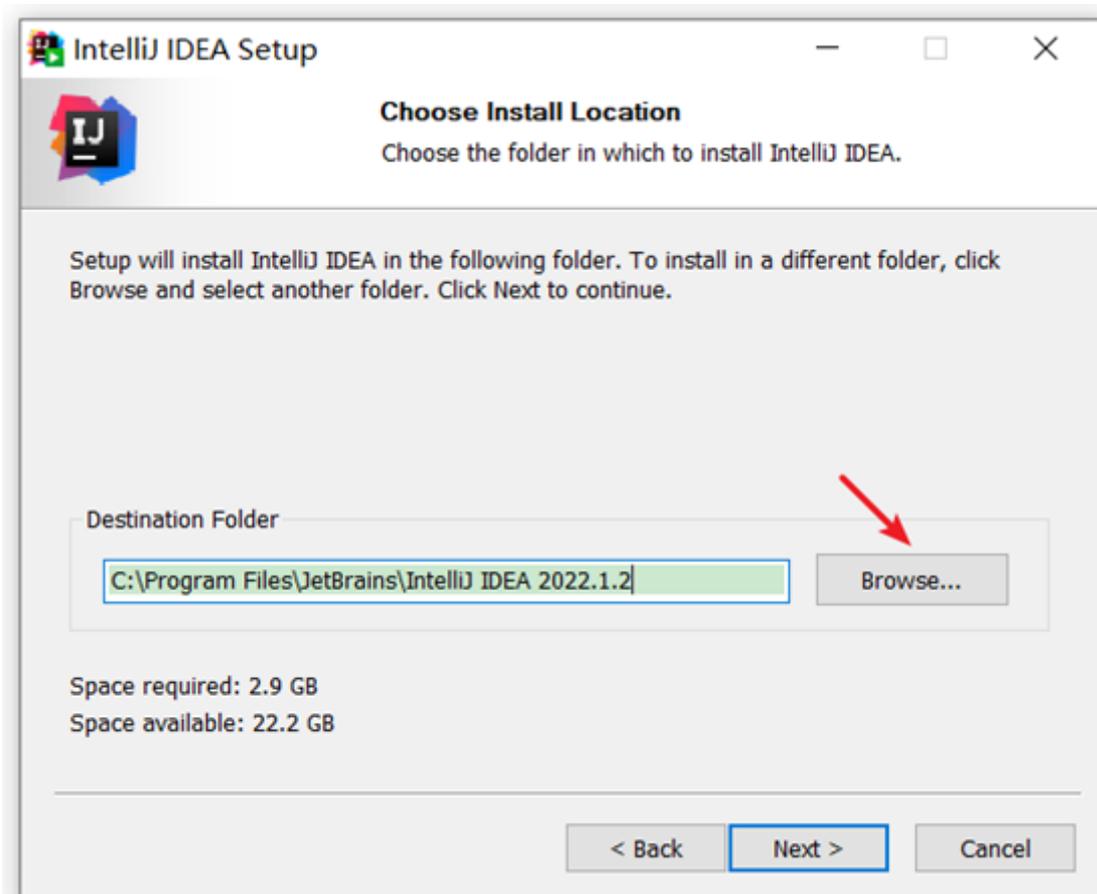


3、是否删除电脑上低版本的IDEA（如果有，可以选择忽略）

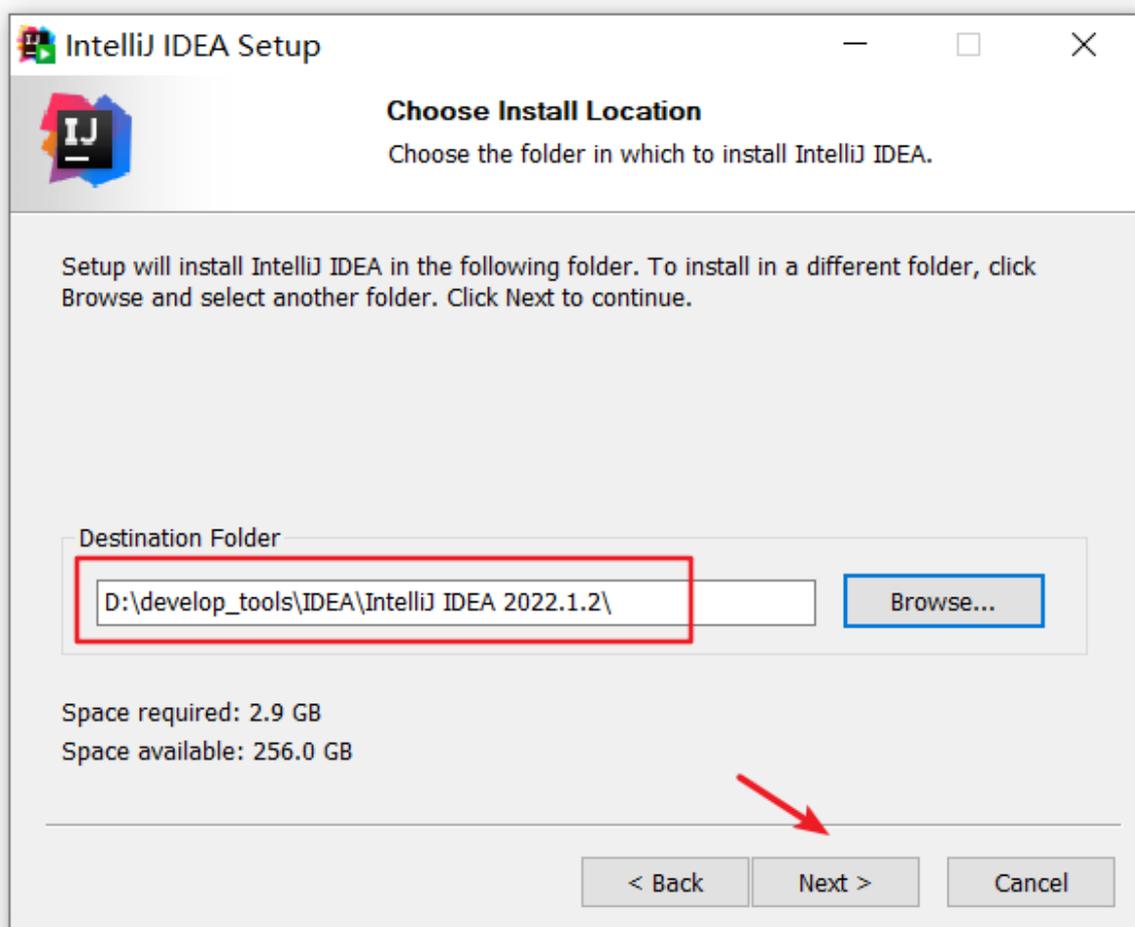


- 如果电脑上有低版本的IDEA，可以选择删除或保留。
- 这里没有卸载旧版本，如果需要卸载，记得勾选下面的保留旧的设置和配置。

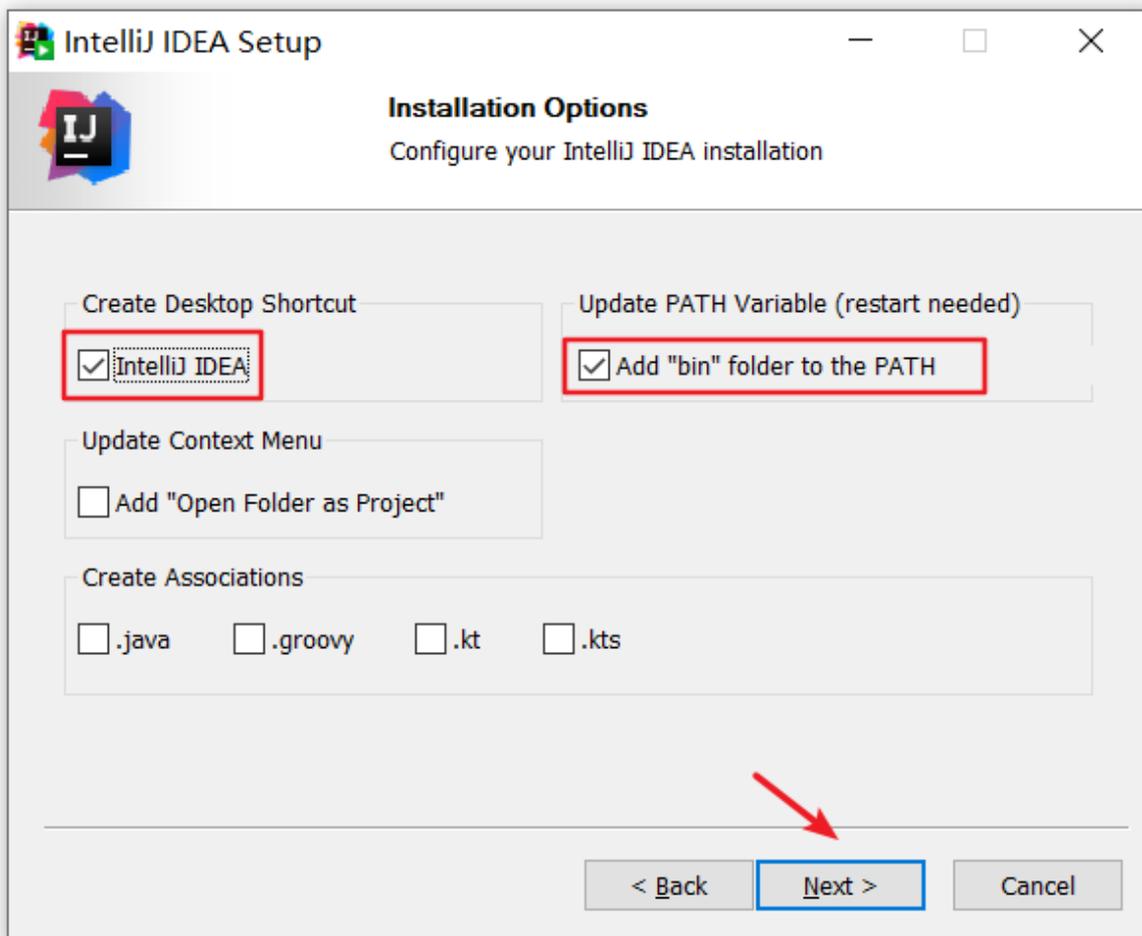
4、选择安装目录



选择安装目录，目录中要避免中文和空格。

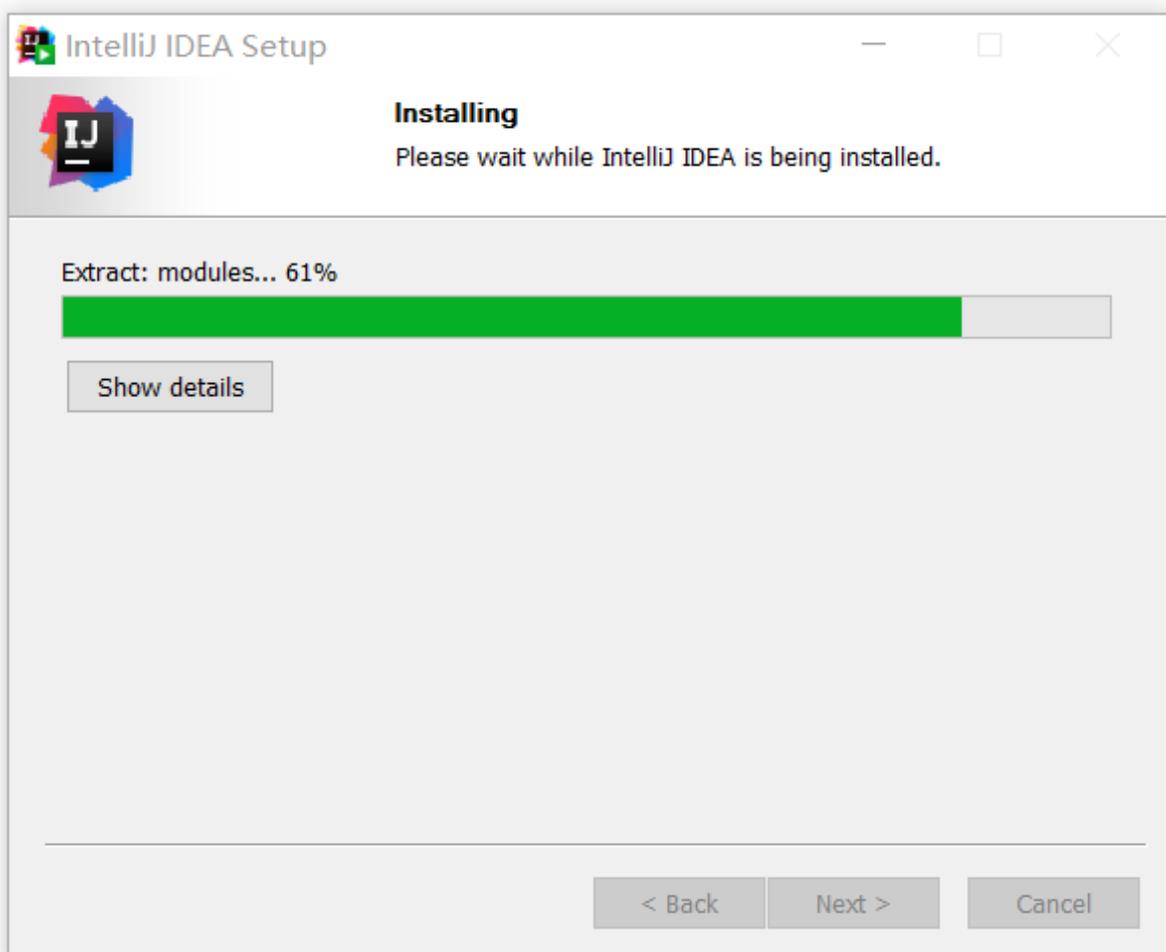
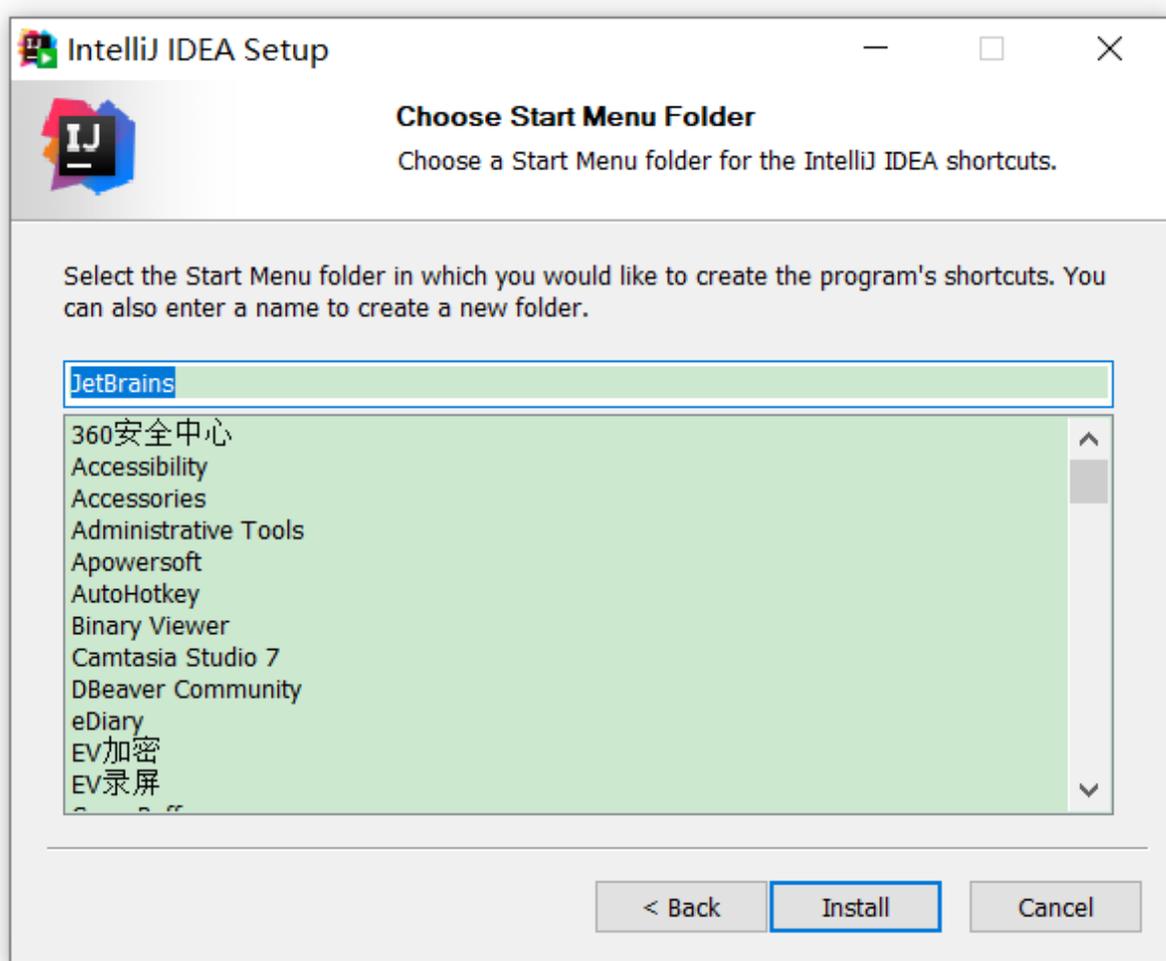


5、创建桌面快捷图标等

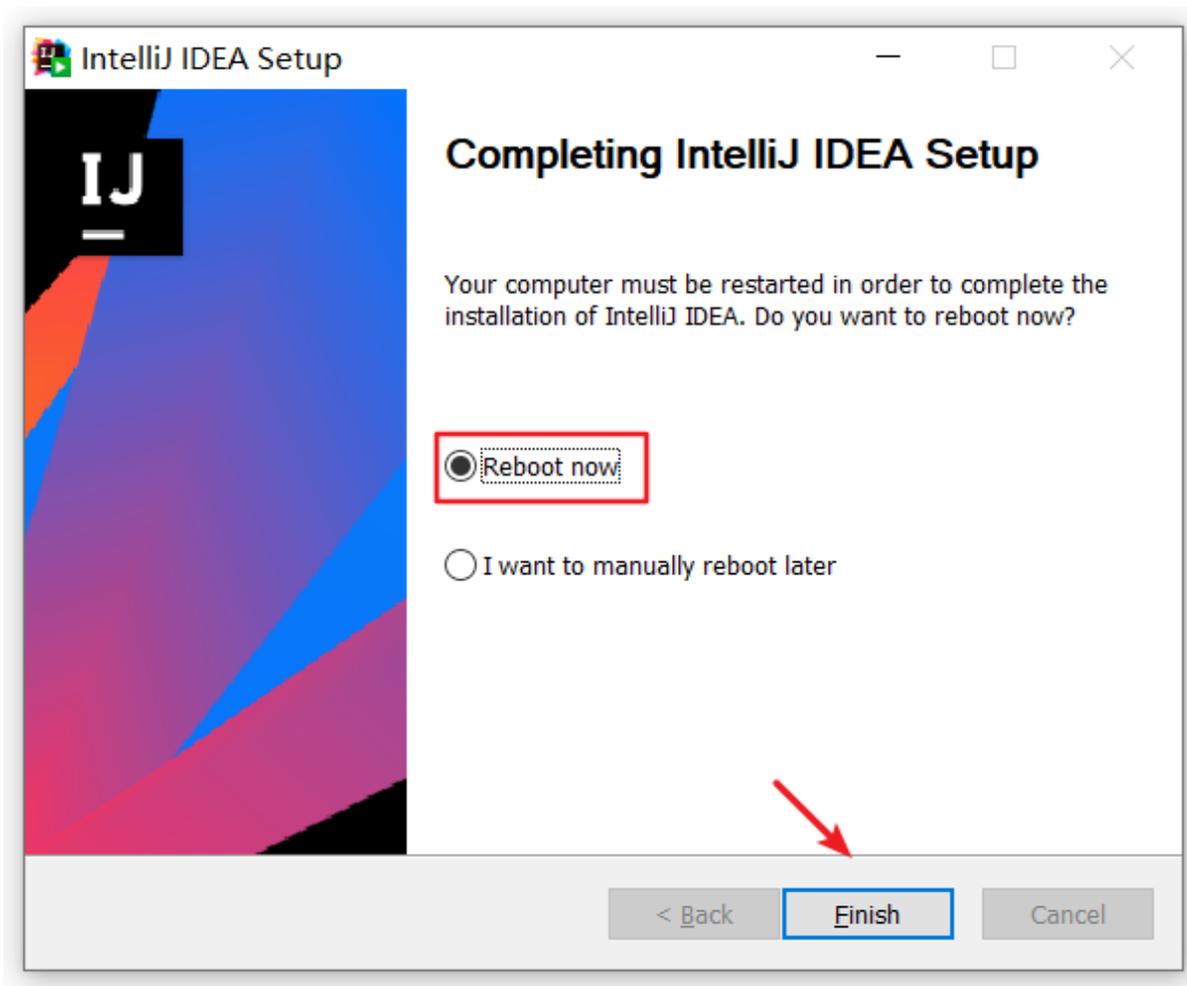


确认是否与.java、.groovy、.kt 格式文件进行关联。这里建议不关联。

6、在【开始】菜单新建一个文件夹（这里需要确认文件夹的名称），来管理IDEA的相关内容。



7、完成安装

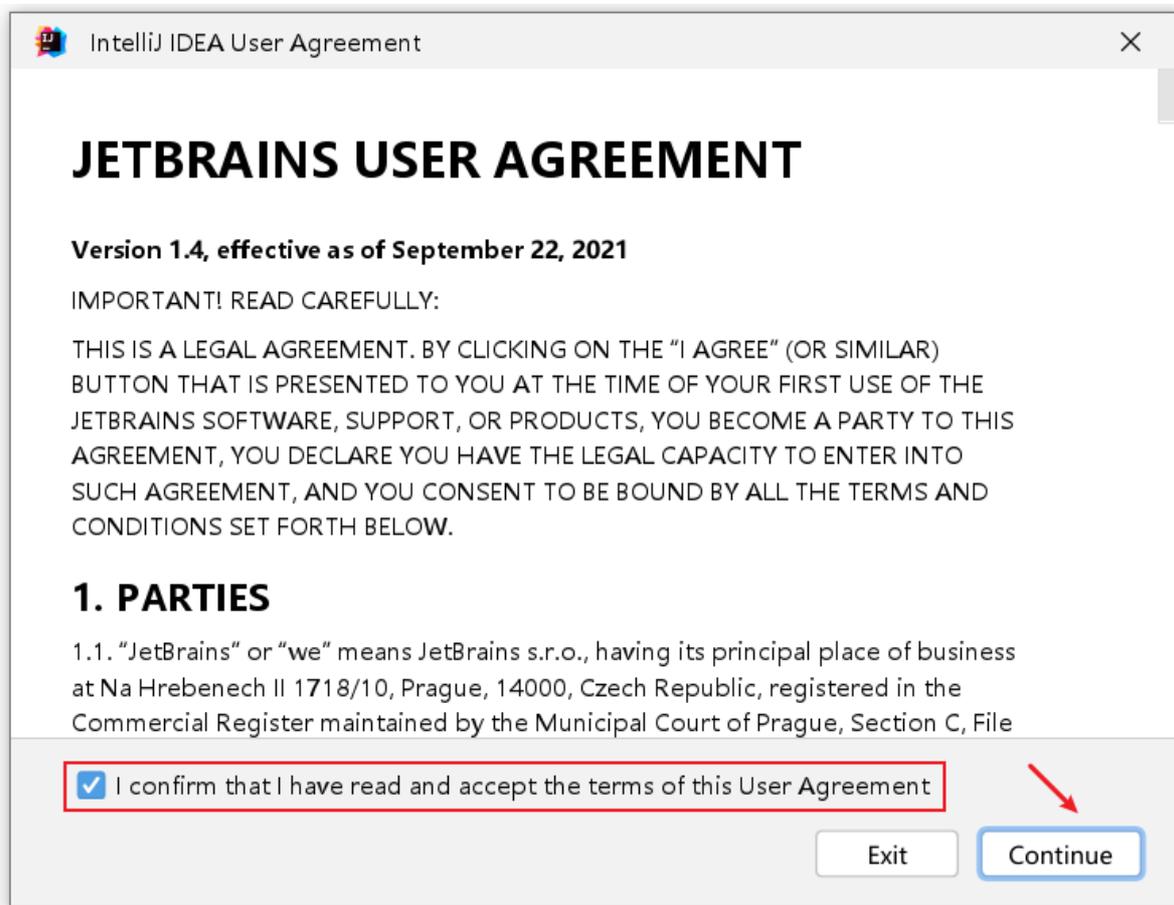


重启以后，单击登录：

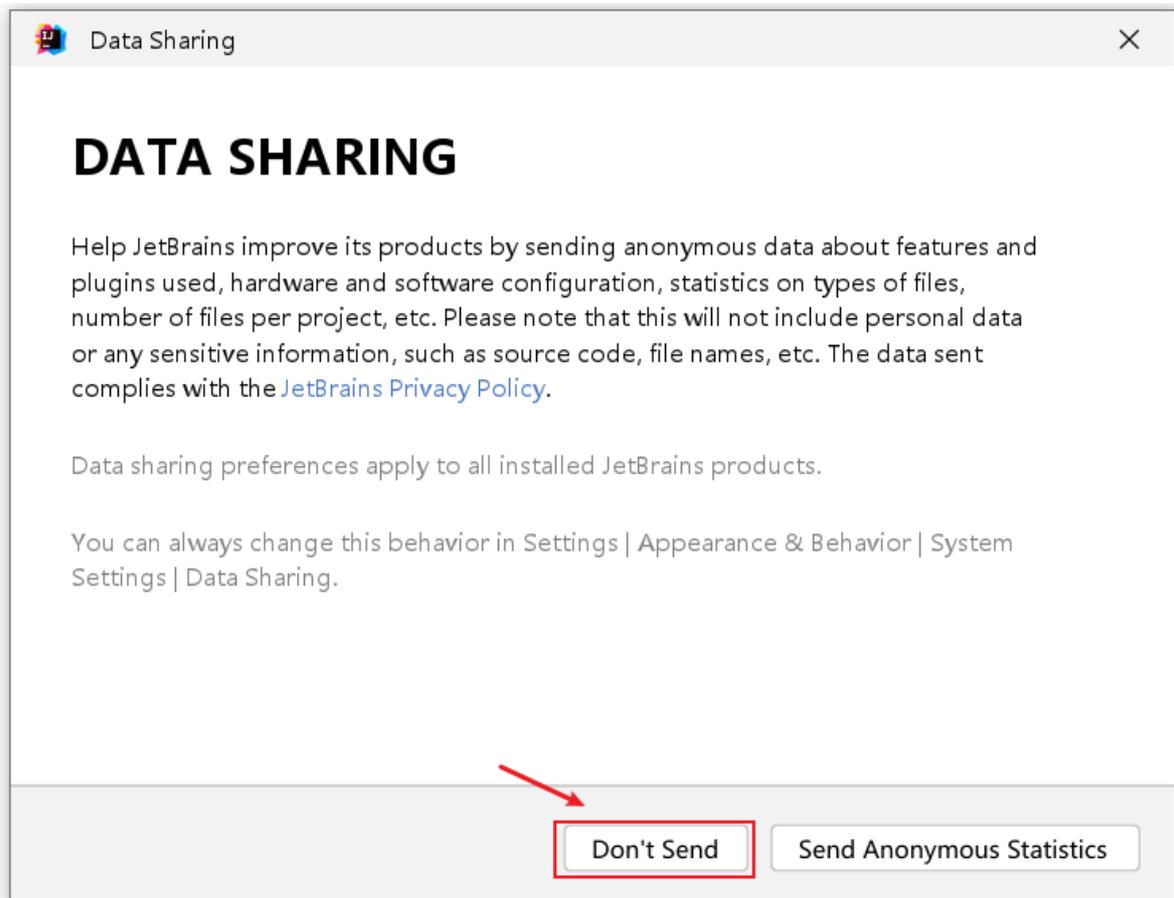


2.4 注册

首先，需要通过用户协议：

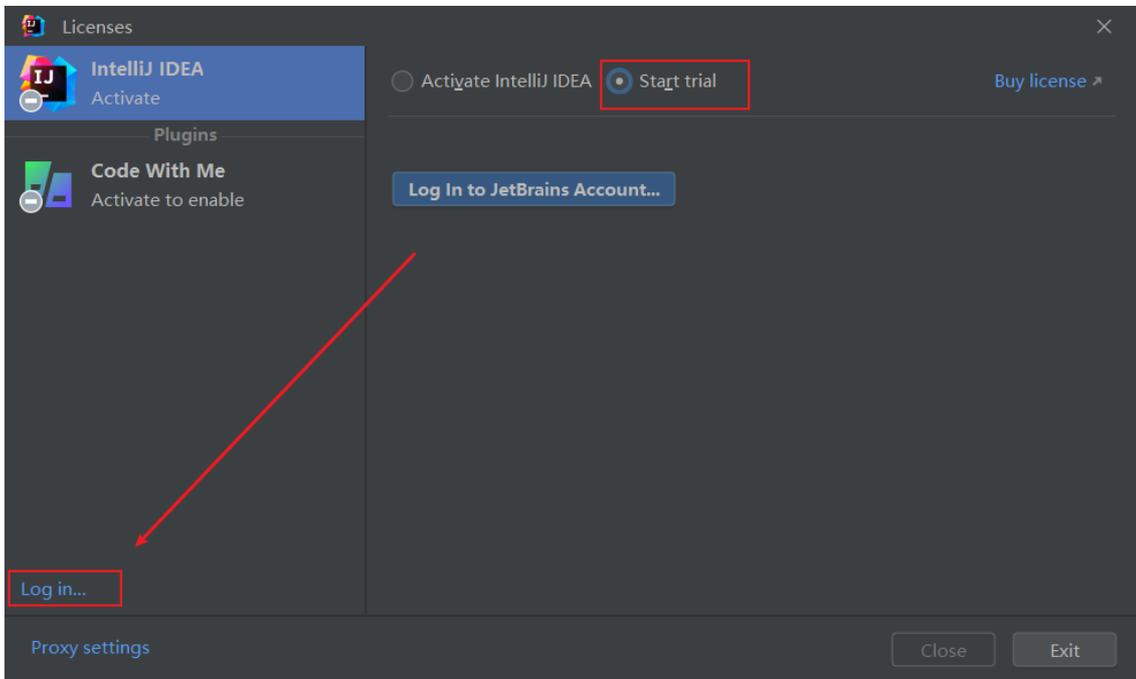


是否同意发送用户数据（特性、使用的插件、硬件与软件配置等），建议选择：不发送。

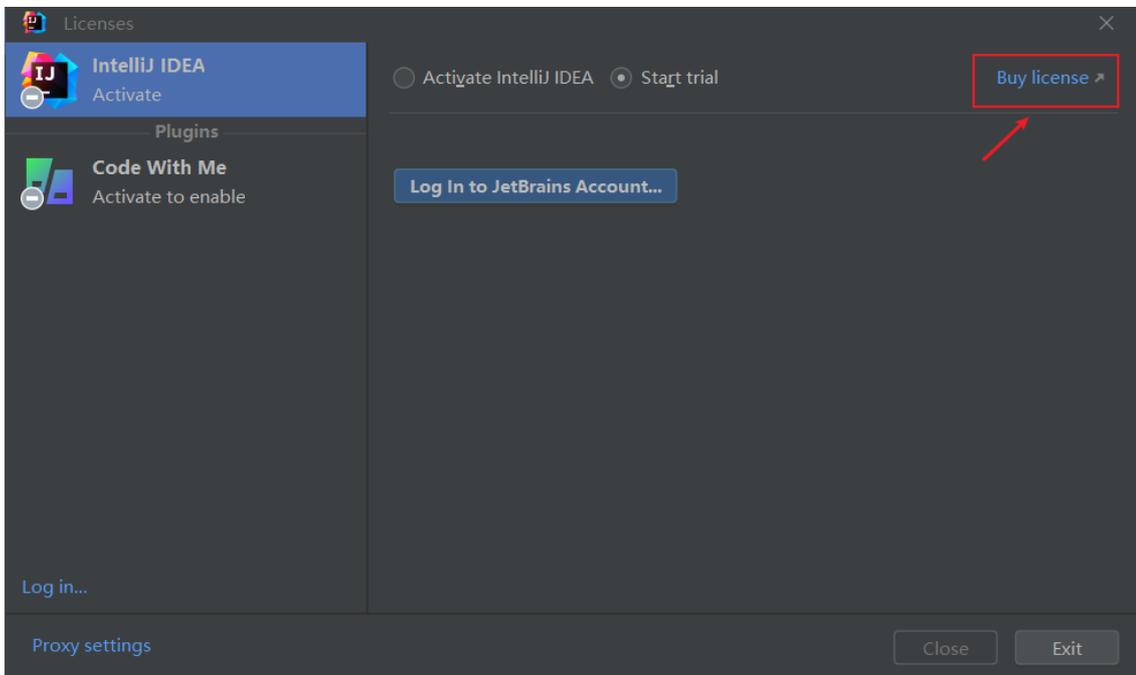


接着，会提示我们进行注册。

- 选择1：适用30天。在IDEA2022.1.2版本中，需要先登录，才能开启适用。



- 选择2: 付费购买旗舰版



IntelliJ IDEA Ultimate

The most intelligent IDE for Java and Kotlin



per user, first year	US \$599.00
second year	US \$479.00
third year onwards	US \$359.00

[Get quote](#) [Buy](#)

- 选择3: (推荐)

- 大家参照《[.\03-资料\01-IDEA 2022.1 注册文档\IntelliJ IDEA2022.1注册文档.docx](#)》操作即可。
- 由于存在时效性，如果失效，大家可以自行搜索注册方式即可。

2.5 闪退问题

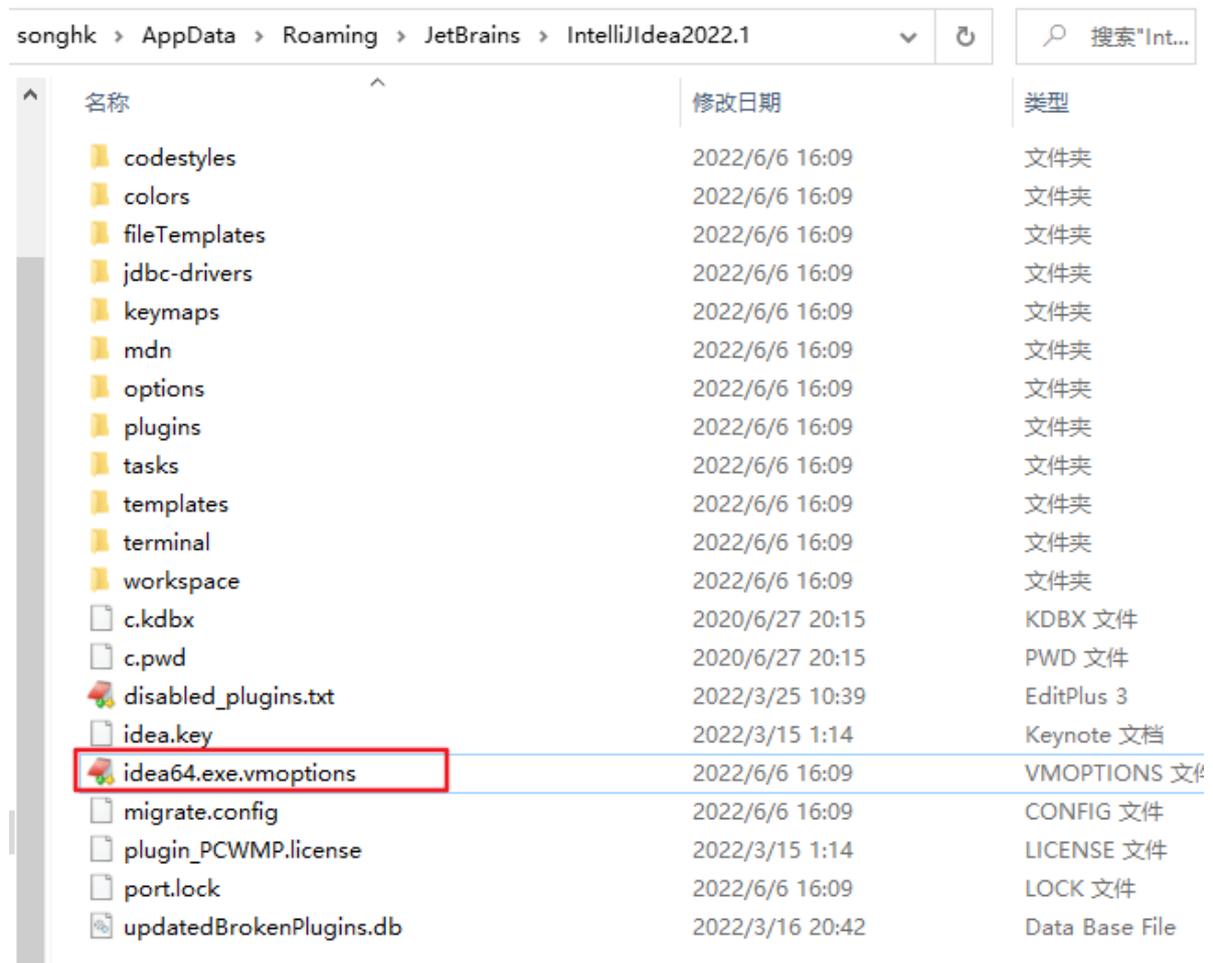
问题描述1：2022.1启动不了，双击桌面图标，没有响应。

问题描述2：进入到安装目录...\IntelliJ IDEA 2022.1.2\bin，打开CMD。输入idea，发现报错。

解决办法：

打开

[C:\Users\songhk\AppData\Roaming\JetBrains\IntelliJIdea2022.1\idea64.exe.vmoptions](#) 这个文件。



名称	修改日期	类型
codestyles	2022/6/6 16:09	文件夹
colors	2022/6/6 16:09	文件夹
fileTemplates	2022/6/6 16:09	文件夹
jdbc-drivers	2022/6/6 16:09	文件夹
keymaps	2022/6/6 16:09	文件夹
mdn	2022/6/6 16:09	文件夹
options	2022/6/6 16:09	文件夹
plugins	2022/6/6 16:09	文件夹
tasks	2022/6/6 16:09	文件夹
templates	2022/6/6 16:09	文件夹
terminal	2022/6/6 16:09	文件夹
workspace	2022/6/6 16:09	文件夹
c.kdbx	2020/6/27 20:15	KDBX 文件
c.pwd	2020/6/27 20:15	PWD 文件
disabled_plugins.txt	2022/3/25 10:39	EditPlus 3
idea.key	2022/3/15 1:14	Keynote 文档
idea64.exe.vmoptions	2022/6/6 16:09	VMOPTIONS 文件
migrate.config	2022/6/6 16:09	CONFIG 文件
plugin_PCWMP.license	2022/3/15 1:14	LICENSE 文件
port.lock	2022/6/6 16:09	LOCK 文件
updatedBrokenPlugins.db	2022/3/16 20:42	Data Base File

内容如下所示：

```
-Xmx2048m
-Djava.net.preferIPv4Stack=true
-Djdk.attach.allowAttachSelf
-Deditable.java.test.console=true
-javaagent:D:\develop_tools\IDEA\IntelliJ IDEA 2021.2.2\bin\filter-agent.jar
```

删除红框的数据以后，再登录即可正常进入。

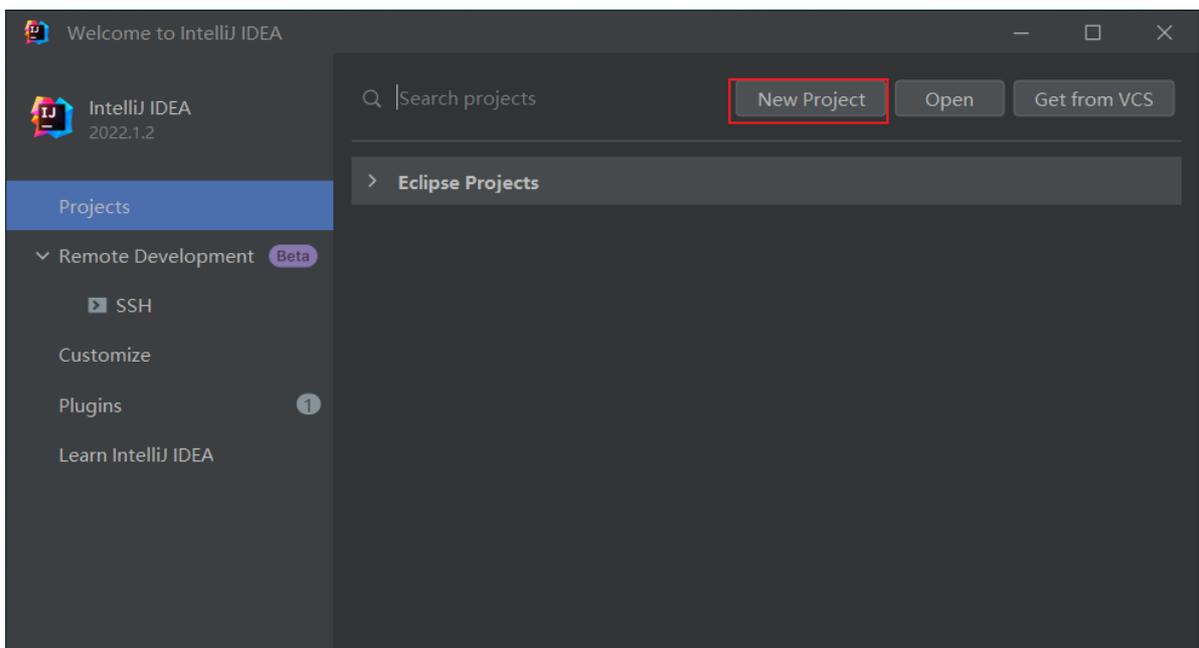


原因：之前使用过的比如2021.2.2版本，pojie了。新版IEDA太智能了，把现有的启运参数也都复制过去了。又因为最新的IDEA，不兼容pojie程序-javaagent:D:\develop_tools\IDEA\IntelliJ IDEA 2021.2.2\bin\jetbrains-agent.jar了，所以报错了，所以JVM结束了，所以没有启动画面，凉凉了。

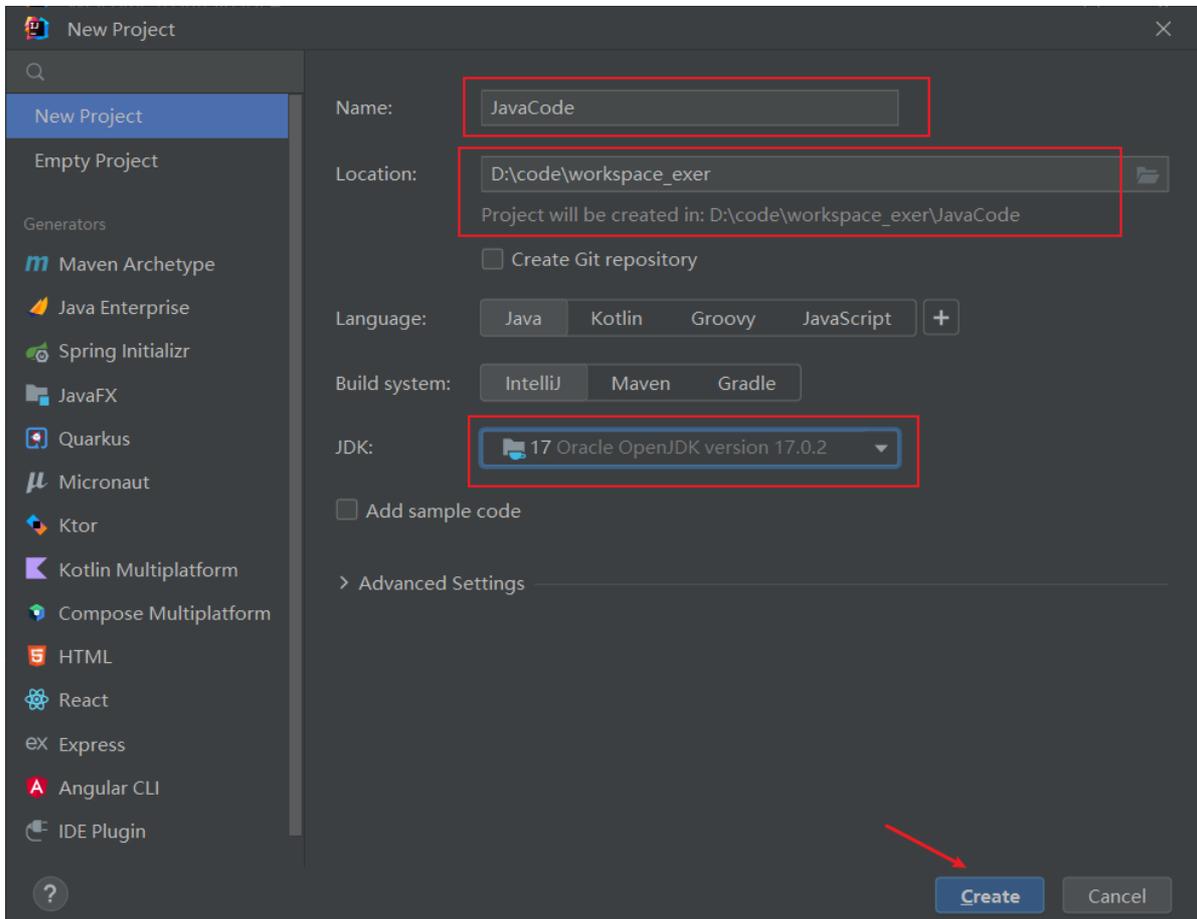
3. HelloWorld的实现

3.1 新建Project - Class

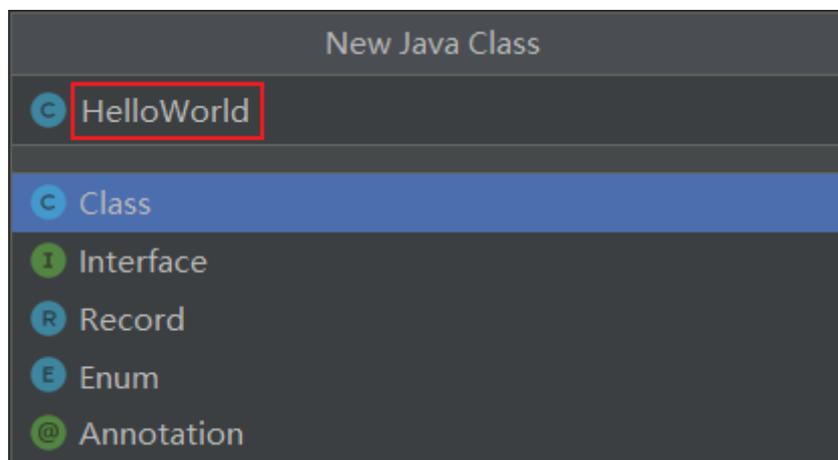
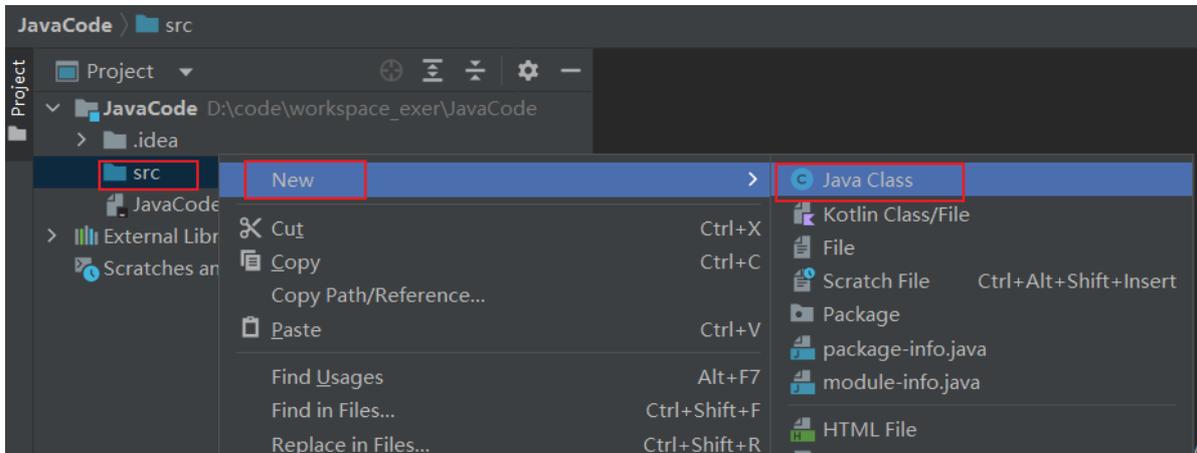
选择"New Project":



指名工程名、使用的JDK版本等信息。如下所示:



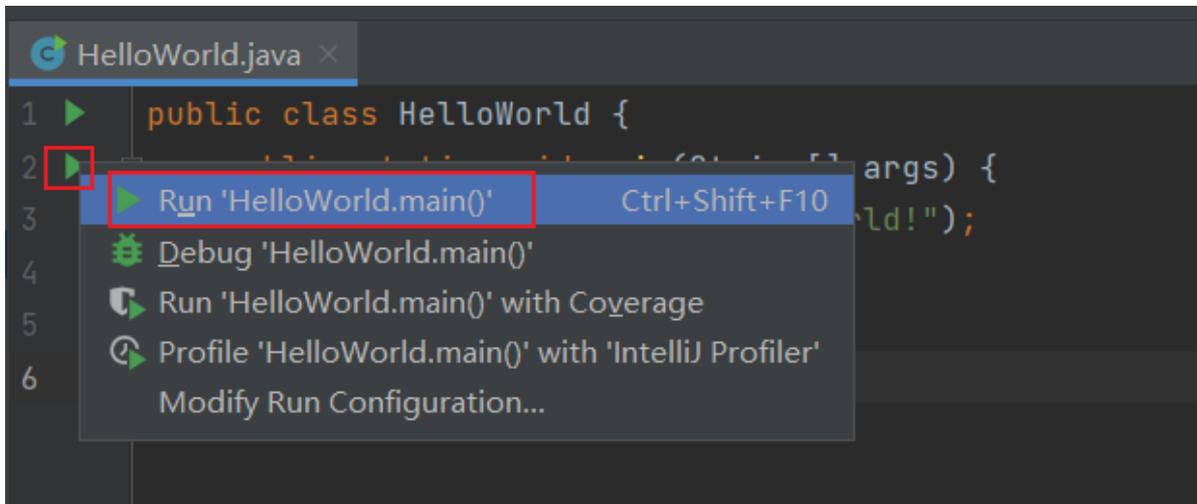
接着创建Java类:



3.2 编写代码

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello,World!");  
    }  
}
```

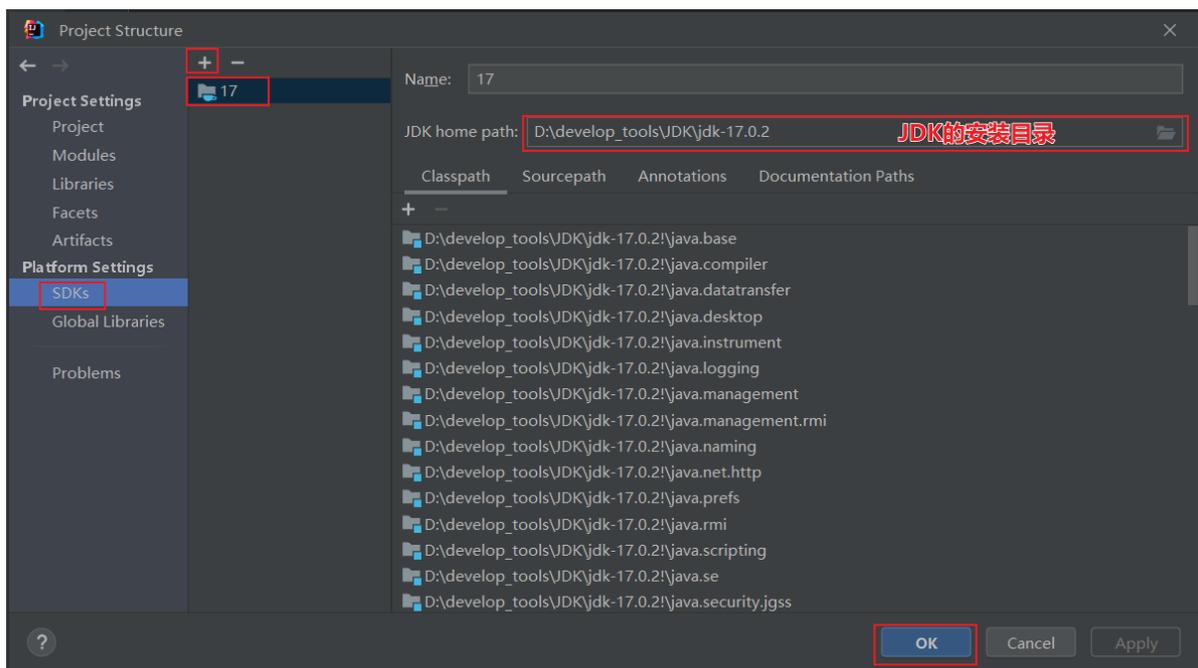
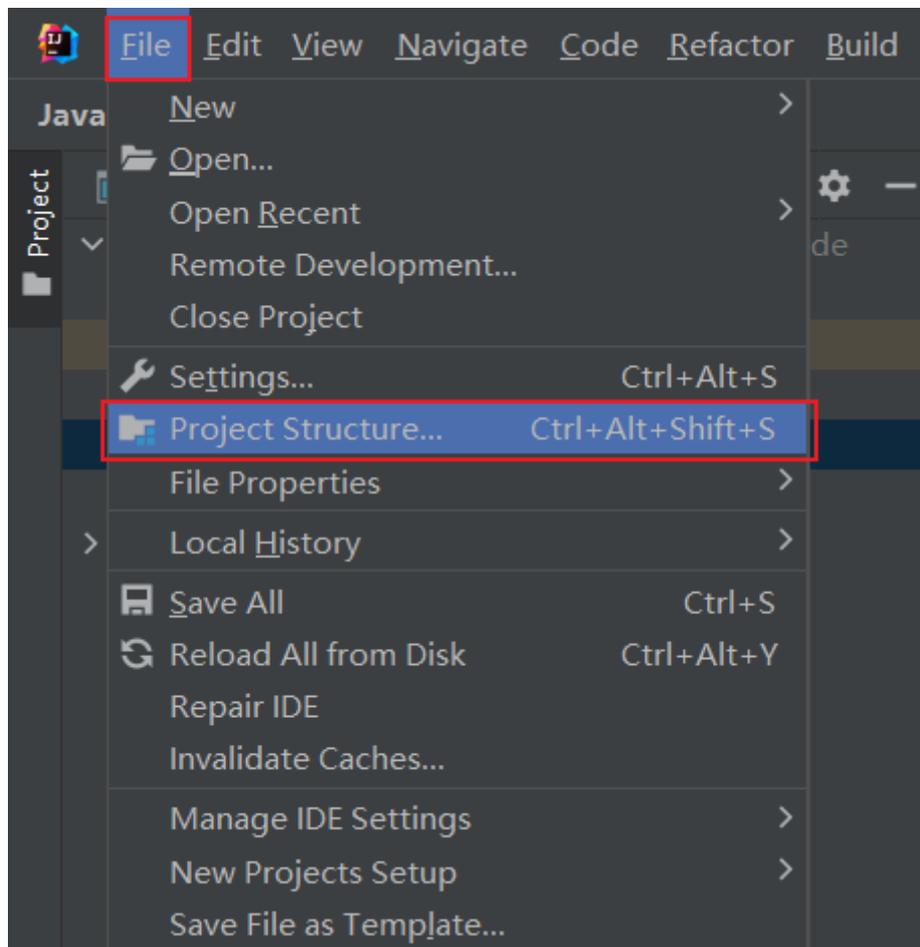
3.3 运行



4. JDK相关设置

4.1 项目的JDK设置

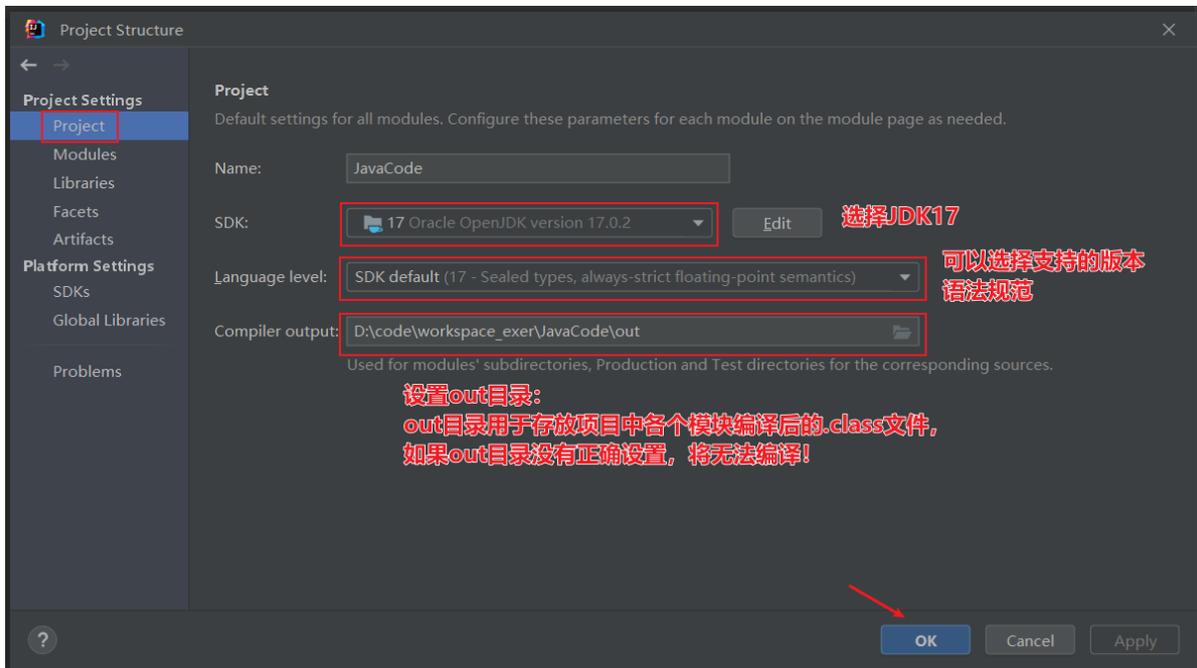
File-->Project Structure...-->Platform Settings -->SDKs



- 注1: SDKs全称是Software Development Kit，这里一定是选择JDK的安装根目录，不是JRE的目录。
- 注2: 这里还可以从本地添加多个JDK。使用“+”即可实现。

4.2 out目录和编译版本

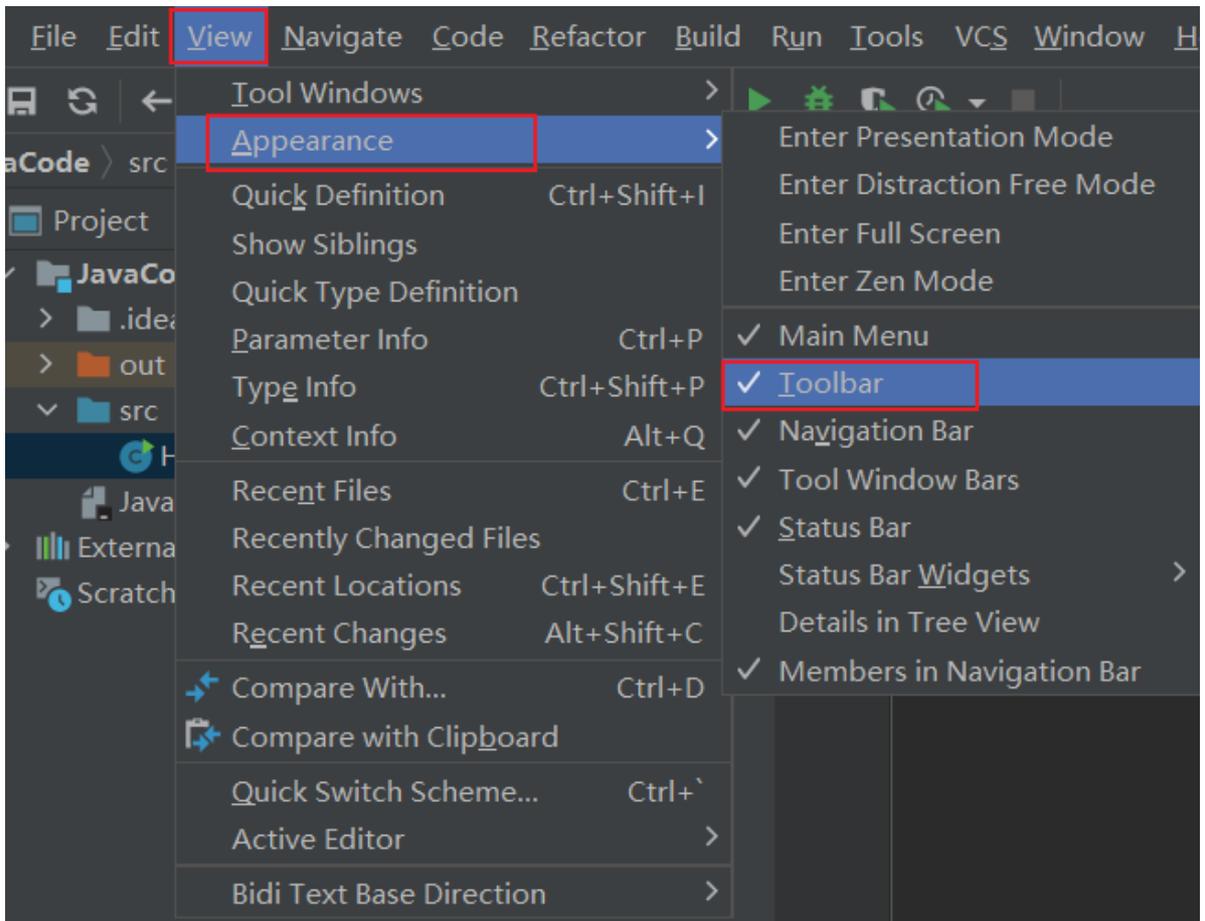
File-->Project Structure...-->Project Settings -->Project



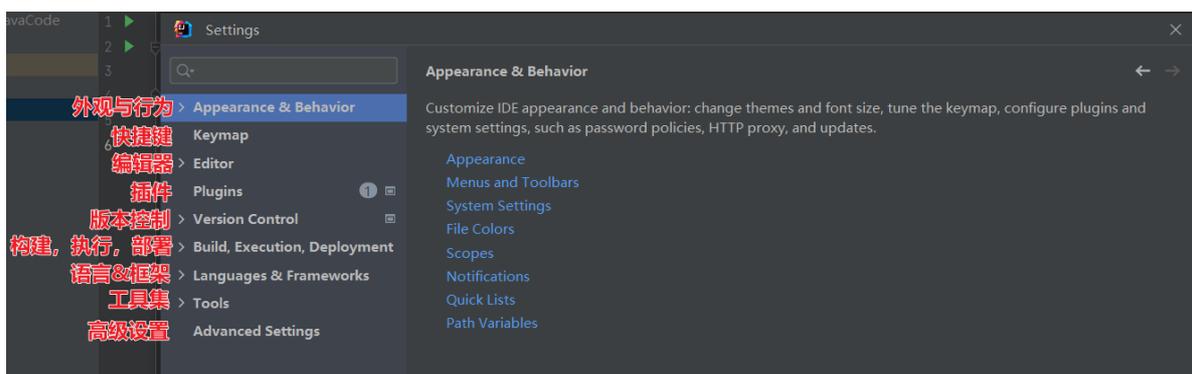
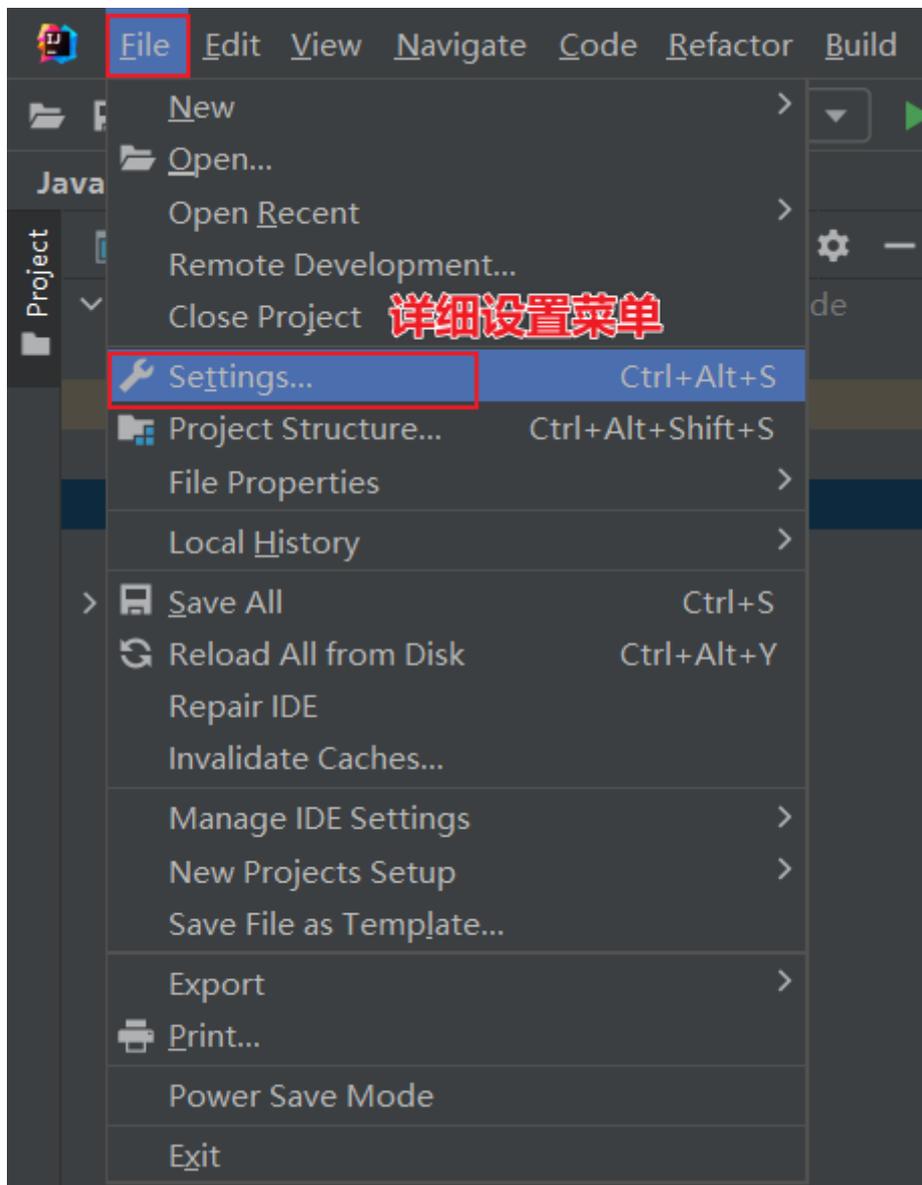
5. 详细设置

5.1 如何打开详细配置界面

1、显示工具栏

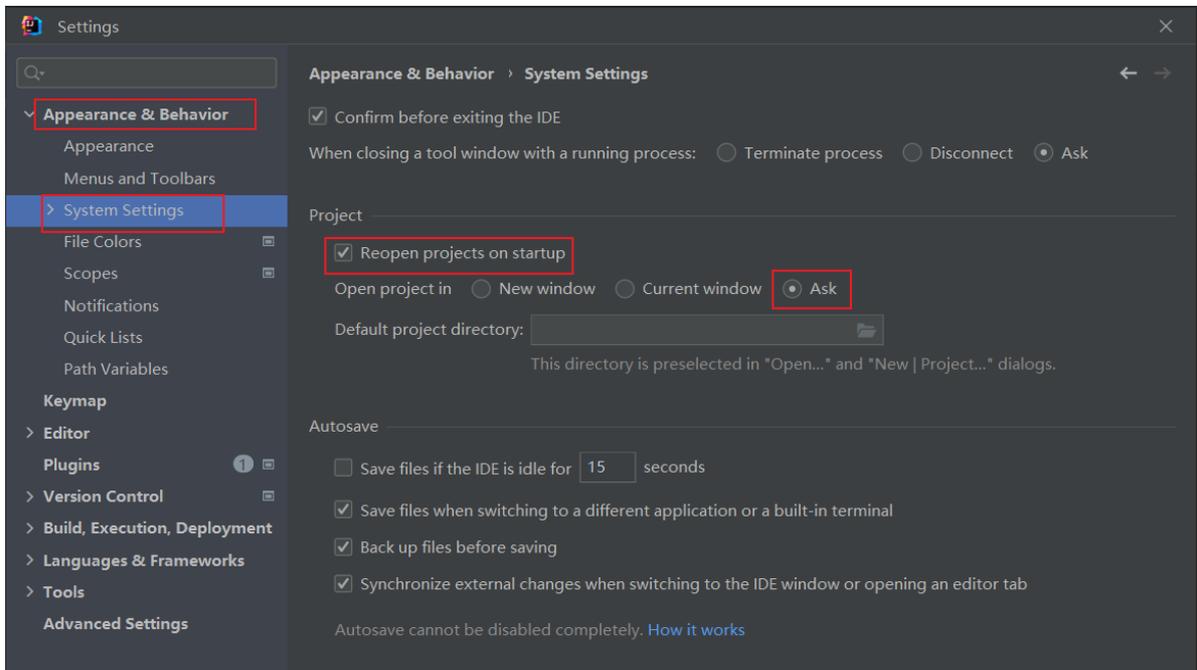


2、选择详细配置菜单或按钮



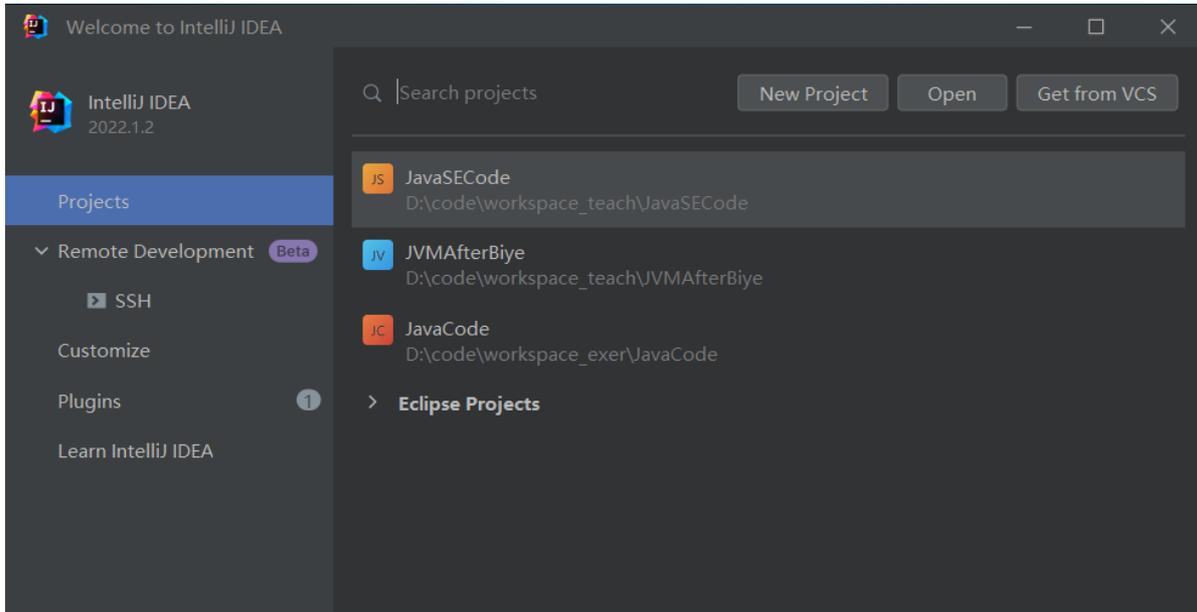
5.2 系统设置

1、默认启动项目配置



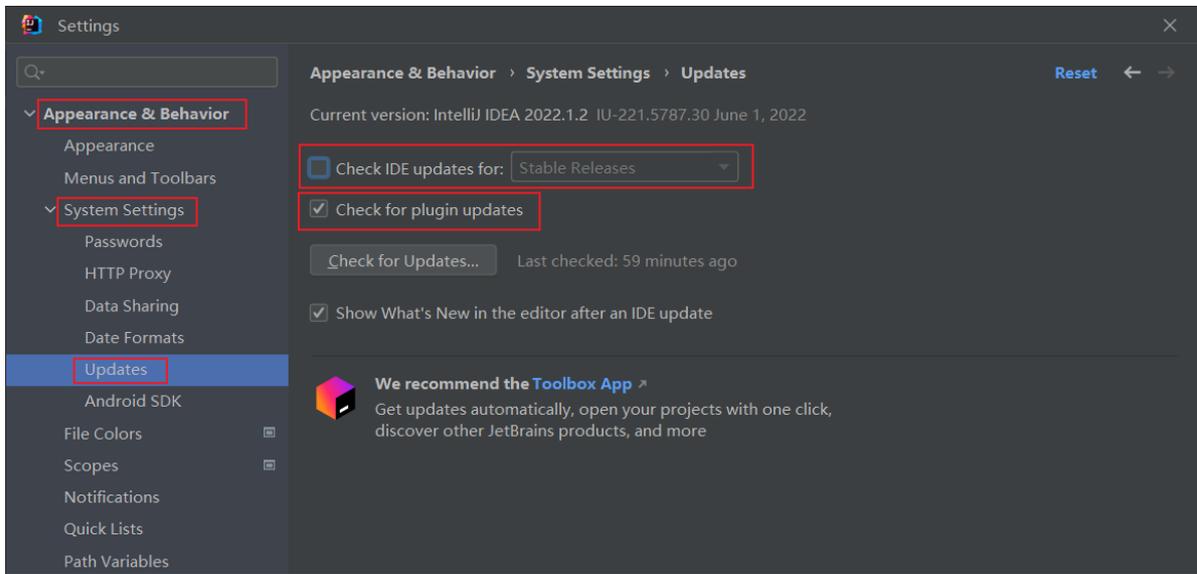
启动IDEA时，默认自动打开上次开发的项目？还是自己选择？

如果去掉Reopen last project on startup前面的对勾，每次启动IDEA就会出现如下界面：



2、取消自动更新

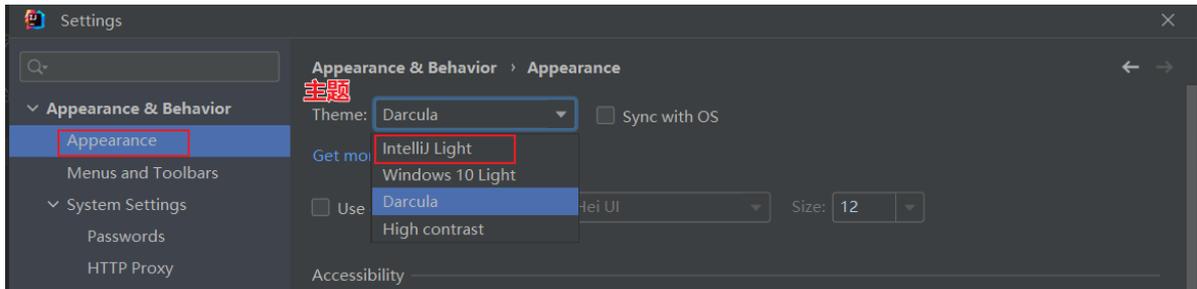
Settings-->Appearance & Behavior->System Settings -> Updates



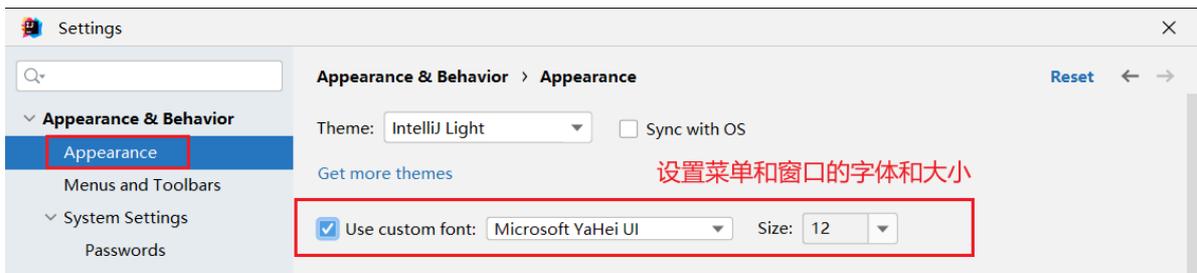
默认都打√了，建议检查IDE更新的√去掉，检查插件更新的√选上。

5.3 设置整体主题

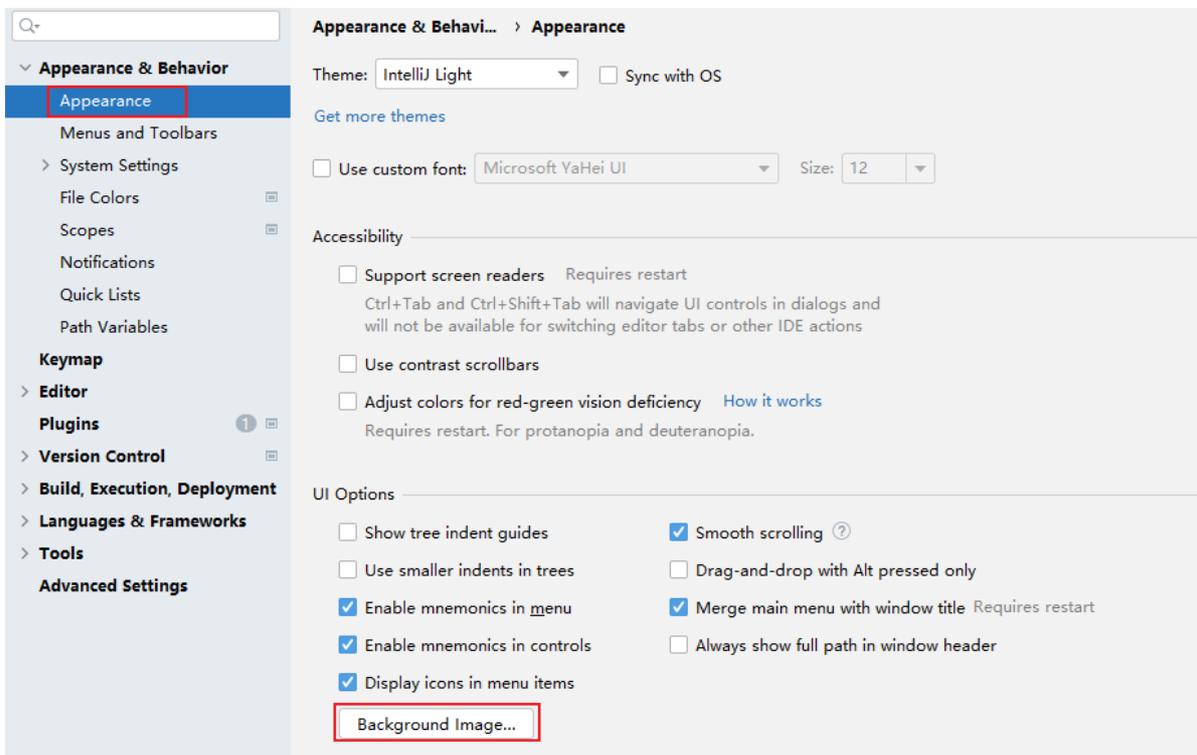
1、选择主题



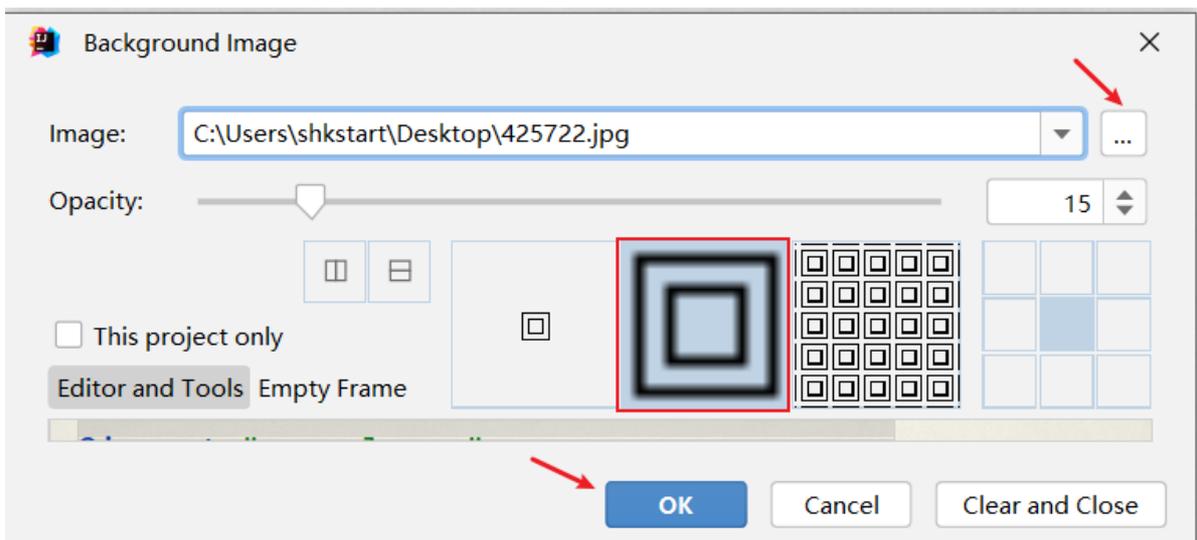
2、设置菜单和窗口字体和大小



3、设置IDEA背景图

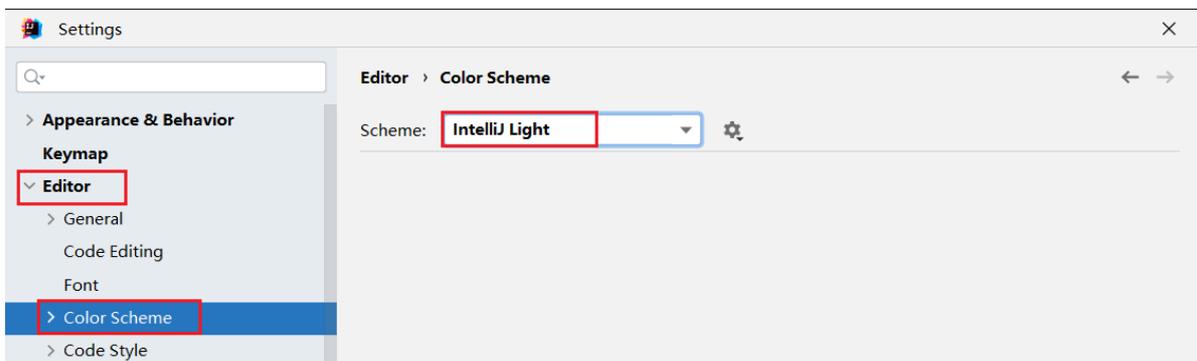


选择一张合适的图片作为背景，即可。

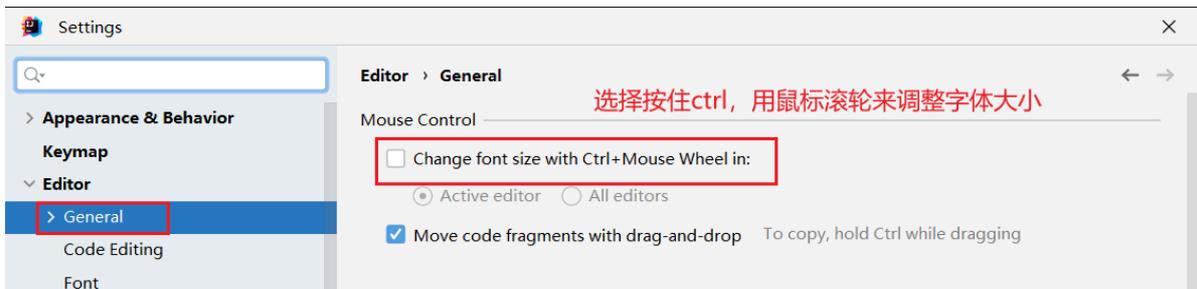


5.4 设置编辑器主题样式

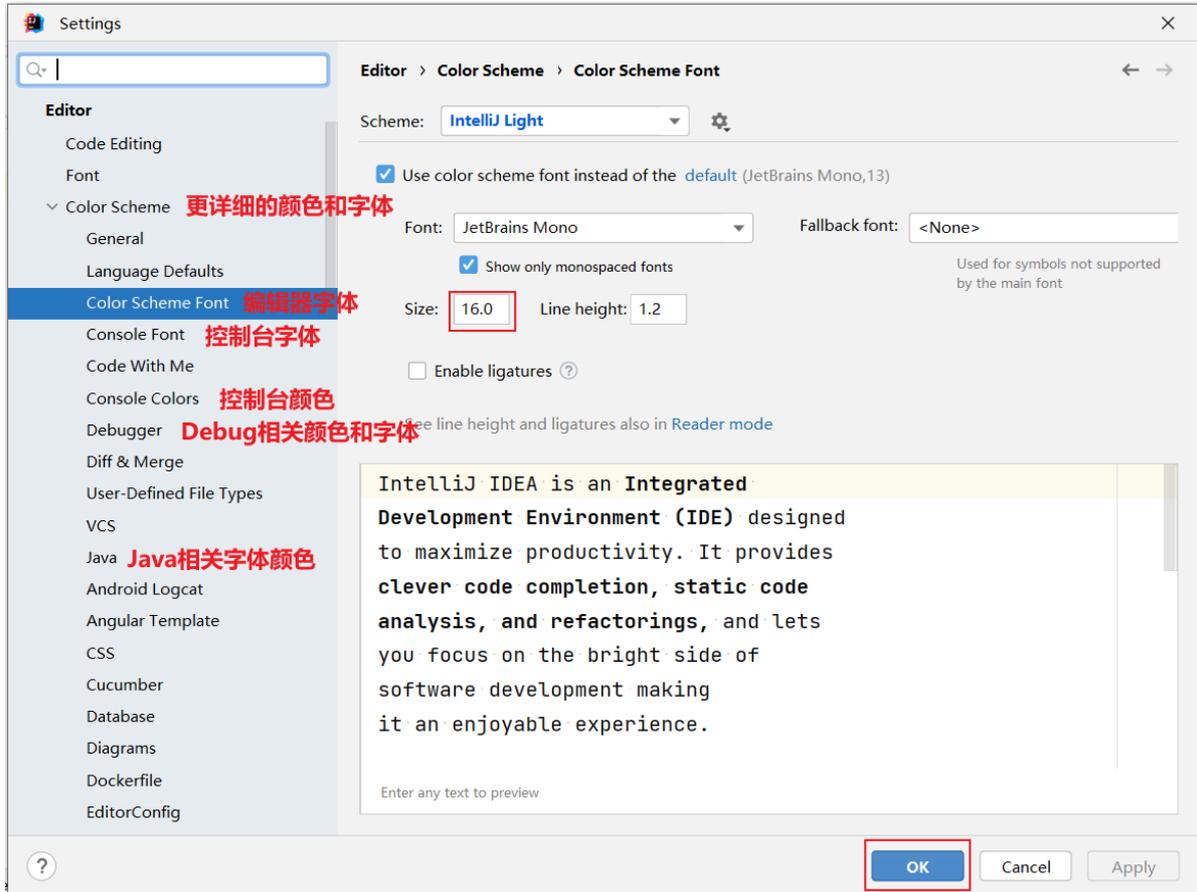
1、编辑器主题



2、字体大小

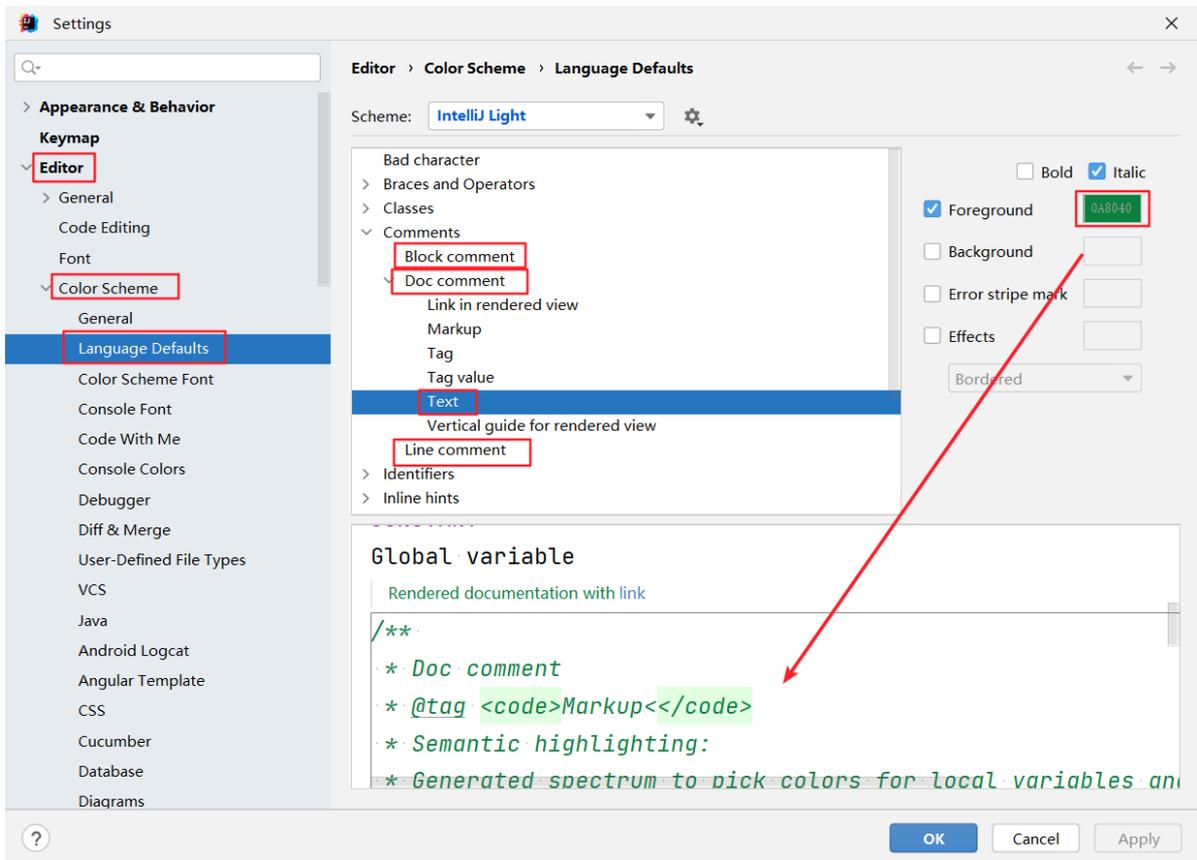


更详细的字体与颜色如下:



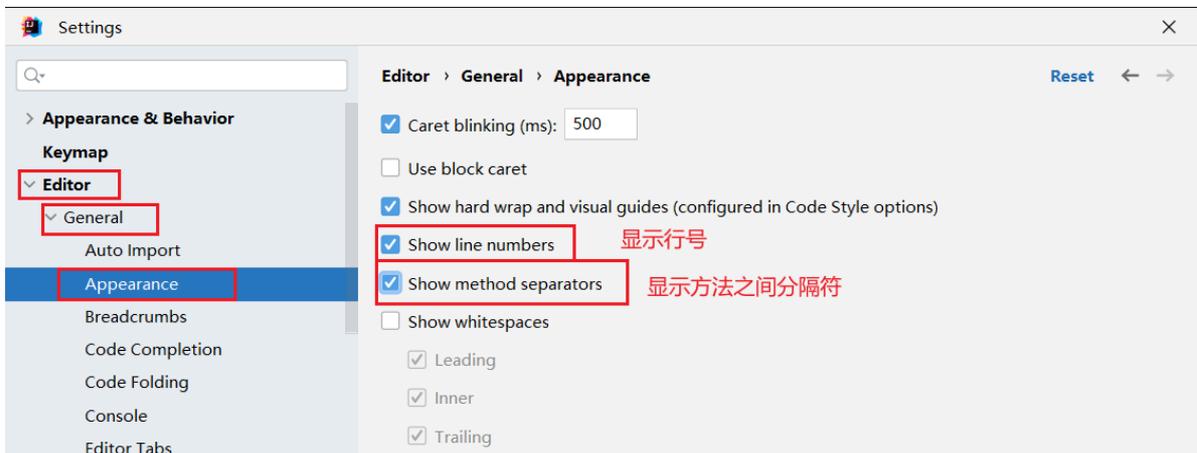
温馨提示: 如果选择某个font字体, 中文乱码, 可以在fallback font (备选字体) 中选择一个支持中文的字体。

3、注释的字体颜色

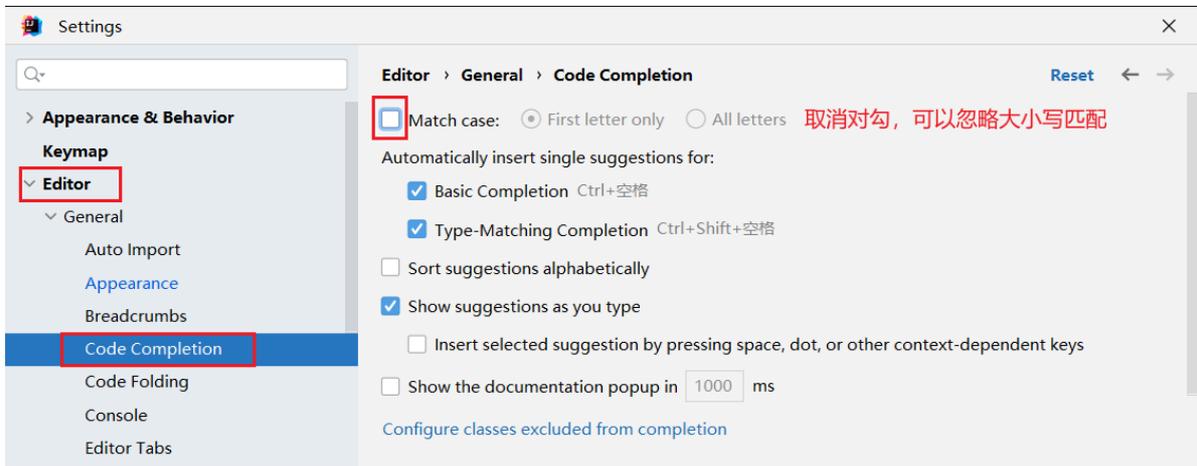


- Block comment: 修改多行注释的字体颜色
- Doc Comment -> Text: 修改文档注释的字体颜色
- Line comment: 修改单行注释的字体颜色

5.5 显示行号与方法分隔符



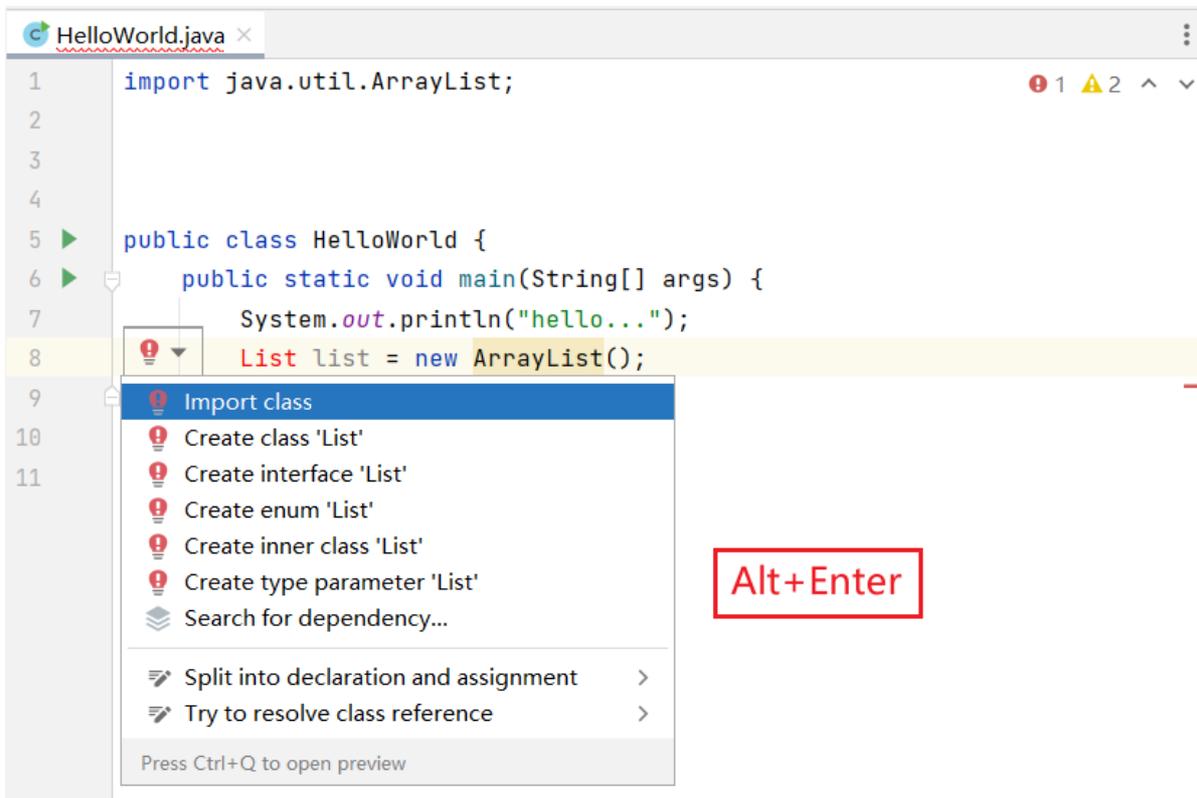
5.6 代码智能提示功能



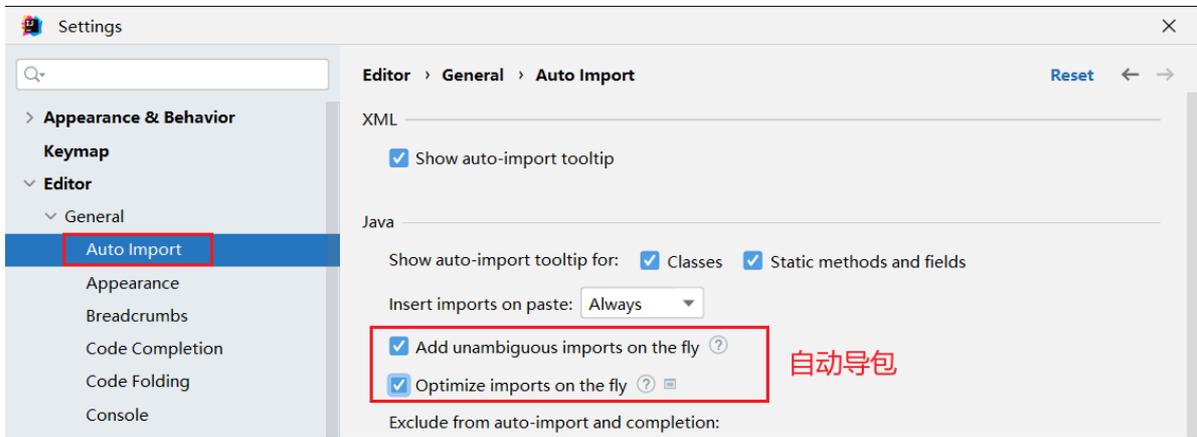
IntelliJ IDEA 的代码提示和补充功能有一个特性：**区分大小写**。如果想不区分大小写的话，就把这个对勾去掉。**建议去掉勾选**。

5.7 自动导包配置

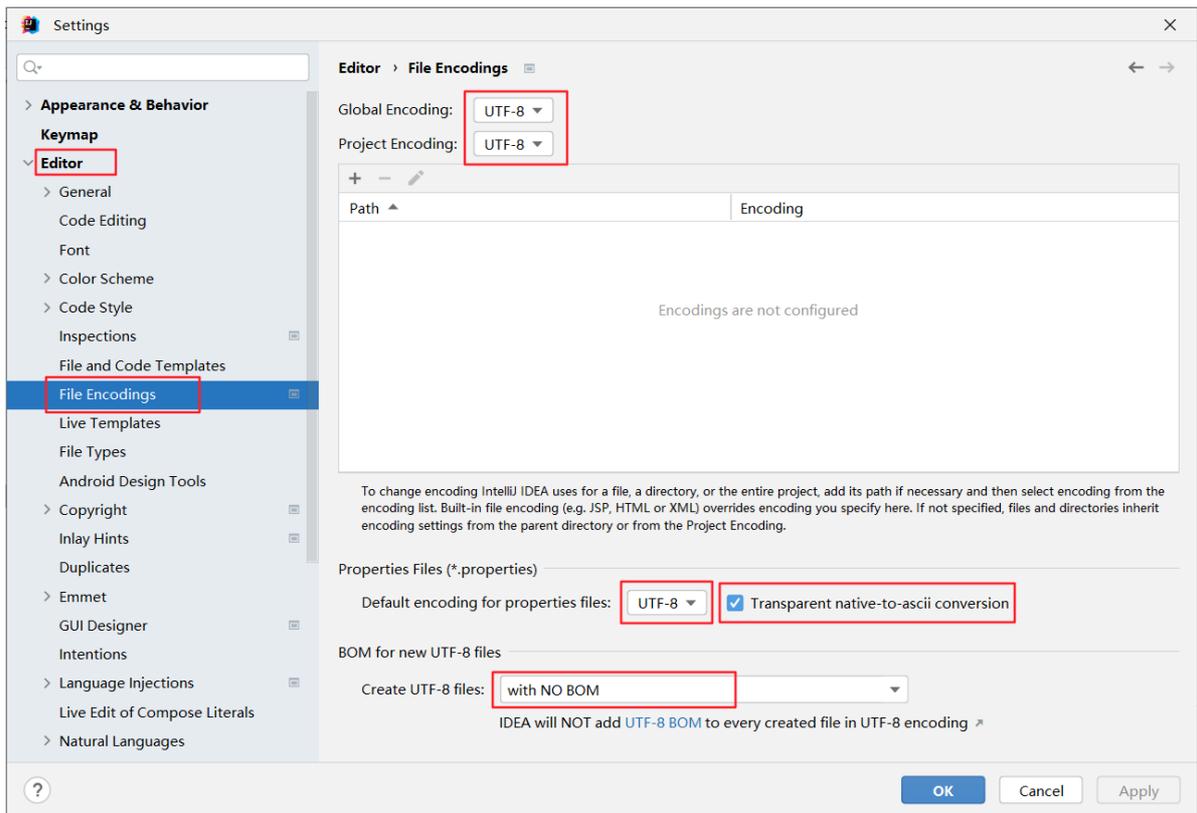
- 默认需要自己手动导包，Alt+Enter快捷键



- 自动导包设置
 - 动态导入明确的包：Add unambiguous imports on the fly，该设置具有全局性；
 - 优化动态导入的包：Optimize imports on the fly，该设置只对当前项目有效；

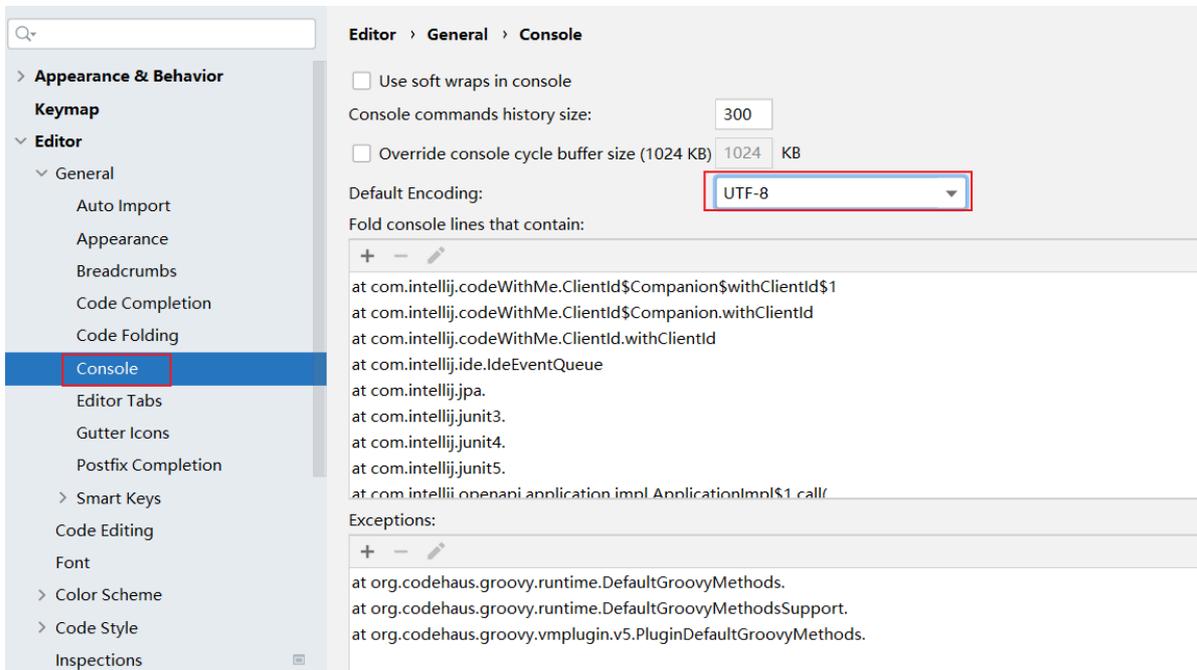


5.8 设置项目文件编码（一定要改）

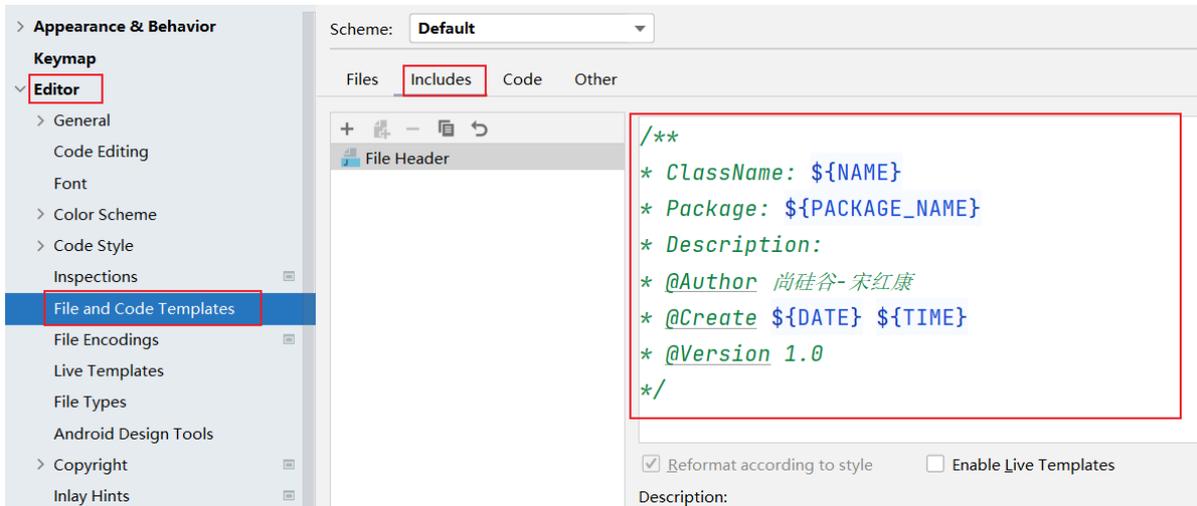


说明：Transparent native-to-ascii conversion主要用于转换ascii，显式原生内容。一般都要勾选。

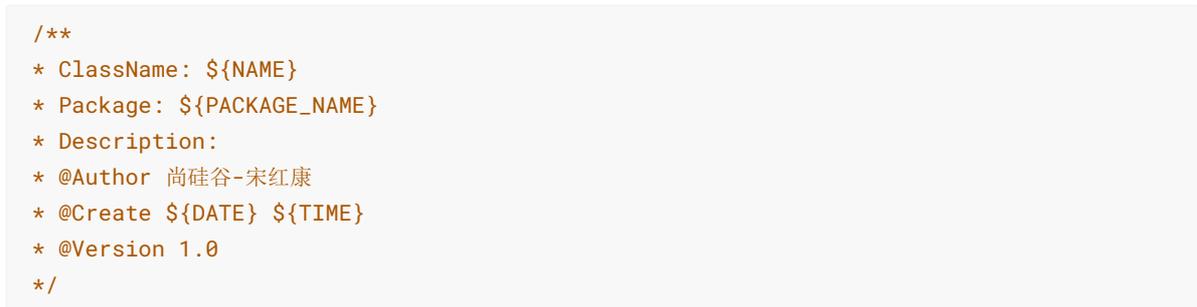
5.9 设置控制台的字符编码



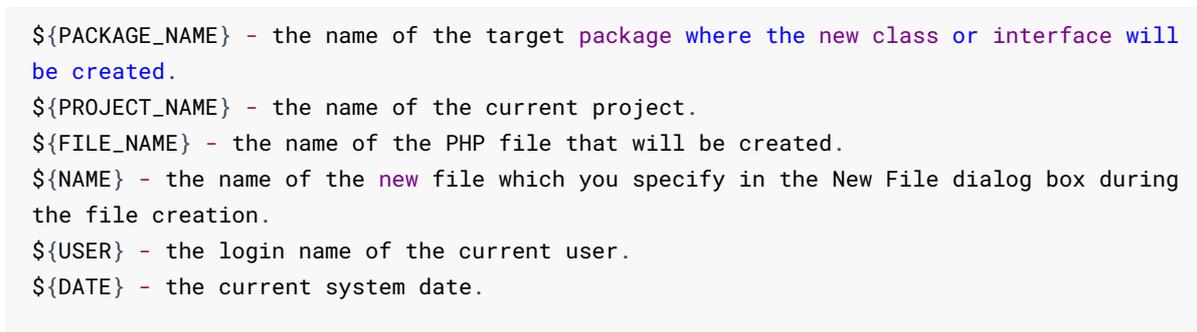
5.10 修改类头的文档注释信息



比如：



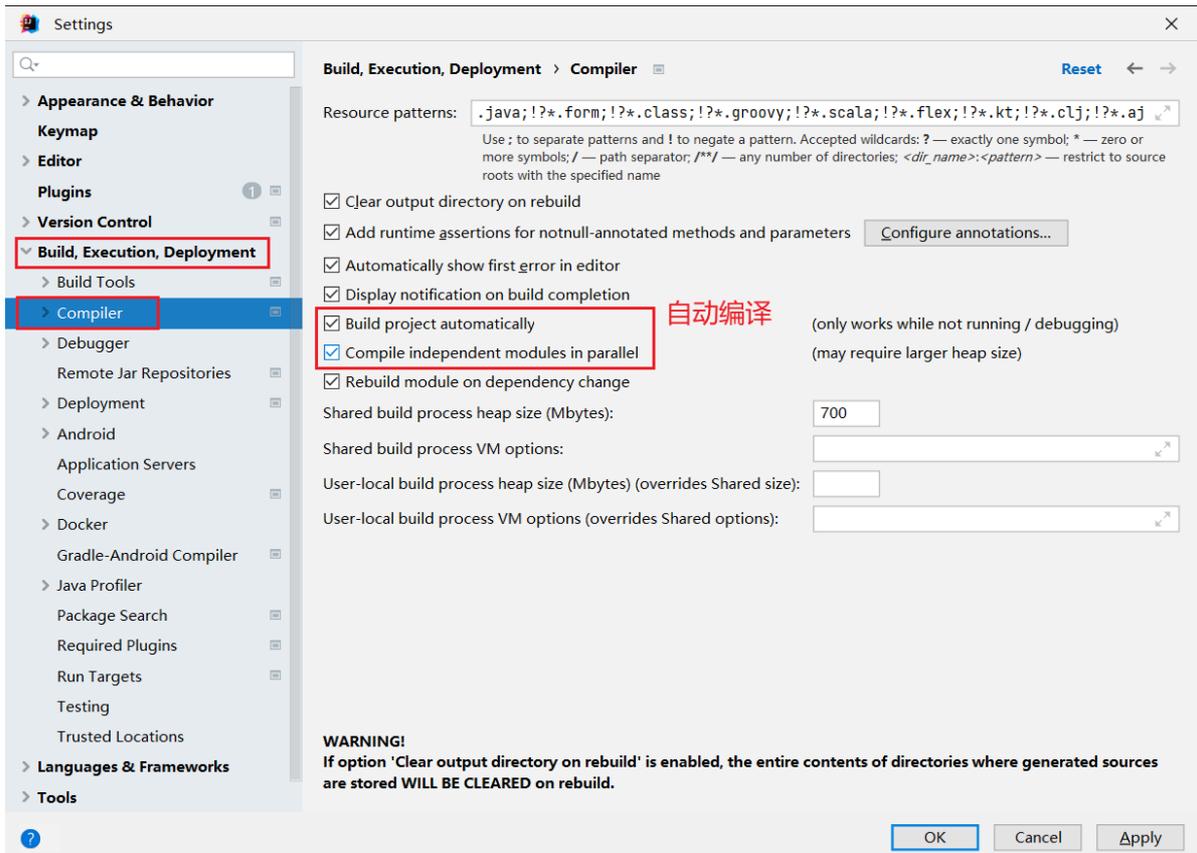
常用的预设的变量，这里直接贴出官网给的：



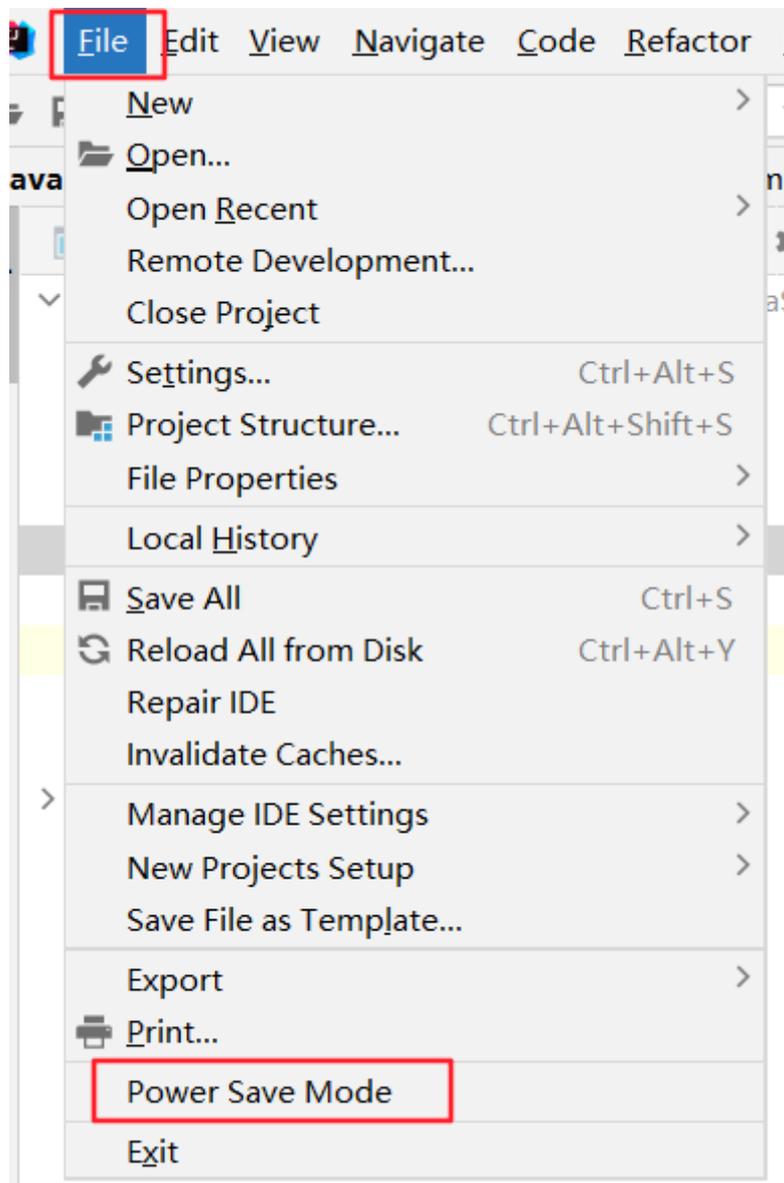
`${TIME}` - the current system time.
`${YEAR}` - the current year.
`${MONTH}` - the current month.
`${DAY}` - the current day of the month.
`${HOUR}` - the current hour.
`${MINUTE}` - the current minute.
`${PRODUCT_NAME}` - the name of the IDE in which the file will be created.
`${MONTH_NAME_SHORT}` - the first 3 letters of the month name. Example: Jan, Feb, etc.
`${MONTH_NAME_FULL}` - full name of a month. Example: January, February, etc.

5.11 设置自动编译

Settings-->Build, Execution, Deployment-->Compiler



5.12 设置为省电模式 (可忽略)



IntelliJ IDEA 有一种叫做 **省电模式** 的状态，开启这种模式之后 IntelliJ IDEA 会 **关掉代码检查** 和 **代码提示** 等功能。所以一般也可认为这是一种 **阅读模式**，如果你在开发过程中遇到突然代码文件不能进行检查和提示，可以来看看这里是否有开启该功能。

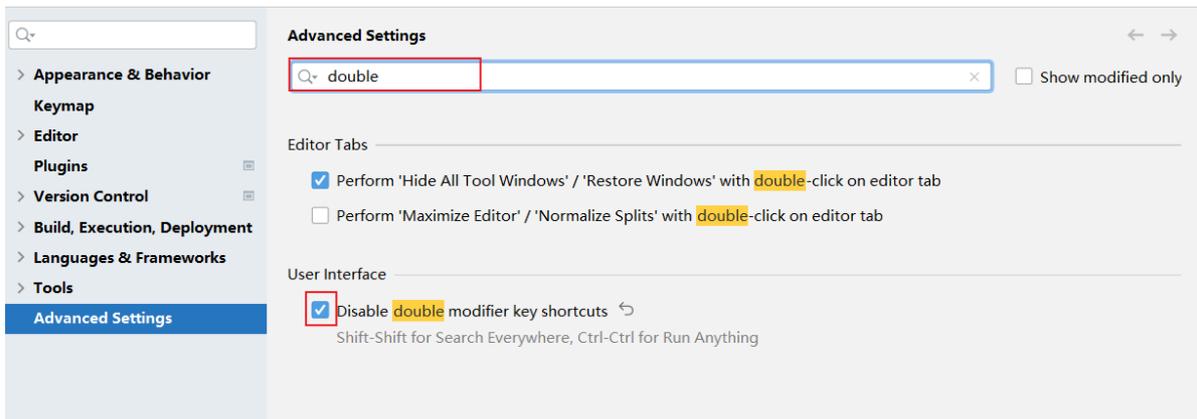
5.13 取消双击shift搜索

因为我们按shift切换中英文输入方式，经常被按到，总是弹出搜索框，太麻烦了。可以取消它。



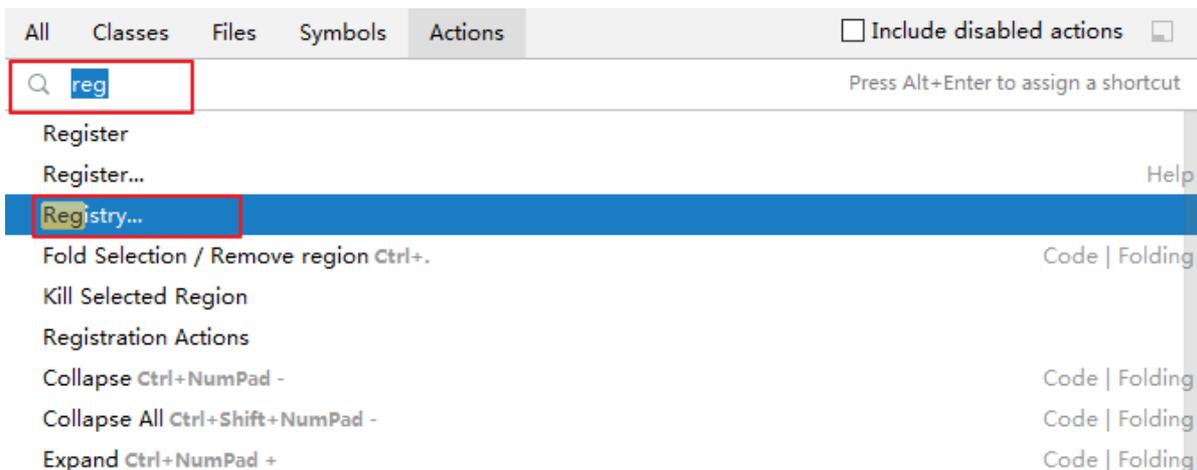
- 方式1: 适用于IDEA 2022.1.2版本

在2022.1版本中, 采用如下方式消除双击shift出现搜索框: 搜索double即可, 勾选Disable double modifier key shortcuts, 禁用这个选项。

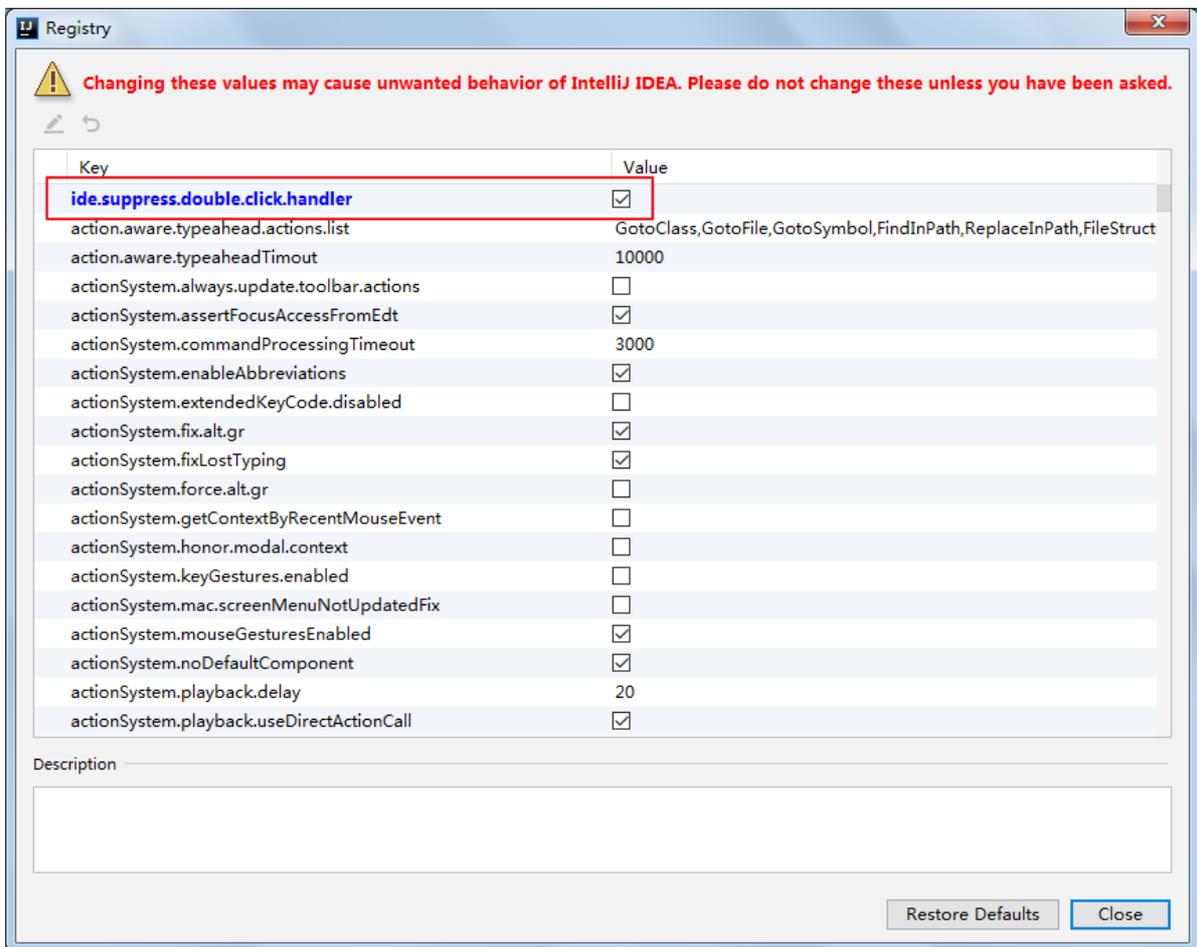


- 方式2: 适用于IDEA 2022.1.2之前版本

双击shift 或 ctrl + shift + a, 打开如下搜索窗口:

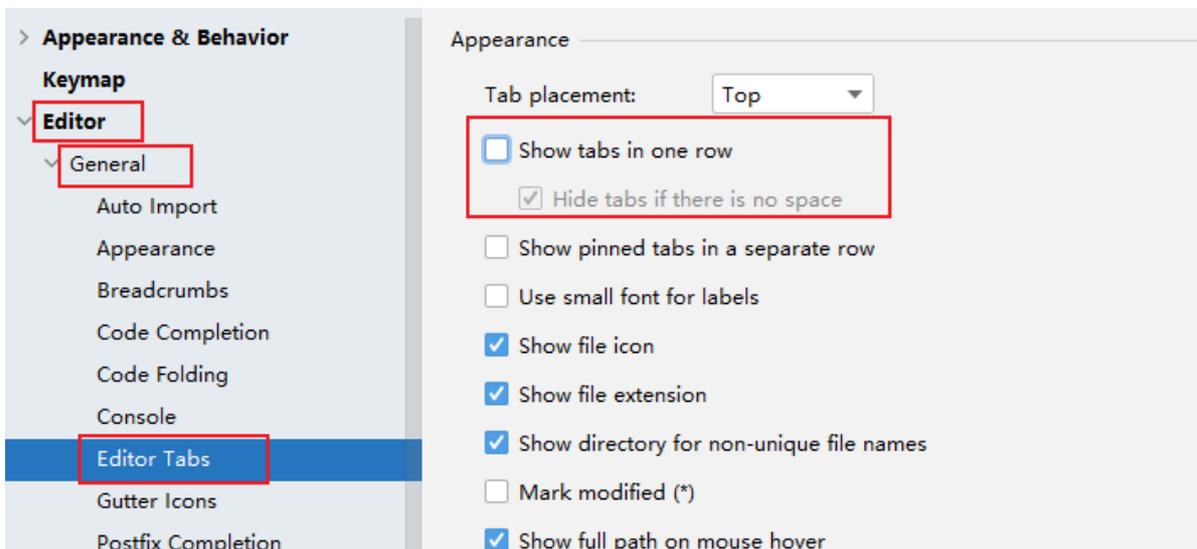


选择registry..., 找到"ide.suppress.double.click.handler", 把复选框打上勾就可以取消双击shift出现搜索框了。

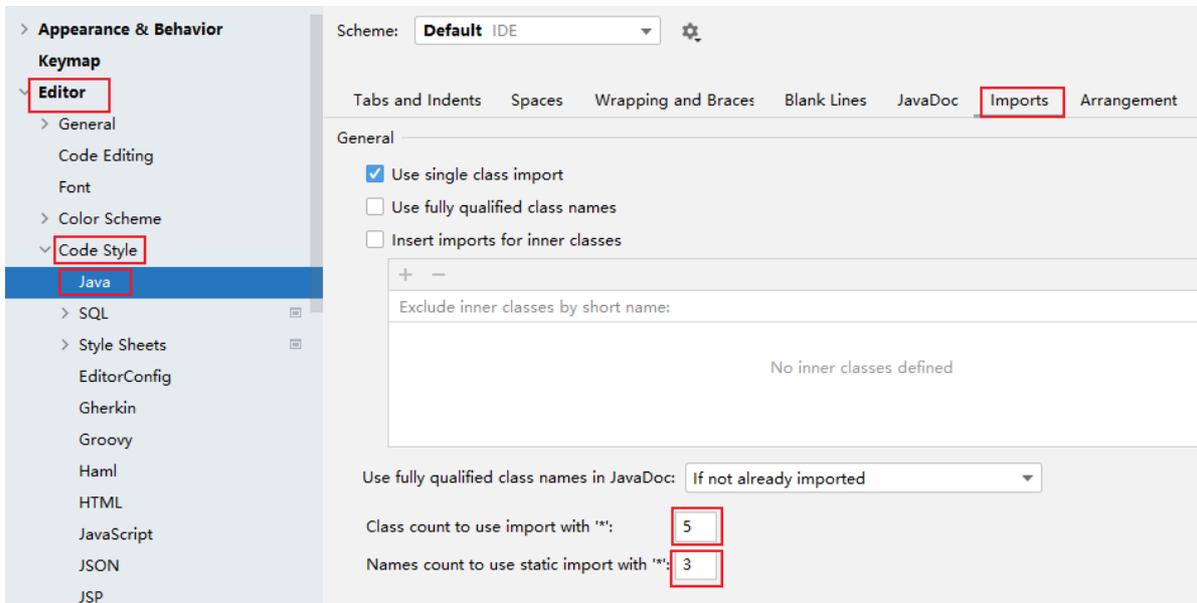


5.14 其它设置

- 是否在单行显式编辑器选项卡 (建议去掉勾选)



- 设置代码样式: 比如, 设置import显示"*"时的个数



总结:

以上这些设置看似只是针对当前Project设置的, 但是新建的其它Project也是同样适用的。

6. 工程与模块管理

6.1 IDEA项目结构

层级关系:

project(工程) - module(模块) - package(包) - class(类)

具体的:

一个project中可以创建多个module

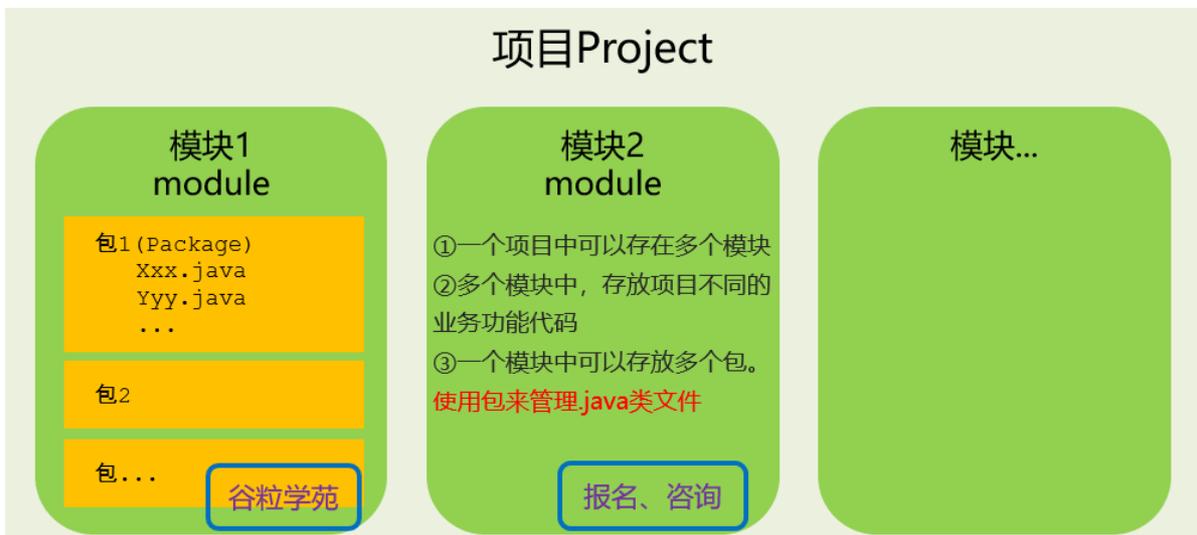
一个module中可以创建多个package

一个package中可以创建多个class

这些结构的划分, 是为了方便管理功能代码。

6.2 Project和Module的概念

在 IntelliJ IDEA 中, 提出了Project和Module这两个概念。



在 IntelliJ IDEA 中Project是 **最顶级的结构单元**，然后就是Module。目前，主流的大型项目结构基本都是多Module的结构，这类项目一般是 **按功能划分** 的，比如：user-core-module、user-facade-module和user-hessian-module等等，模块之间彼此可以 **相互依赖**，有着不可分割的业务关系。因此，对于一个Project来说：

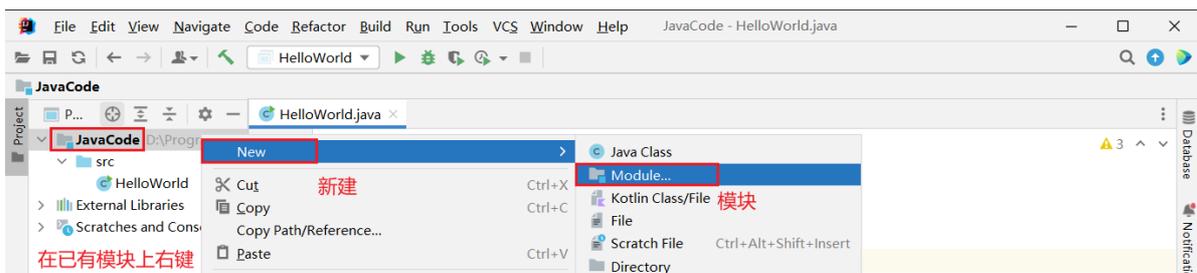
- 当为单Module项目的时候，这个单独的Module实际上就是一个Project。
- 当为多Module项目的时候，多个模块处于同一个Project之中，此时彼此之间具有 **相互依赖** 的关联关系。
- 当然多个模块没有建立依赖关系的话，也可以作为单独一个“小项目”运行。

6.3 创建Module

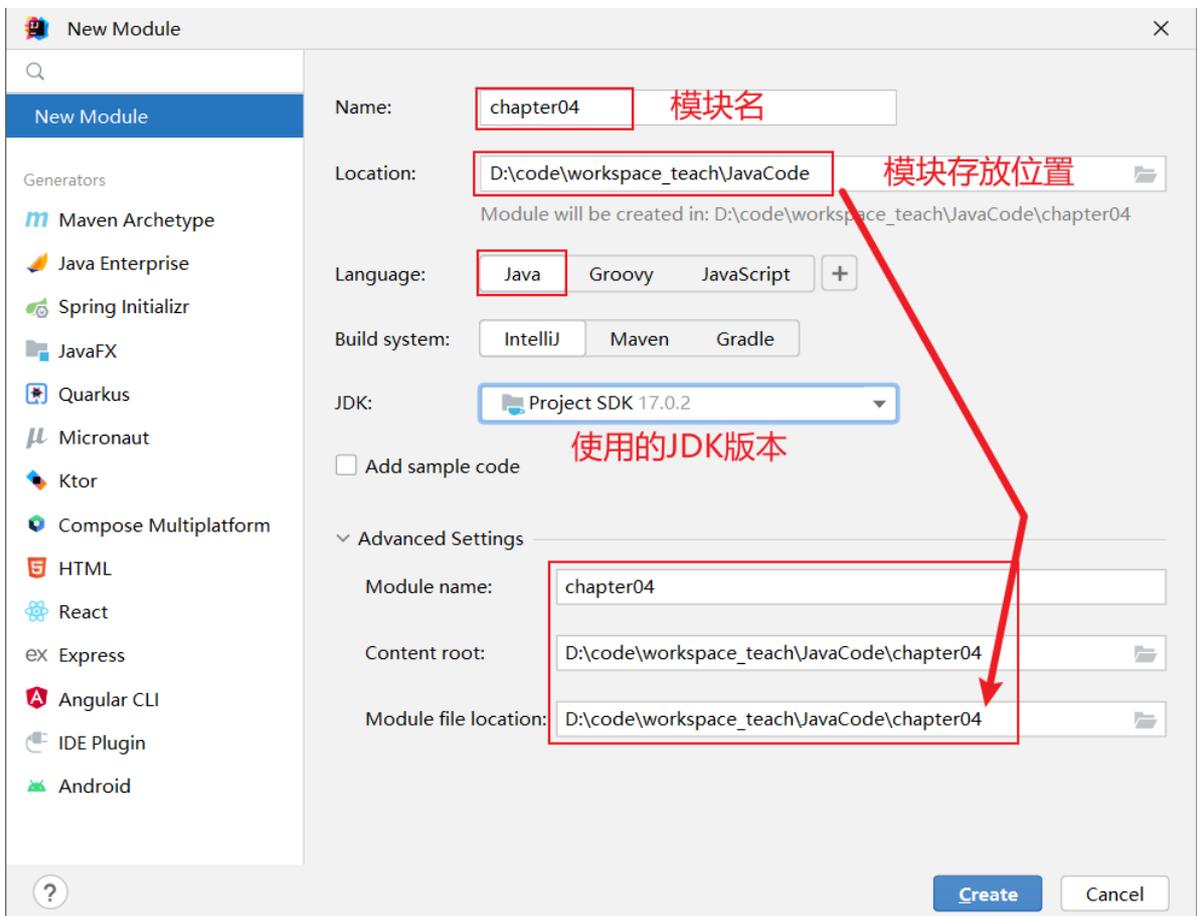
建议创建“Empty空工程”，然后创建多模块，每一个模块可以独立运行，相当于一个小项目。JavaSE阶段不涉及到模块之间的依赖。后期再学习模块之间的依赖。

步骤：

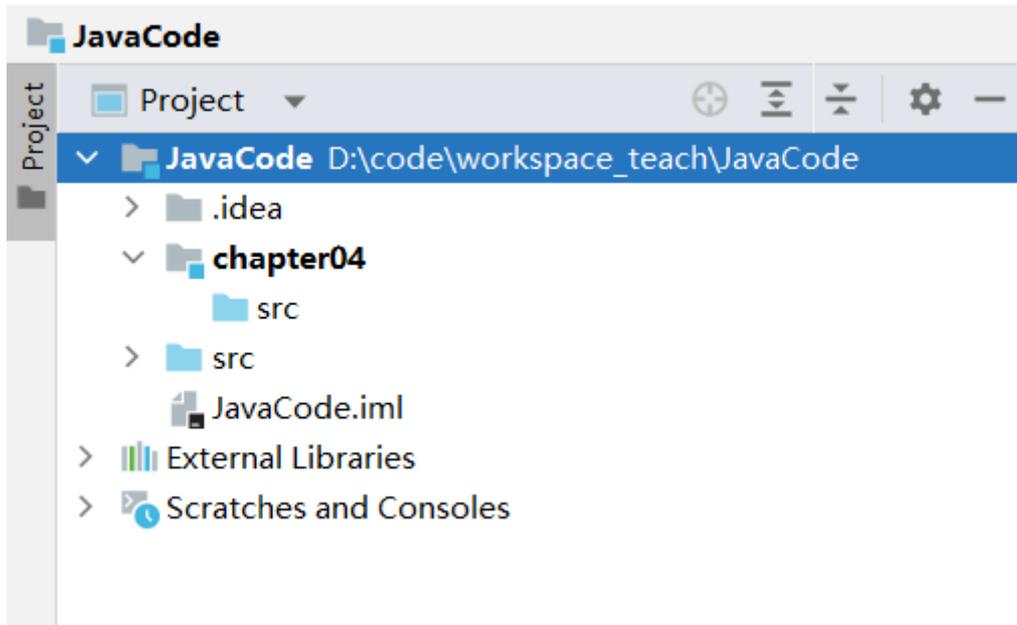
(1) 选择创建模块



(2) 选择模块类型：这里选择创建Java模块，给模块命名，确定存放位置

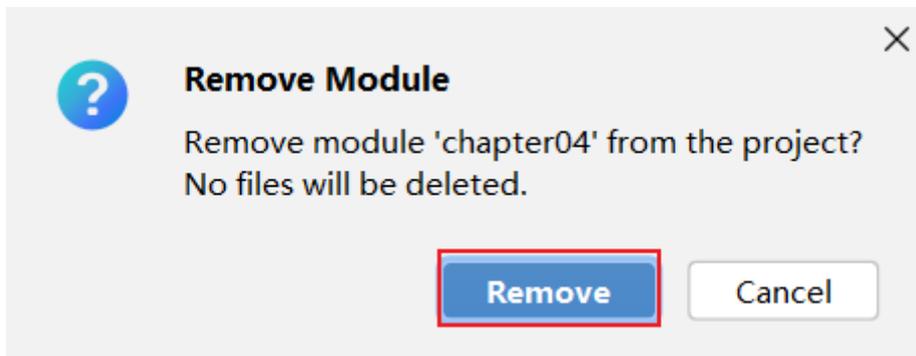
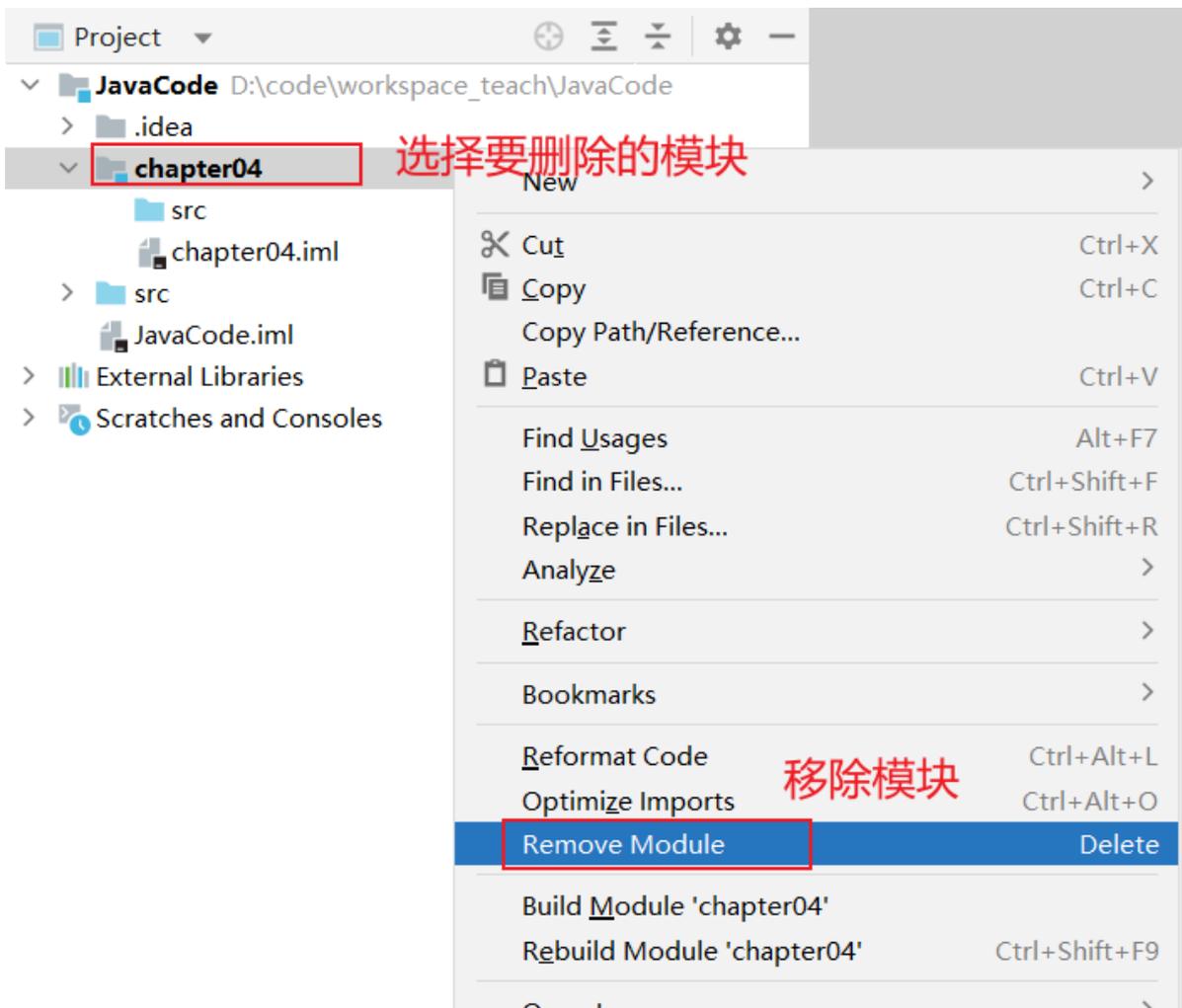


(4) 模块声明在工程下面

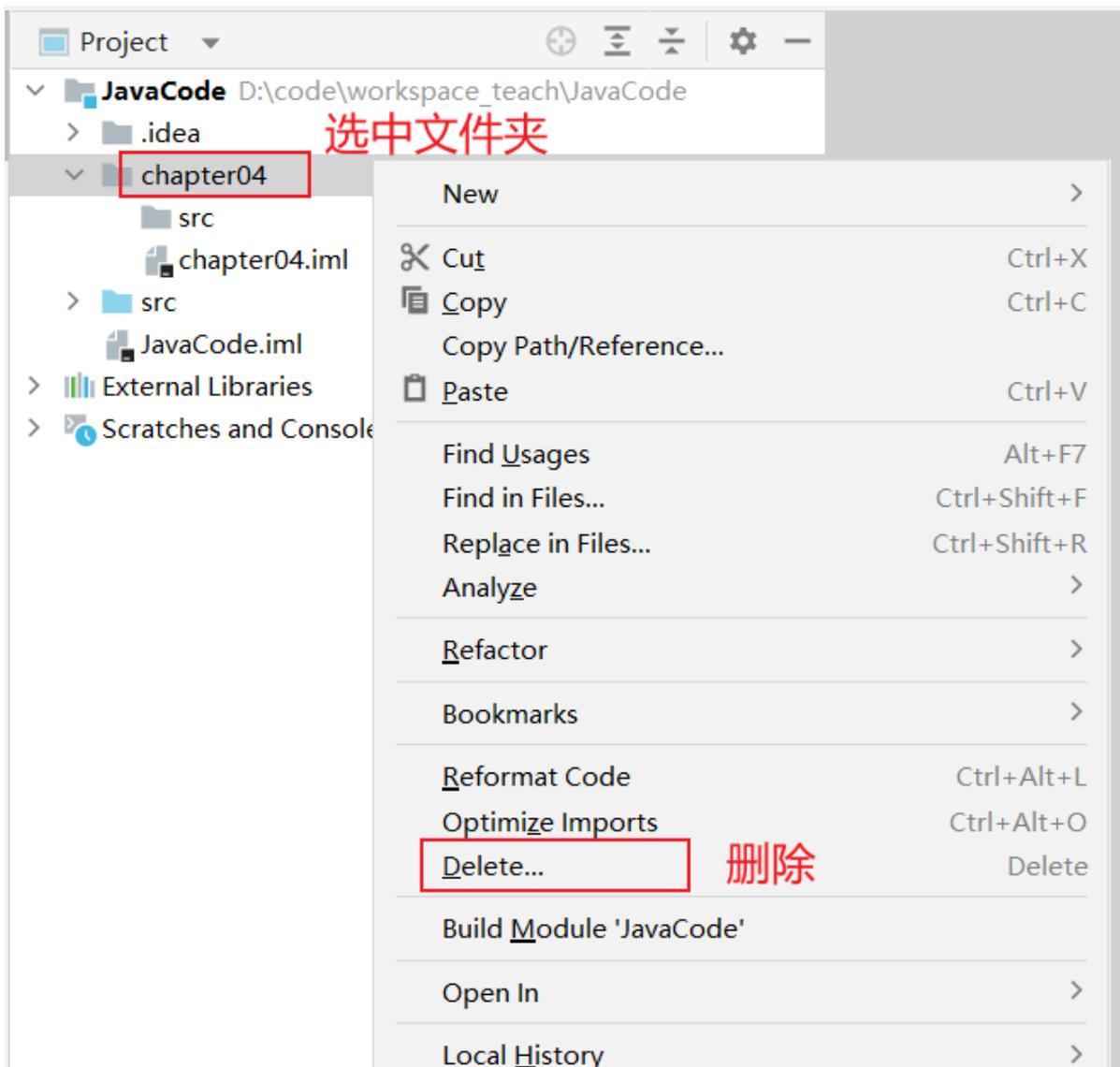


6.4 删除模块

(1) 移除模块

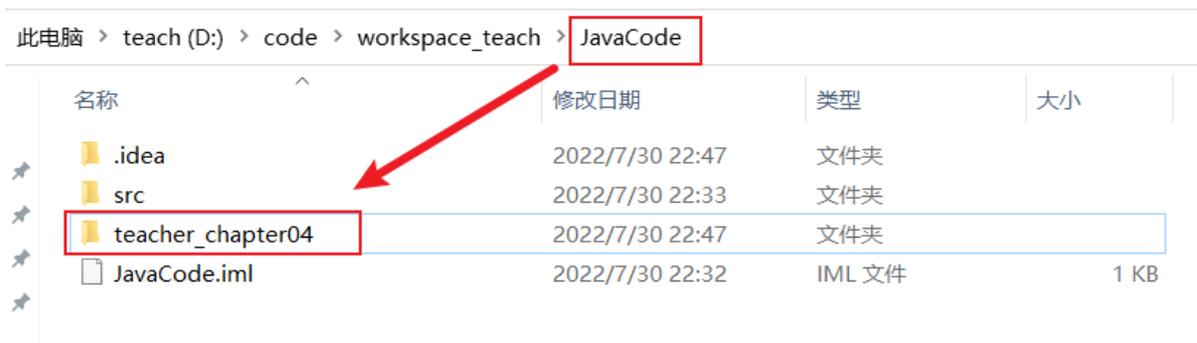


(2) 彻底删除模块

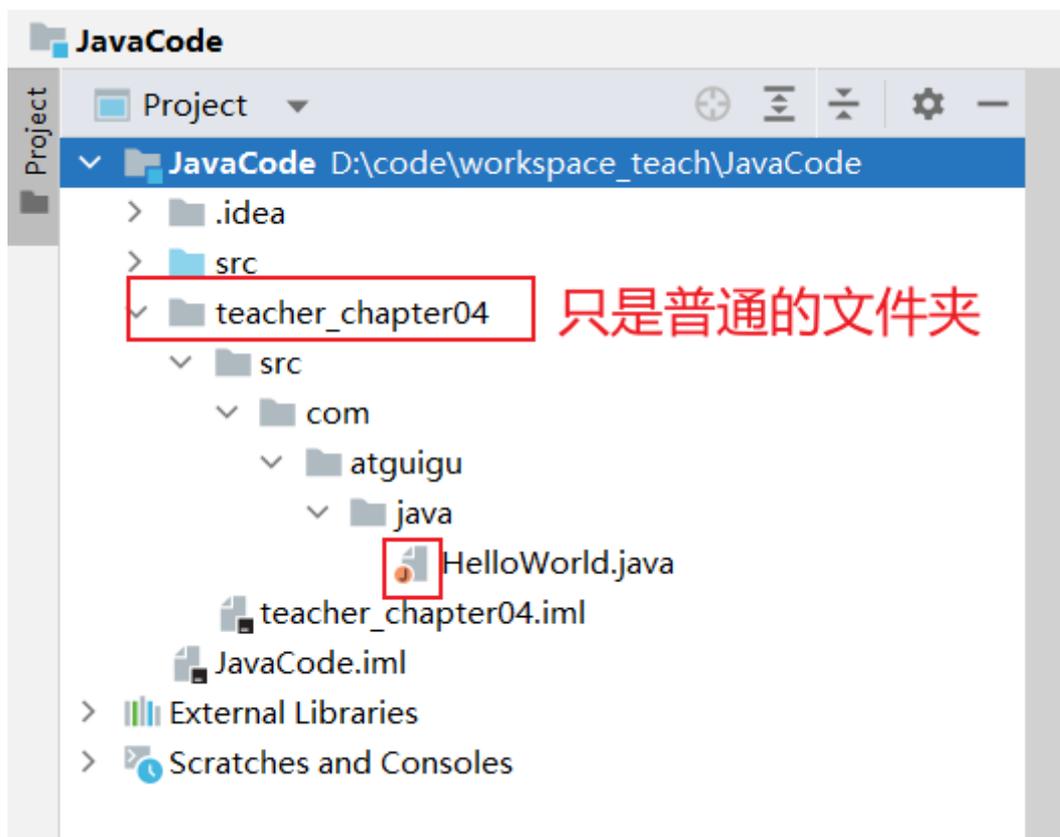


6.5 导入老师的模块

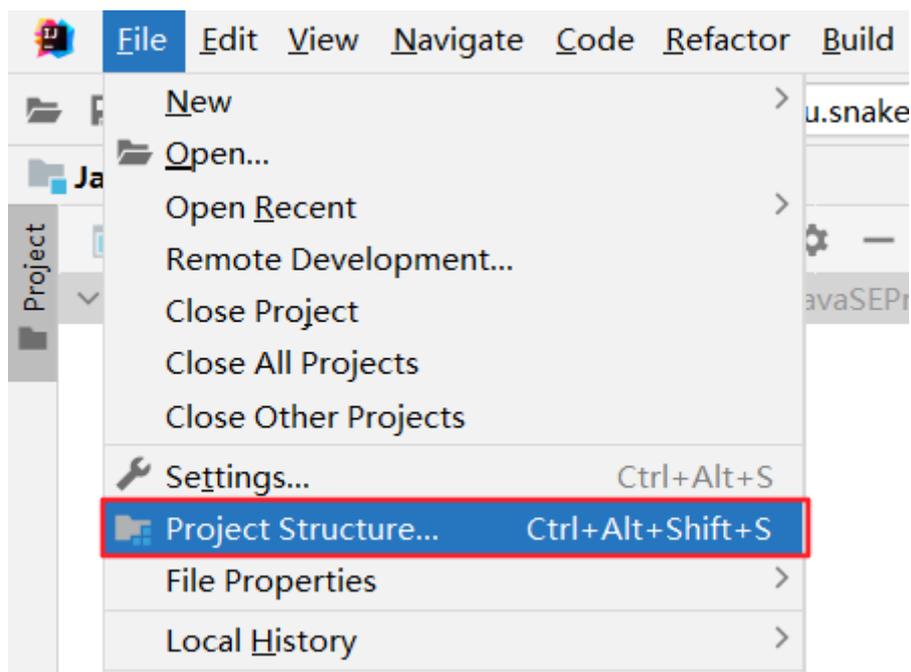
(1) 将老师的模块 `teacher_chapter04` 整个的复制到自己IDEA项目的路径下

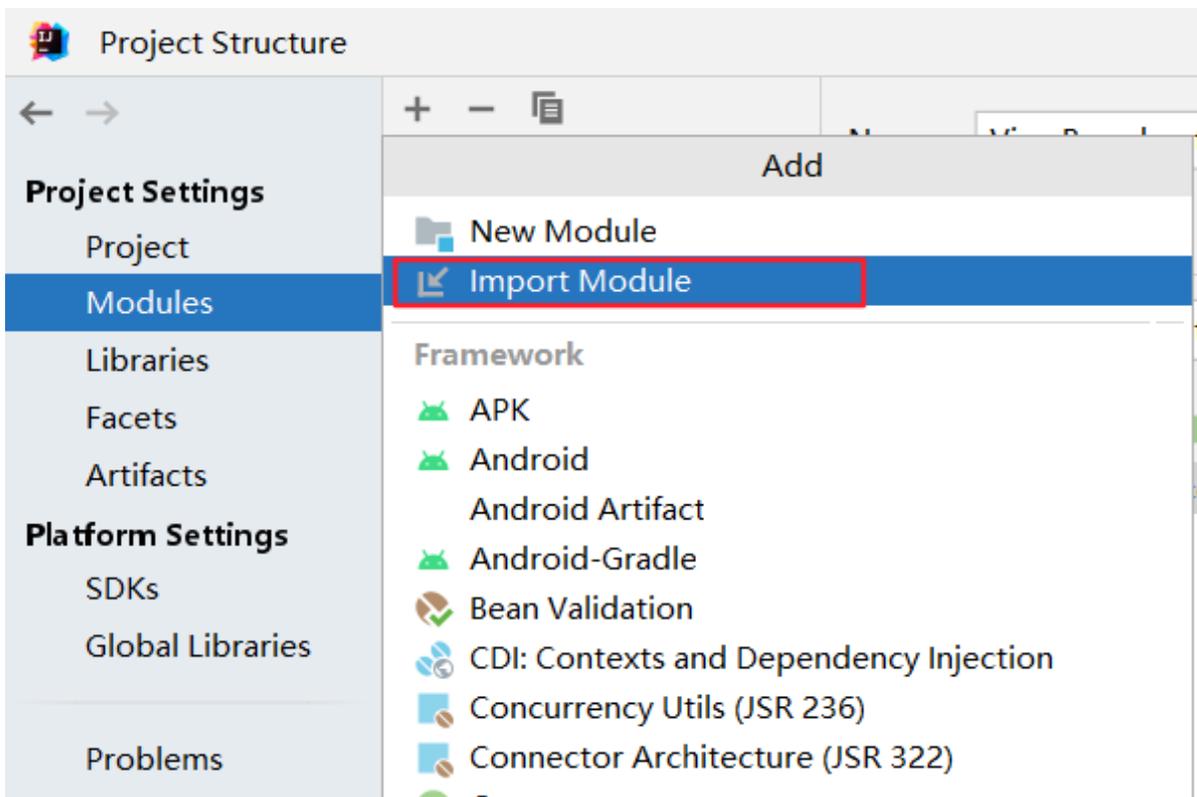


接着打开自己IDEA的项目，会在项目目录下看到拷贝过来的module，只不过不是以模块的方式呈现。

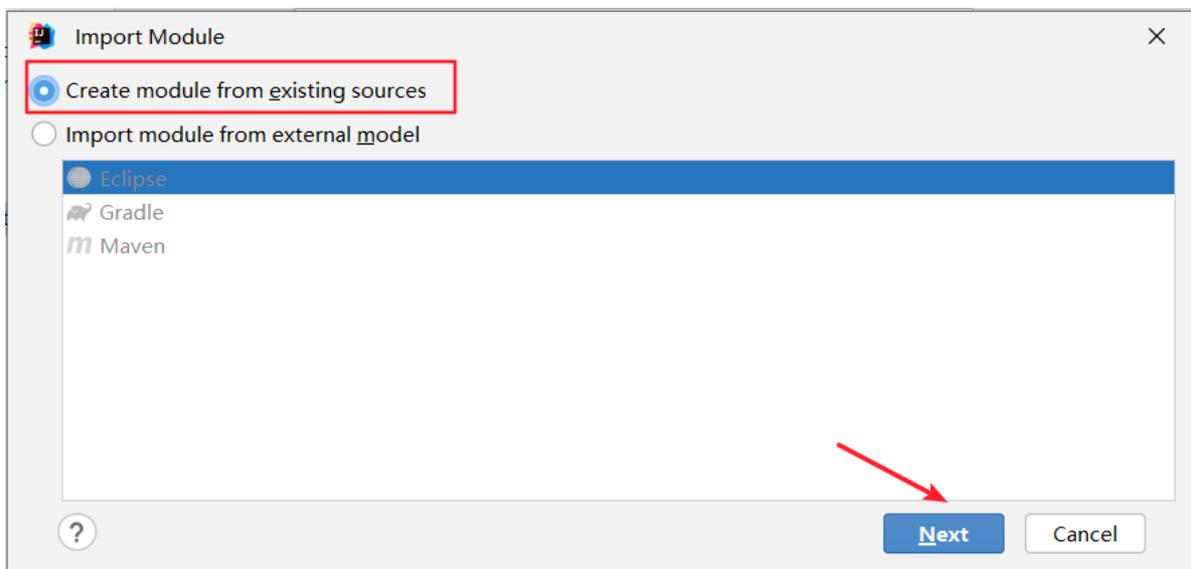
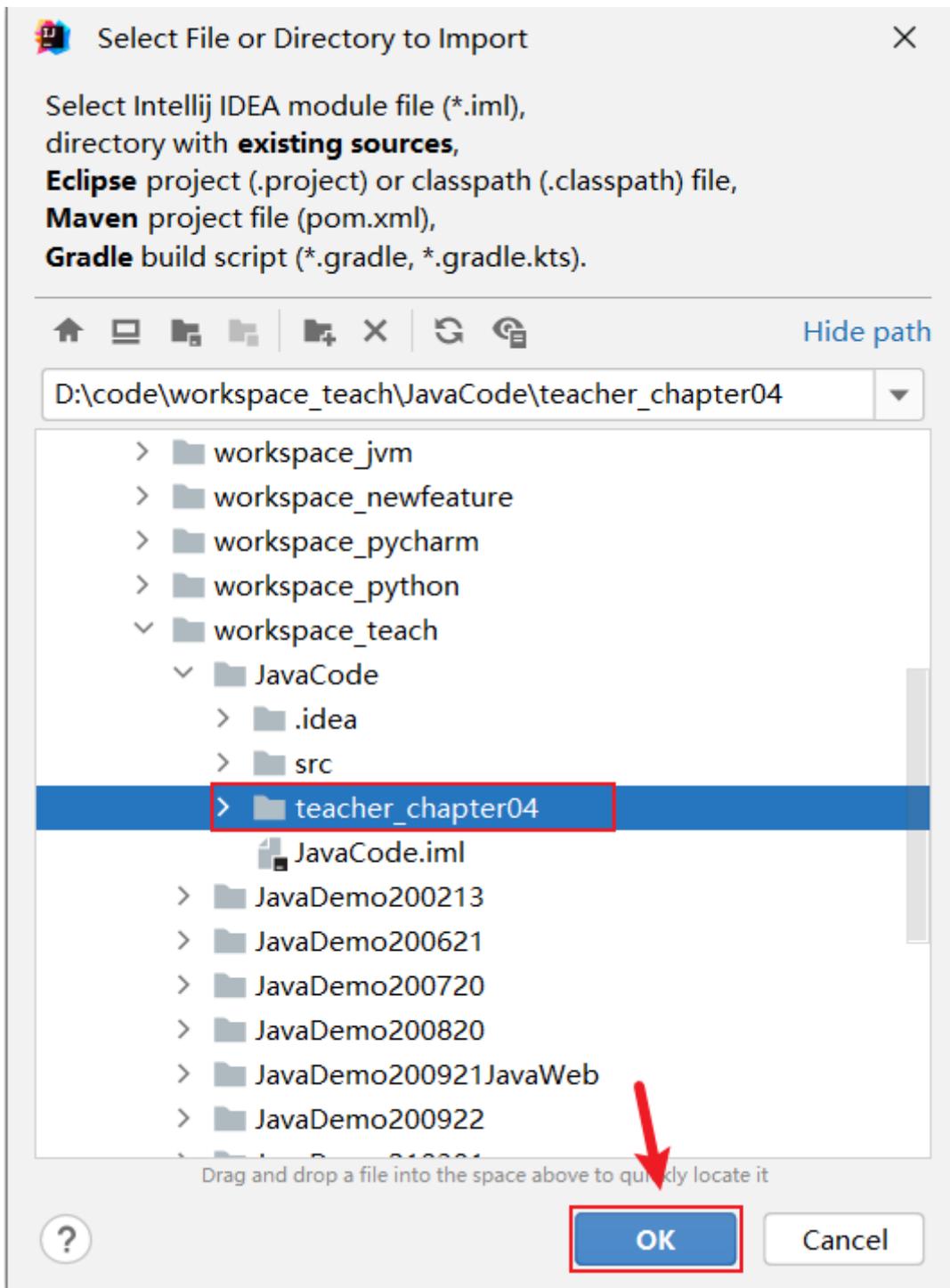


(2) 查看Project Structure, 选择import module

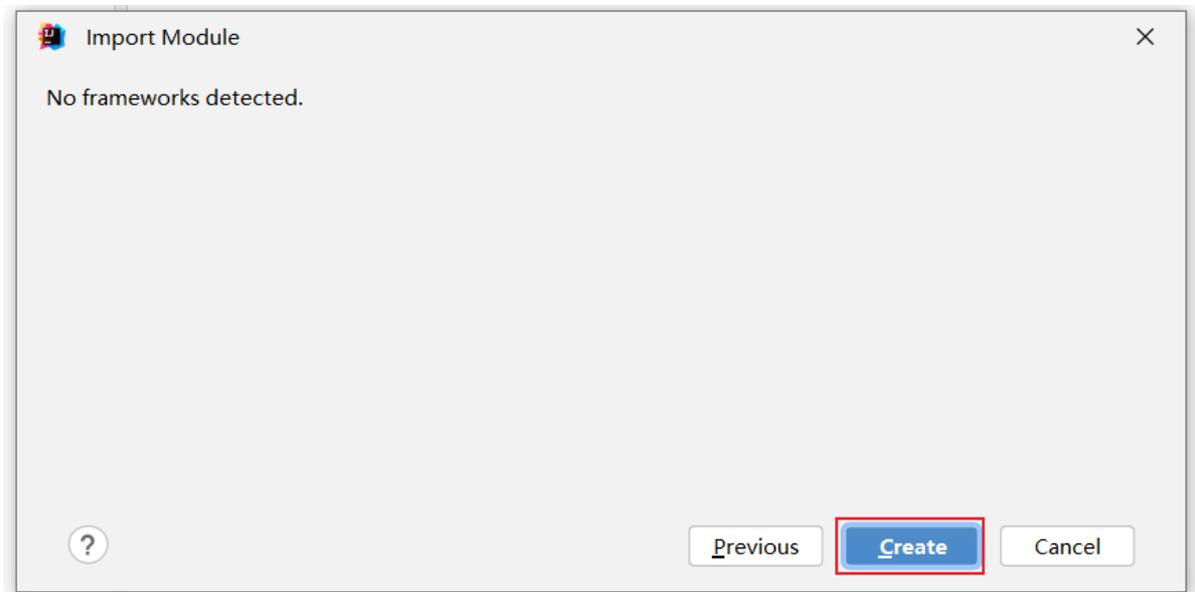
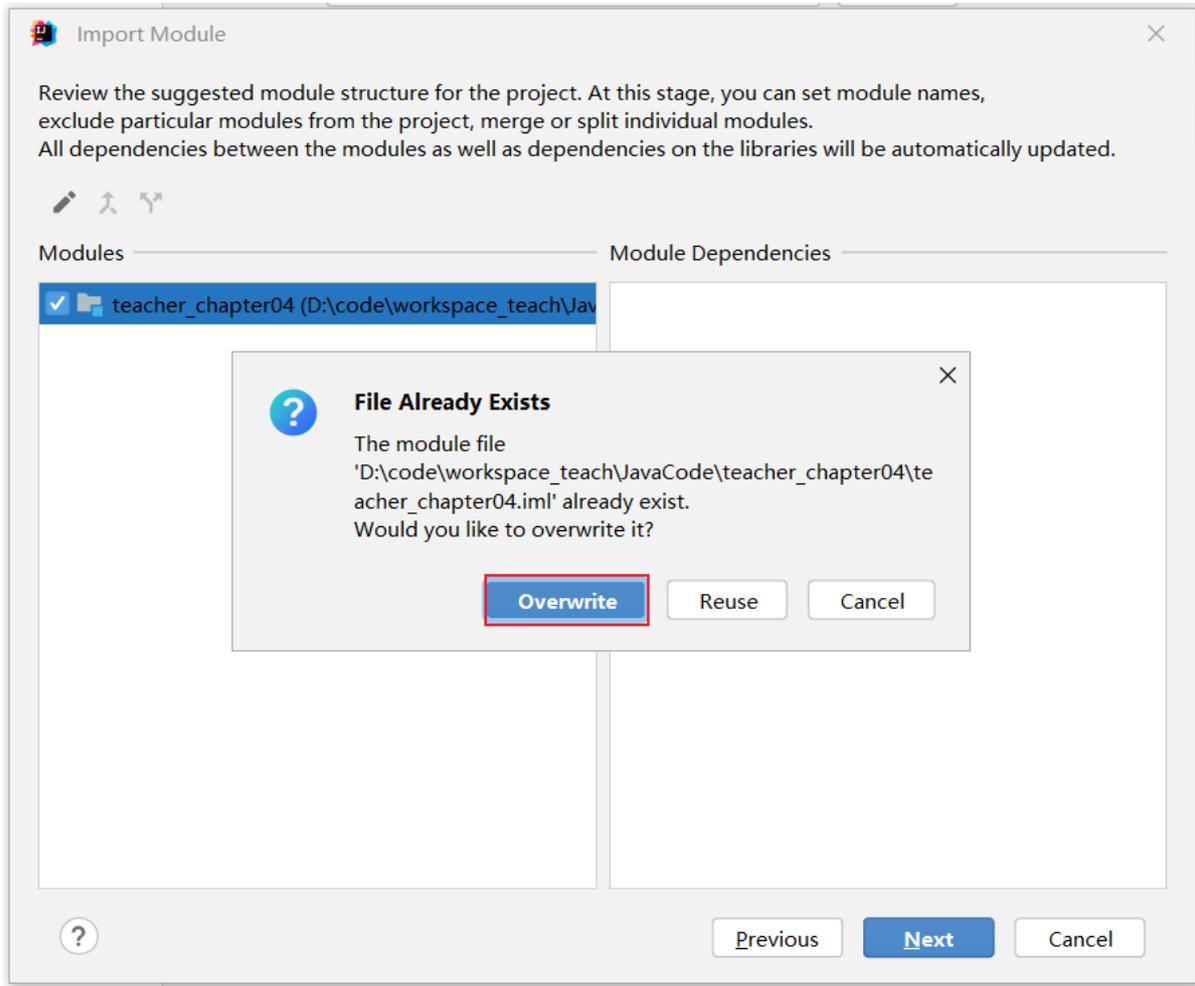




(3) 选择要导入的module:



(4) 接着可以一路Next下去，最后选择Overwrite

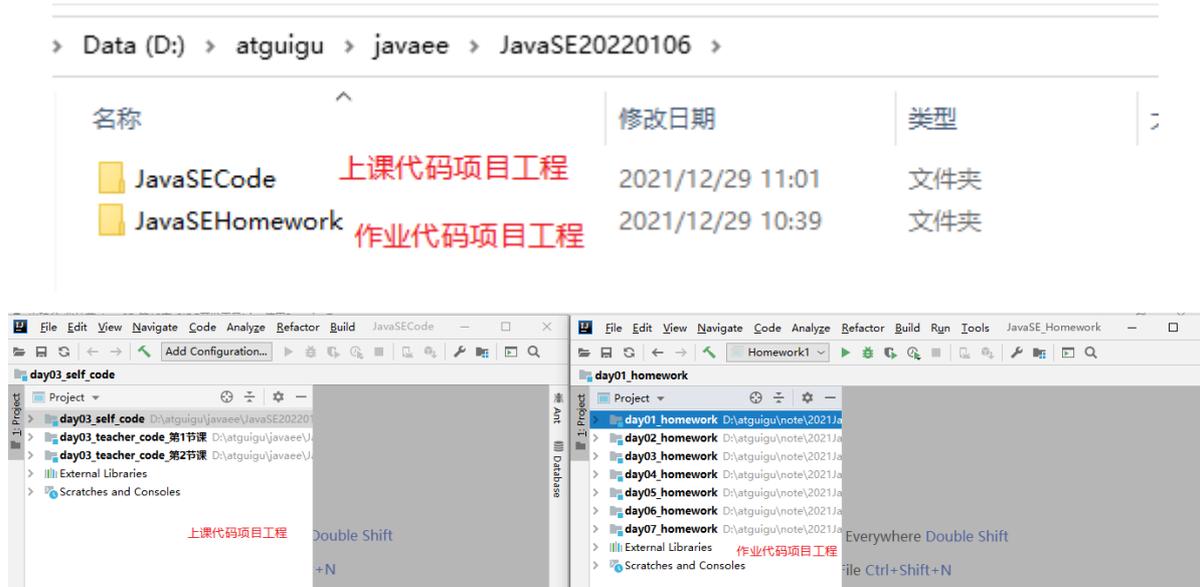


最后点击OK即可了。

6.6 同时打开两个IDEA项目工程

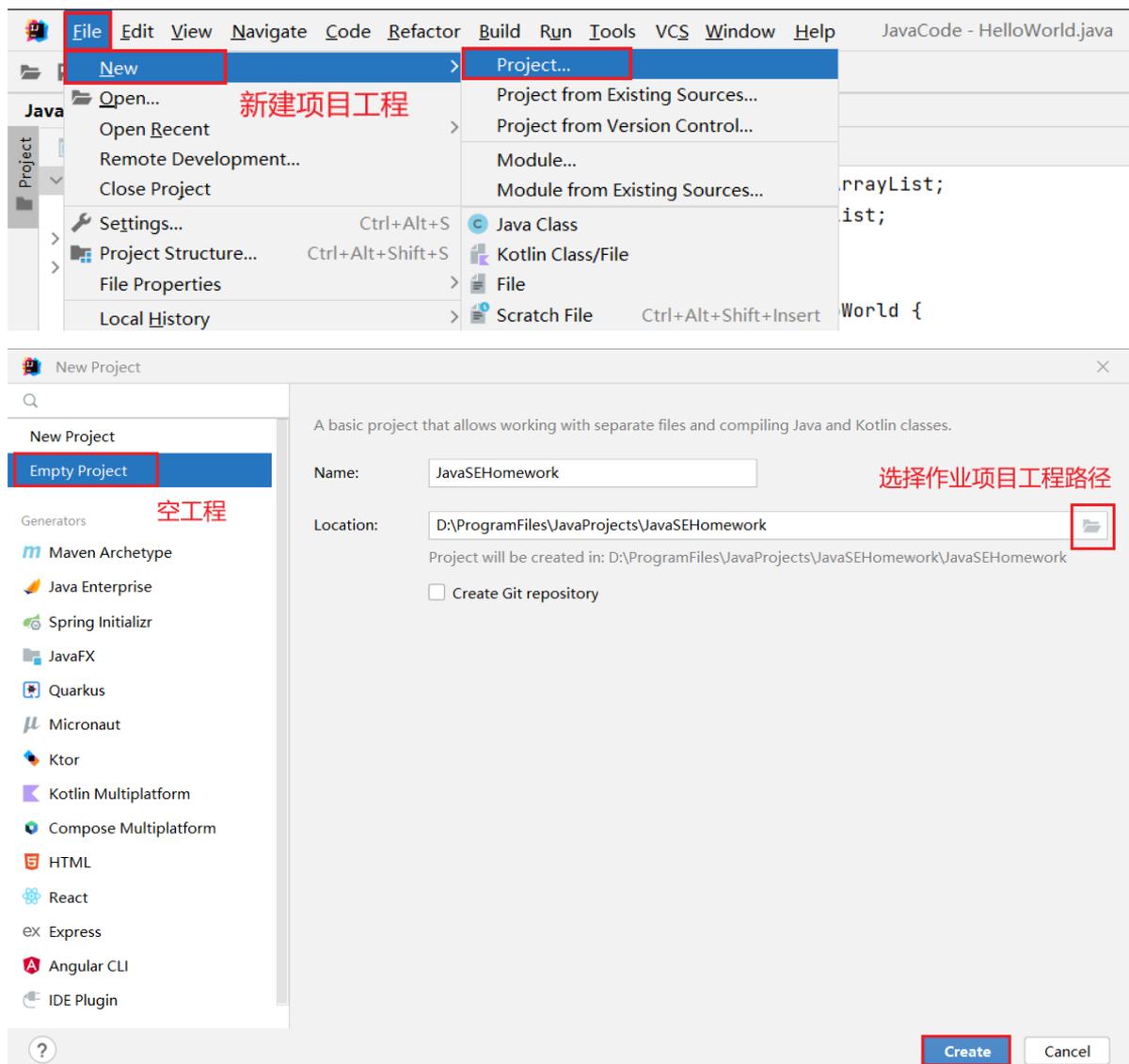
1、两个IDEA项目工程效果

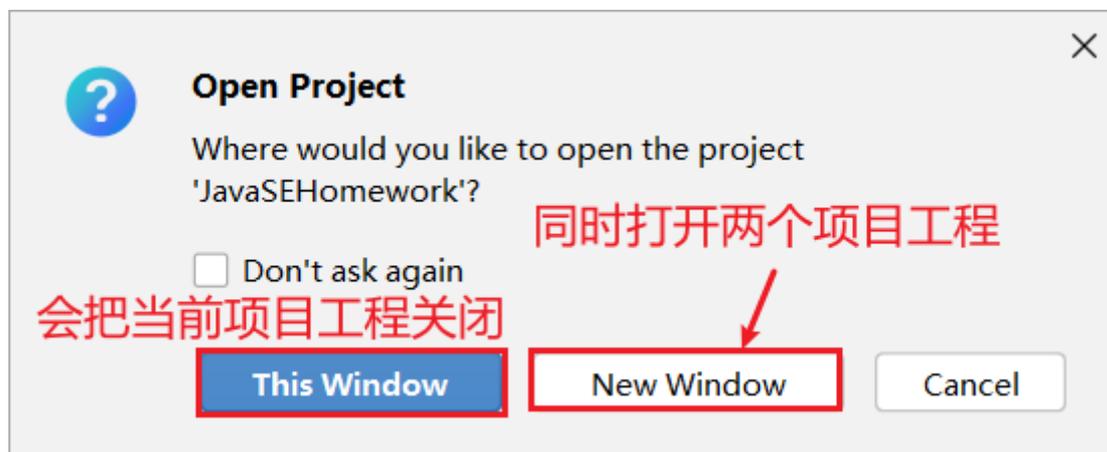
有些同学想要把上课练习代码和作业代码分开两个IDEA项目工程。



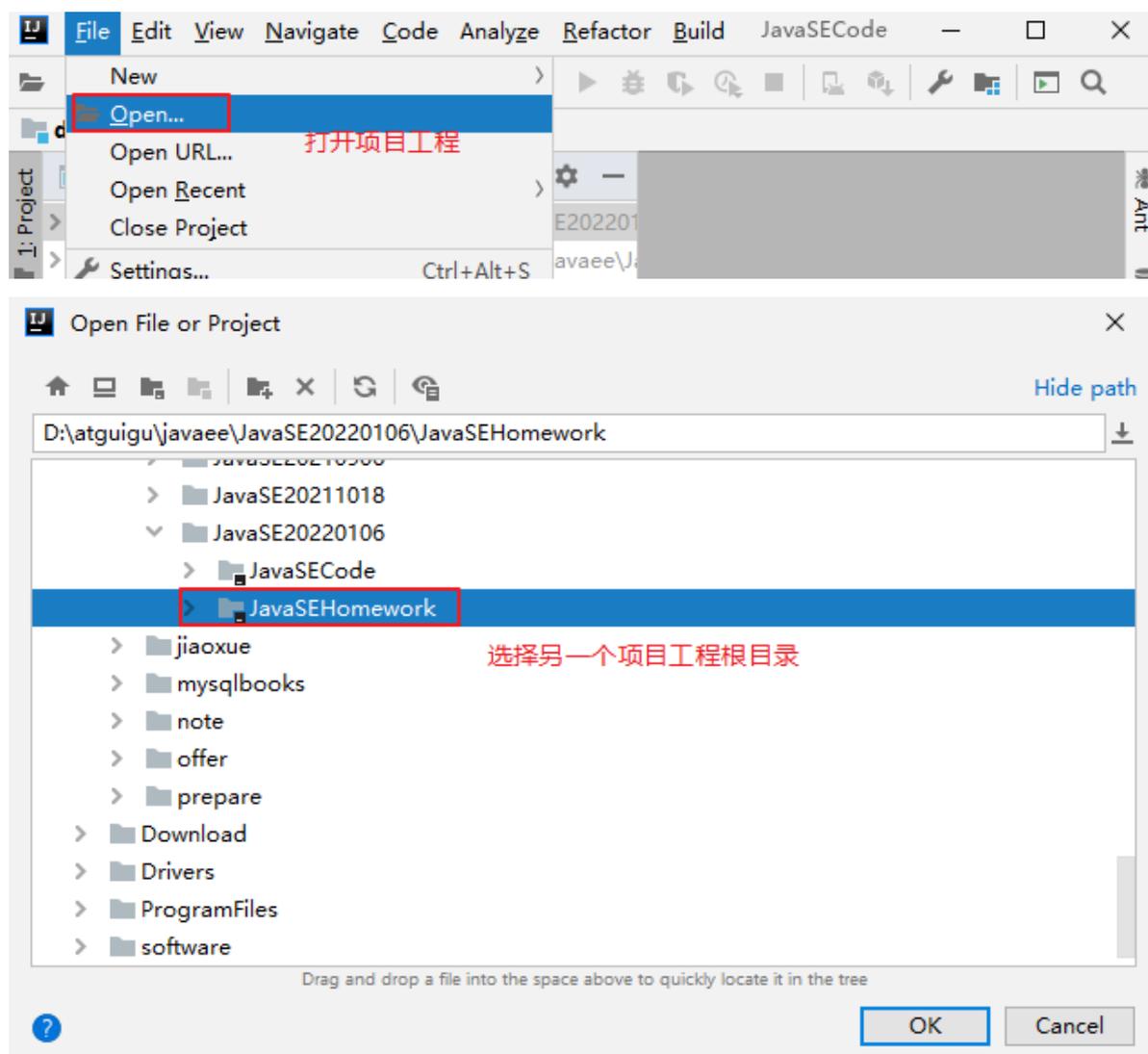
2、新建一个IDEA项目

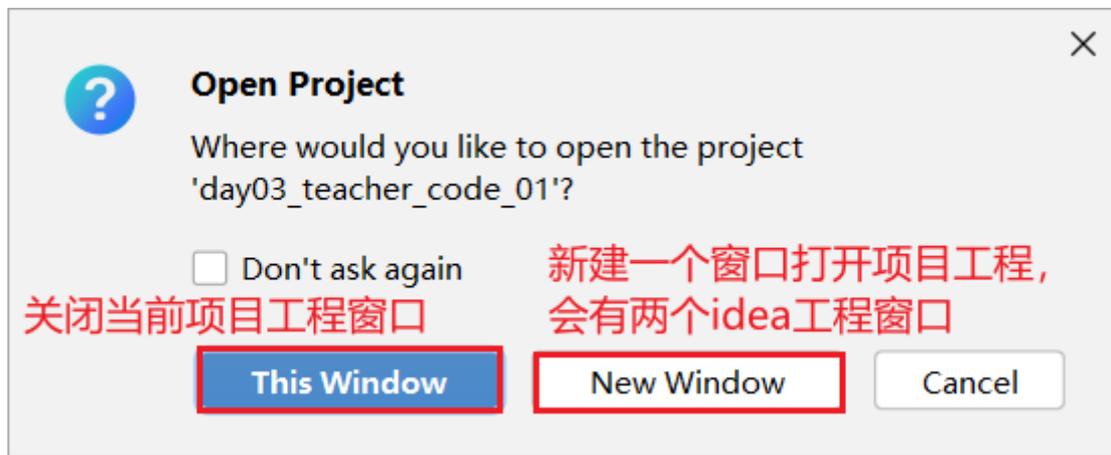
注意：第一次需要新建，之后直接打开项目工程即可





3、打开两个IDEA项目



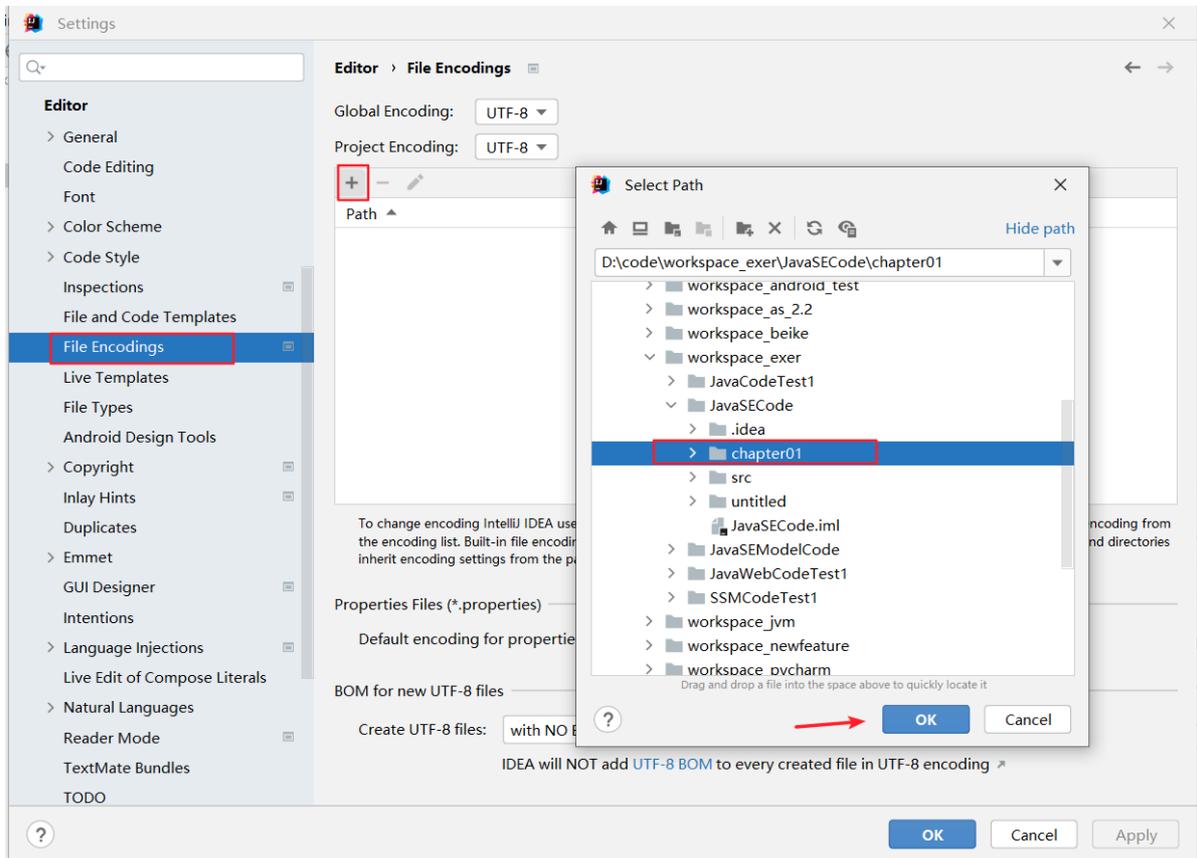


6.7 导入前几章非IDEA工程代码

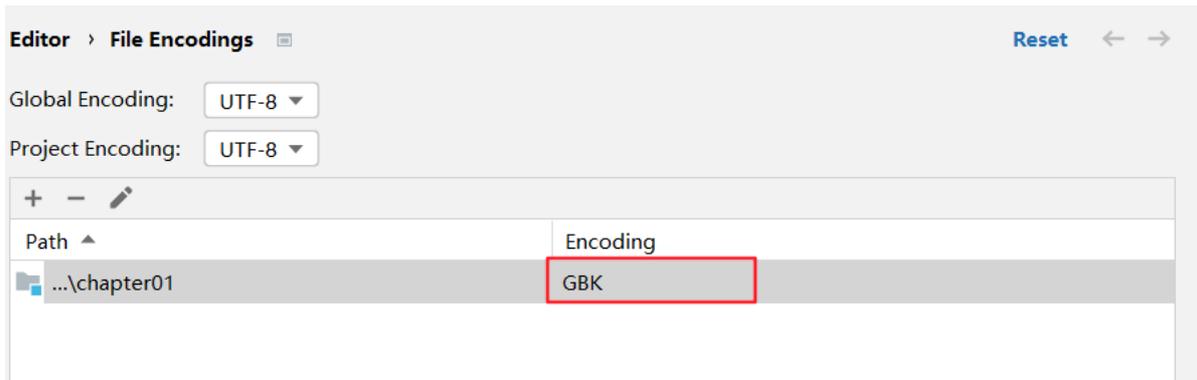
1、创建chapter01、chapter02、chapter03等章节的module

将相应章节的源文件粘贴到module的src下。

myhello	2022/6/15 16:20	文件夹	
章节案例、练习、真题	2022/6/15 16:20	文件夹	
Hello.java.txt	2022/6/10 0:31	EditPlus 3	0 KB
第1个Java程序小结.txt	2022/6/10 11:16	EditPlus 3	2 KB
BeiJing.class	2022/6/10 1:11	Java class file	1 KB
HeartPrint.class	2022/6/10 15:13	Java class file	1 KB
HelloChina.class	2022/6/10 1:11	Java class file	1 KB
HelloJava.class	2022/6/10 12:05	Java class file	1 KB
HelloWorld.class	2022/6/10 1:11	Java class file	1 KB
PersonallInfo.class	2022/6/10 15:06	Java class file	1 KB
ShangHai.class	2022/6/10 1:11	Java class file	1 KB
HeartPrint.java	2022/6/10 15:14	JAVA 文件	1 KB
HelloJava.java	2022/6/10 14:59	JAVA 文件	1 KB
HelloWorld.java	2022/6/10 1:14	JAVA 文件	1 KB
PersonallInfo.java	2022/6/10 15:07	JAVA 文件	1 KB

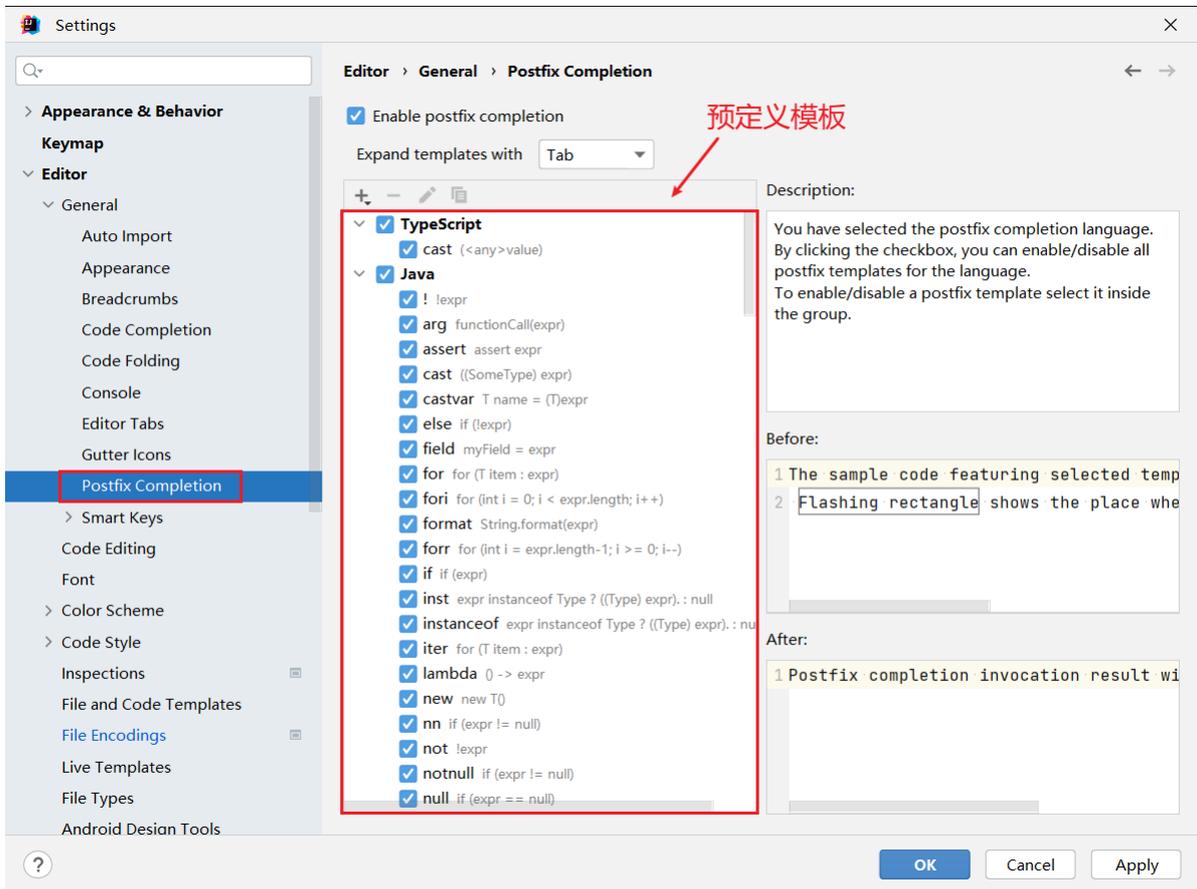


改为GBK，确认即可。如图：

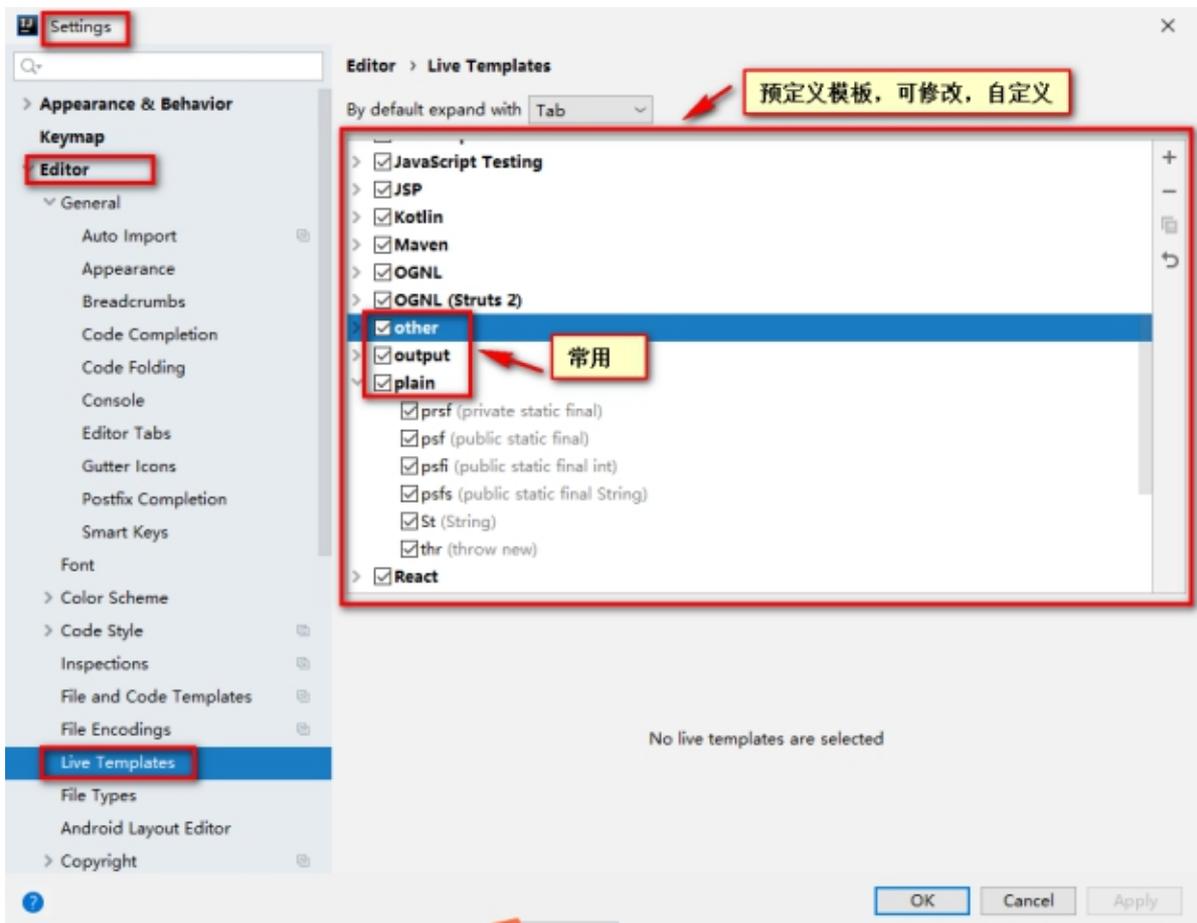


7. 代码模板的使用

7.1 查看Postfix Completion模板(后缀补全)



7.2 查看Live Templates模板(实时模板)



7.3 常用代码模板

1、非空判断

- 变量.null: if(变量 == null)
- 变量.nn: if(变量 != null)
- 变量.notNull: if(变量 != null)
- ifn: if(xx == null)
- inn: if(xx != null)

2、遍历数组和集合

- 数组或集合变量.fori: for循环
- 数组或集合变量.for: 增强for循环
- 数组或集合变量.forr: 反向for循环
- 数组或集合变量.iter: 增强for循环遍历数组或集合

3、输出语句

- sout: 相当于System.out.println
- soutm: 打印当前方法的名称
- soutp: 打印当前方法的形参及形参对应的实参值
- soutv: 打印方法中声明的最近的变量的值
- 变量.sout: 打印当前变量值
- 变量.soutv: 打印当前变量名及变量值

4、对象操作

- 创建对象
 - Xxx.new .var : 创建Xxx类的对象, 并赋给相应的变量
 - Xxx.new .field: 会将方法内刚创建的Xxx对象抽取为一个属性
- 强转
 - 对象.cast: 将对象进行强转
 - 对象.castvar: 将对象强转后, 并赋给一个变量

5、静态常量声明

- psf: public static final
- psfi: public static final int
- psfs: public static final String
- prsf: private static final

7.4 自定义代码模板

7.4.1 自定义后缀补全模板

Keymap

Editor

- General
 - Auto Import
 - Appearance
 - Breadcrumbs
 - Code Completion
 - Code Folding
 - Console
 - Editor Tabs
 - Gutter Icons
 - Postfix Completion**
 - Smart Keys
 - Code Editing
 - Font

Enable postfix completion

Expand templates with

- 1 JavaScript and TypeScript (from*/just())
- 2 **Java** (ust(iter))
- 3 SQL
- 4 Groovy
- 5 TypeScript (pr)

- toObservable Observable.from*/just()
- toSingle Single.from*/just
- try try { exp } catch(Exception e)
- twr try(Type f = new Type()) catch (Exception e)
- var T name = expr
- while while (expr) {}
- list

JavaScript

- ! !expr

Edit 'list' Template

Key:

Minimum language level:

Applicable expression types:

+ -

Nothing to show

Apply to the topmost expression

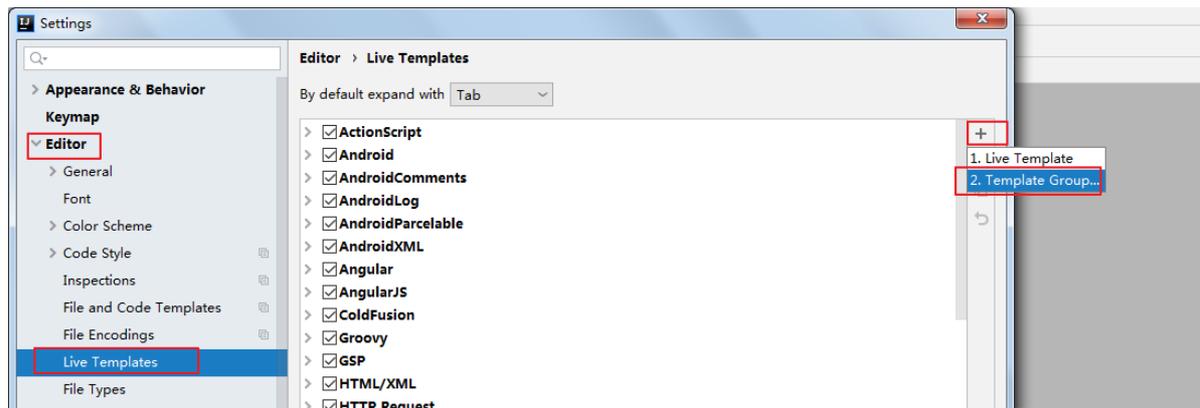
Use \$EXPR\$ variable to refer target expression

Use static import if possible

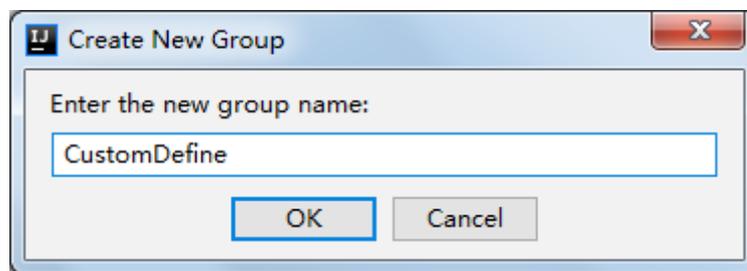
7.4.2 自定义Live Templates

例如：定义sop代表System.out.println();语句

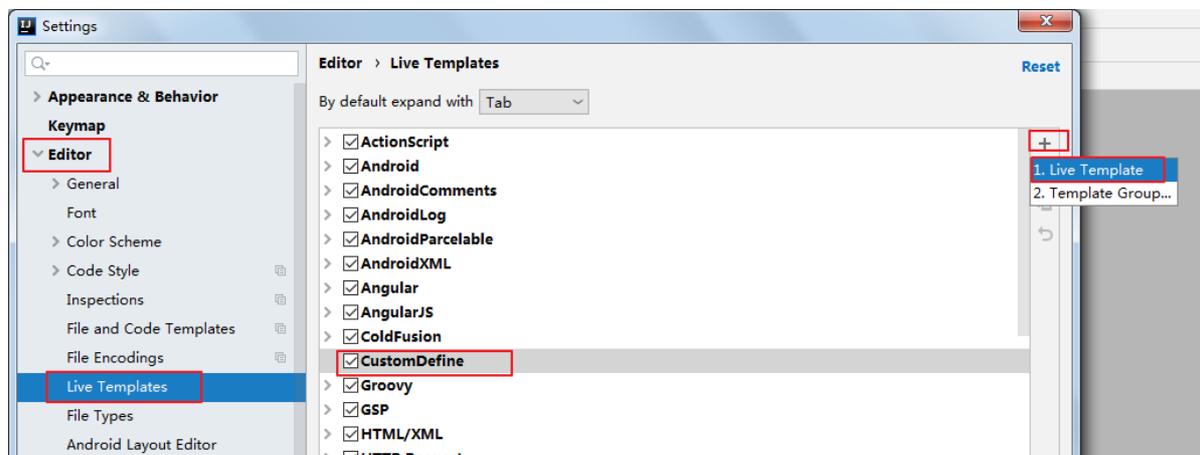
①在Live Templates中增加模板



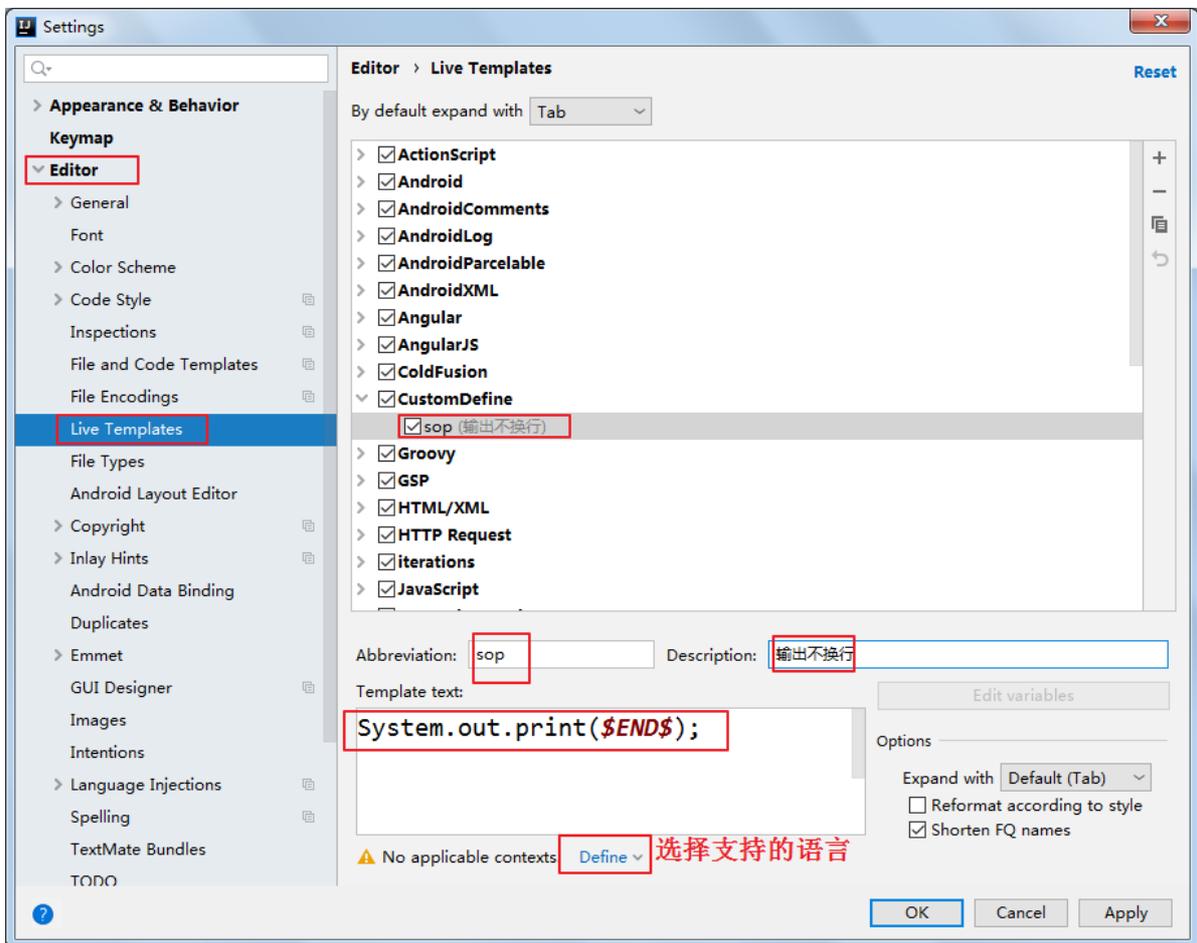
②先定义一个模板的组，这样方便管理所有自定义的代码模板



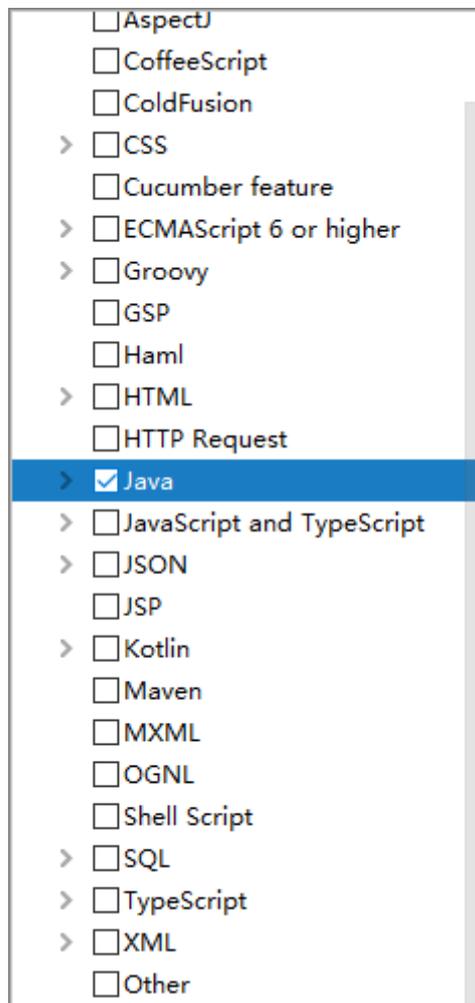
③在模板组里新建模板



④定义模板（以输出语句为例）



- Abbreviation: 模板的缩略名称
- Description: 模板的描述
- Template text: 模板的代码片段
- 模板应用范围。比如点击Define。选择如下：应用在java代码中。



其它模板1: 单元测试模板:

```
@Test
public void test$var1$(){
    $var2$
}
```

Abbreviation: Description:

Template text:

```
@Test
public void test$var1$(){
    $var2$
}
```

[Edit variables](#)

Options

Expand with

Reformat according to style

Use static import if possible

Shorten EQ names

Applicable in Java; Java: statement, consumer function, expression, 'else' position, decl...
[Change](#) ▾

其它模板2: 创建多线程

```
new Thread(){
    public void run(){
        $var$
    }
};
```

Abbreviation: Description:

Template text:

```
new Thread(){
    public void run(){
        $var$
    }
};
```

[Edit variables](#)

Options

Expand with

Reformat according to style

Use static import if possible

Shorten EQ names

Applicable in Java; Java: statement, consumer function, expression, 'else' position, decl...

[Change](#)

其它模板3: 冒泡排序

```
for(int $INDEX$ = 1; $INDEX$ < $ARRAY$.length; $INDEX$++) {
    for(int $INDEX2$ = 0; $INDEX2$ < $ARRAY$.length-$INDEX$; $INDEX2$++) {
        if($ARRAY[$INDEX2$] > $ARRAY[$INDEX2$+1]){
            $ELEMENT_TYPE$ temp = $ARRAY[$INDEX2$];
            $ARRAY[$INDEX2$] = $ARRAY[$INDEX2$+1];
            $ARRAY[$INDEX2$+1] = temp;
        }
    }
}
```

Abbreviation: Description:

Template text:

```
for(int $INDEX$ = 1; $INDEX$ < $ARRAY$.length;
    for(int $INDEX2$ = 0; $INDEX2$ < $ARRAY$.le
        if($ARRAY[$INDEX2$] > $ARRAY[$INDEX2$
            $ELEMENT_TYPE$ temp = $ARRAY[$INDE
            $ARRAY[$INDEX2$] = $ARRAY[$INDEX2
            $ARRAY[$INDEX2$+1] = temp;
```

[Edit variables](#)

Options

Expand with

Reformat according to style

Use static import if possible

Shorten EQ names

Applicable in Java; Java: statement, consumer function, expression, 'else' position, decl...

[Change](#)