

实验 01：开发环境

一、实验目的

- 1、了解 Java 程序编译运行基本原理；
- 2、掌握 Java 开发环境配置方法；
- 3、掌握 Java 开发工具安装配置方法；

二、实验学时

2 学时

三、实验类型

验证性

四、实验需求

1、硬件

每人配备计算机 1 台，建议优先使用个人计算机开展实验。

2、软件

安装 IntelliJ IDEA，以及 Java 运行所需要的相关基础环境。

3、网络

本地主机能够访问互联网和实验中心网络。

4、工具

ipmsg

五、实验任务

- 1、完成 Java 开发环境配置；
- 2、完成 Java 开发工具安装配置；
- 3、使用开发工具完成一个 Hello World 实例；

六、实验内容及步骤

1、配置本地开发环境

(1)配置环境变量

JAVA_HOME=C:\jdk1.8.0_291

CLASSPATH=.;%Java_Home%\bin;%Java_Home%\lib\dt.jar;%Java_Home%\lib\tools.jar;

Path 添加

%Java_Home%\bin;

%Java_Home%\jre\bin;

如图 1-1

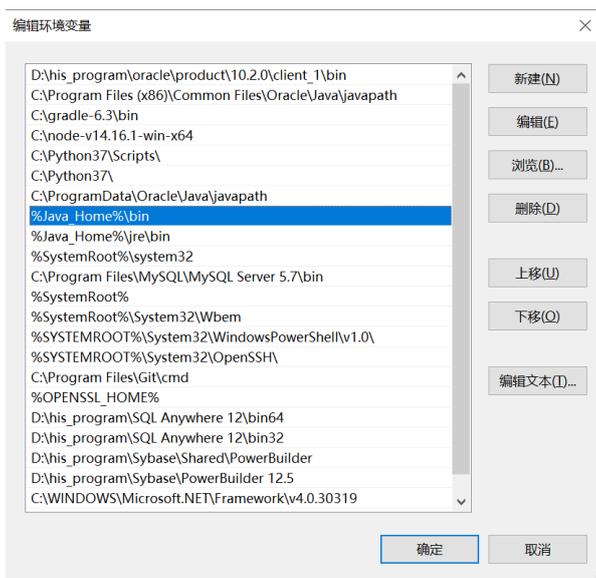


图 1-1

(2)验证环境变量

如图 1-2、 1-3、 1-4

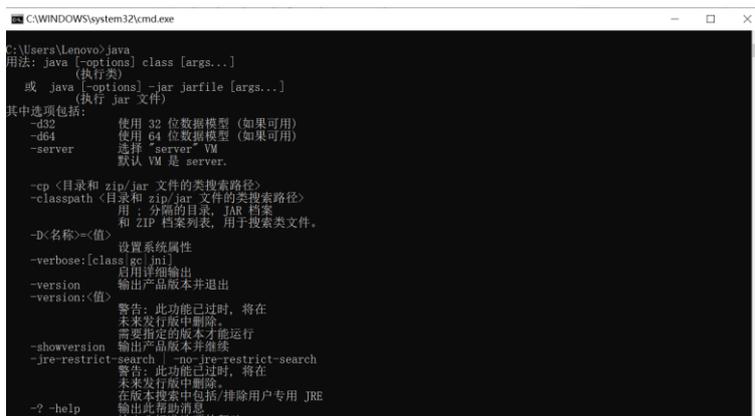


图 1-2

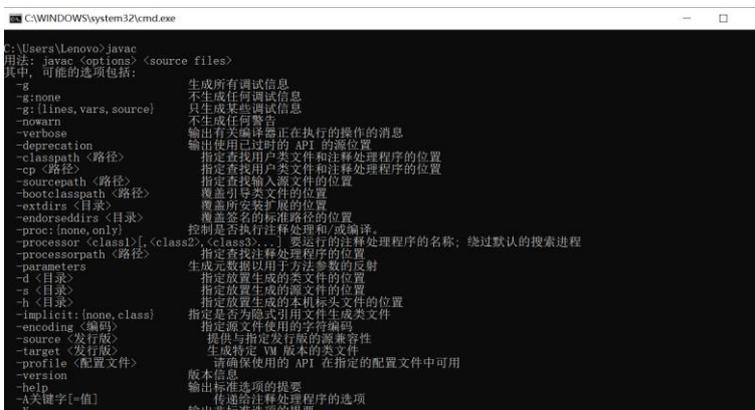


图 1-3

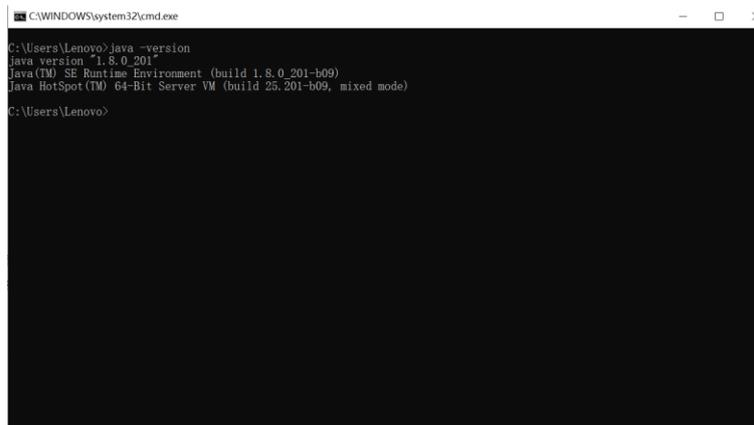


图 1-4

2、下载 IntelliJ IDEA

(1) 访问 jetbrains 官网

<https://www.jetbrains.com/idea/download/>

(2) 下载 IntelliJ IDEA。

如图 2-1

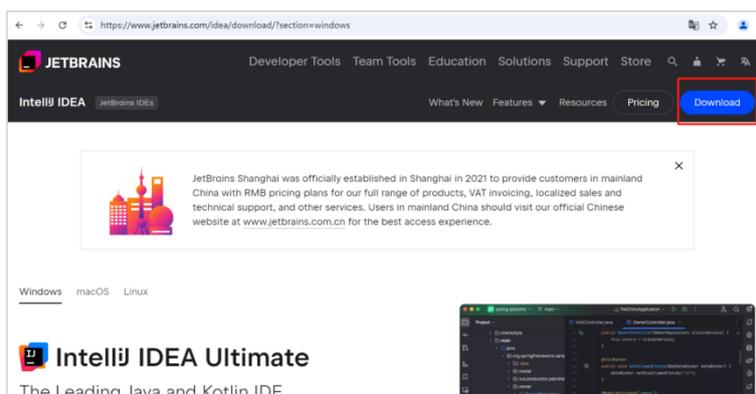


图 2-1

点击"Other versions", 如图 2-2

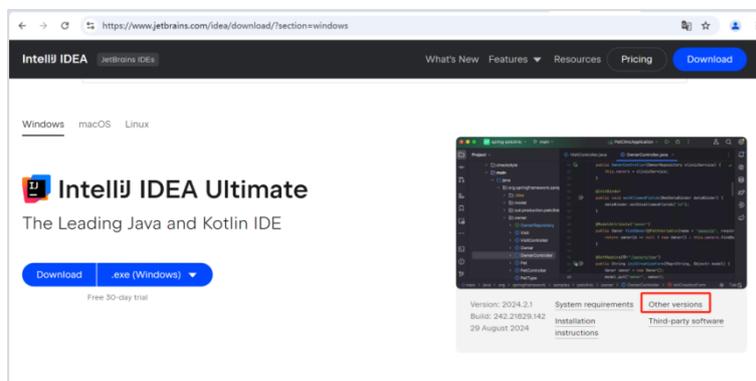


图 2-2

下载 2021.1.3 最终版。如图 2-3

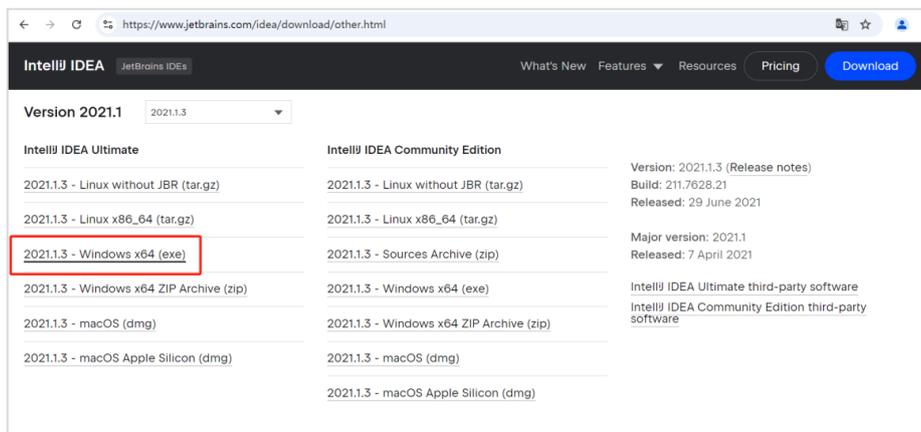


图 2-3

3、安装 IntelliJ IDEA

(1) 双击我们的安装包之后，来到 IDEA 的欢迎界面，我们此时只需要 Next 即可。如图 3-1 所示。

(2) 选择我们要安装 IDEA 的位置，它默认的安装位置是 C:\Program Files\JetBrains\IntelliJ IDEA 2021.1.3，如图 3-2 所示。

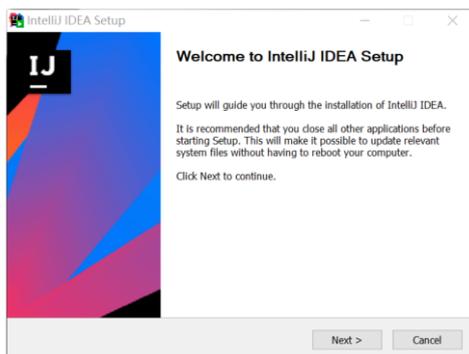


图 3-1

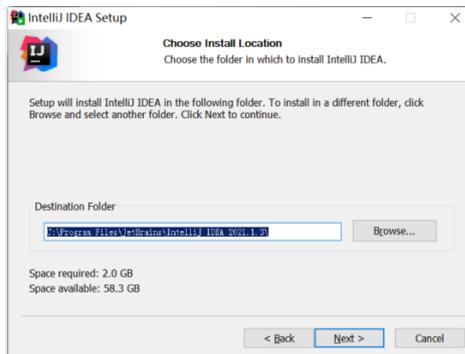


图 3-2

(3) 选择安装的配置，这里主要选择创建 32 还是 64 位的桌面快捷方式、是否要将其加入到系统 PATH 环境变量、是否关联相关后缀文件等等，可以根据自己的使用习惯来进行选择。推荐默认全都不勾选。如图 3-3 所示。

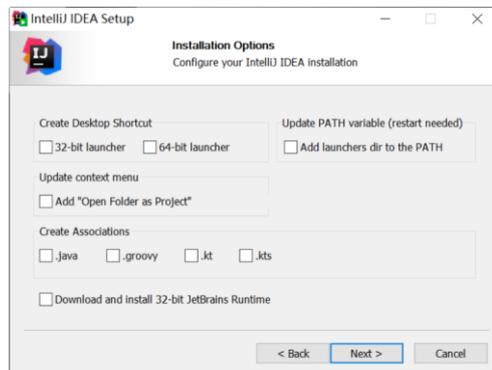


图 3-3

(4)安装的一些配置，选择开始菜单中给 IDEA 的文件夹名，这里默认即可，直接点击 Install，然后就是漫长的等待安装过程。如图 3-4, 3-5。

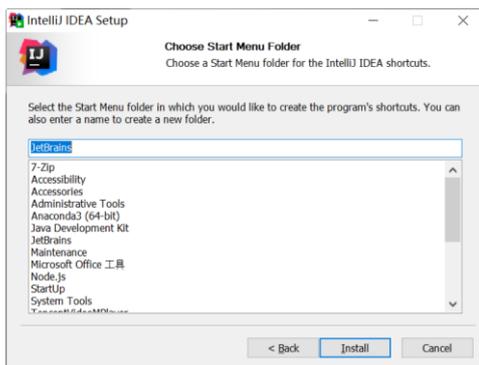


图 3-4

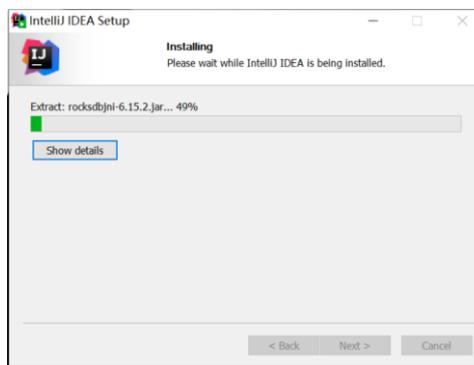


图 3-5

(5)显示此界面就说明我们的 IDEA 已经安装好了，直接点击 Finish 即可。如图 3-6。

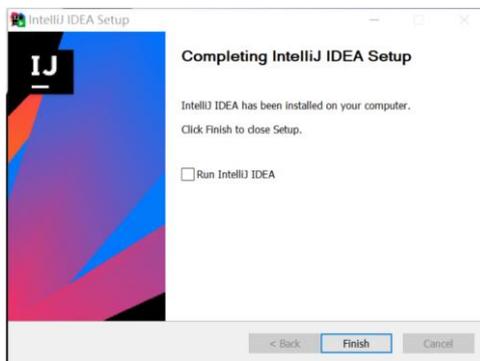


图 3-6

(6)激活。首次打开旗舰版 IDEA 时，会让你激活，这里有 3 种选择：

利用 JetBrains 账号登录激活；

利用 IDEA 激活码激活；

最后则是许可证服务器地址激活；

当然也可以选择 Evaluate for free，它是有 30 天的试用期的，只需要在这 30 天之内激活即可，否则 30 天之后将会弹窗不可用。如图 3-7。我们也可以下载社区版，对于社区版，没有激活这一说法，因为它是免费的，我们直接打开用就可以了，但相较于专业版部分功能缺失且需要单独购买插件。

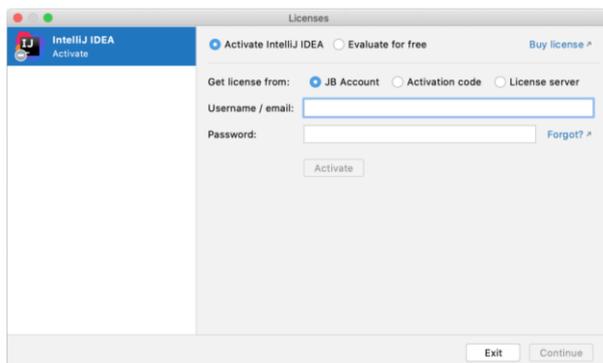


图 3-7

4、IntelliJ IDEA 配置

通常情况下使用默认配置即可

（1）编码设置

一般 IDEA 的编码是和系统保持一致的，而为了通用，减少不必要的编码问题，所以一般统一设置为 UTF-8，设置入口如下，依次打开 Settings→Editor →File Encodings，然后将 Global Encoding、Project Encoding、Default encoding for properties files 均设置为 UTF-8。

如图 4-1。

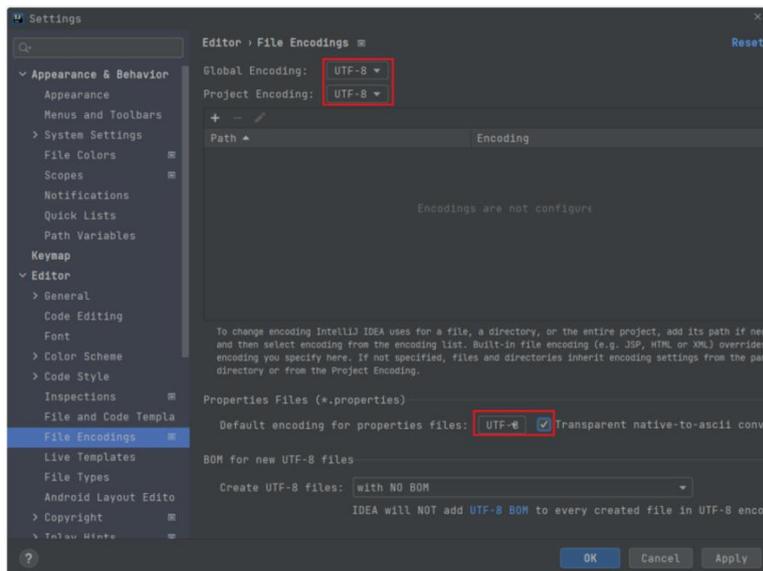


图 4-1

（2）外观设置

这里可以根据自己的喜好进行主题选择，也可以勾选右方的 Sync with OS 选项，这样 IDEA 就会根据你系统的明亮还是黑暗主题进行自动切换了。

如图 4-2。

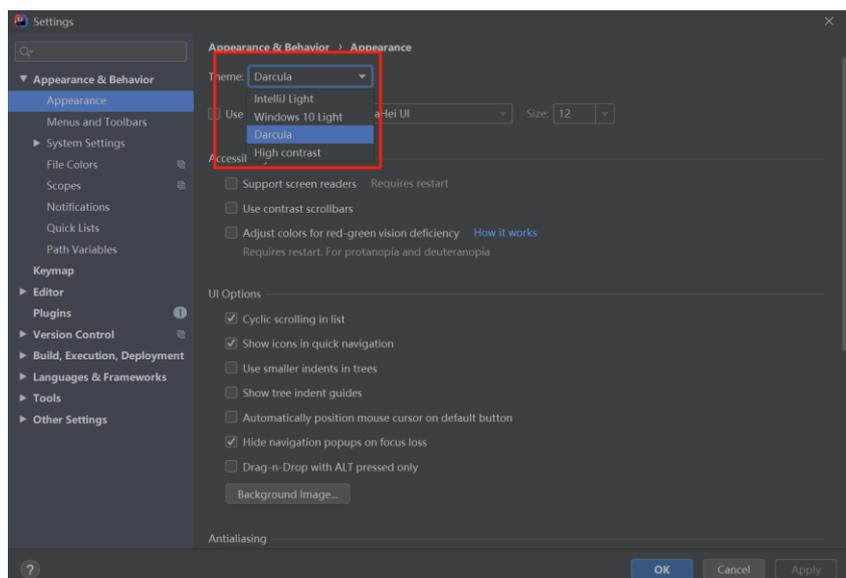


图 4-2

(3)主题字体和字体大小

依次进入 Settings → Appearance & Behavior → Appearance, 然后在右边设置主题, 字体以及字体大小即可。

如图 4-3

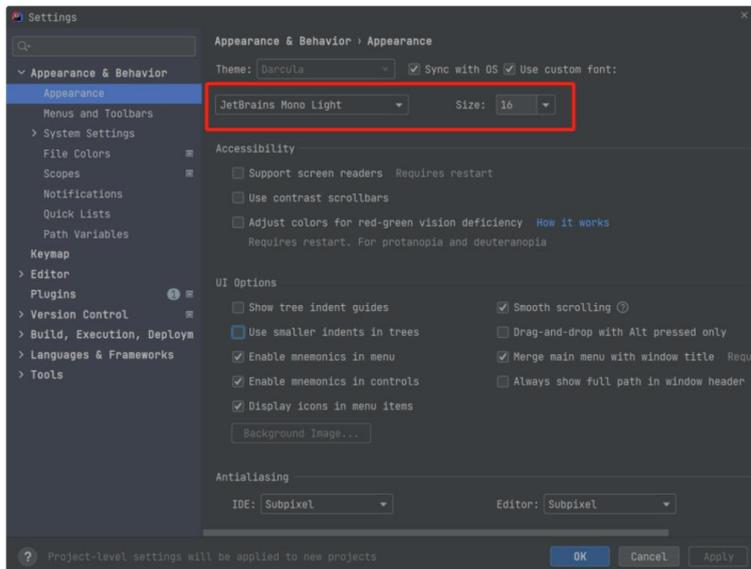


图 4-3

除上面设置编辑区字体大小的方法外, 我们还可以按住 Ctrl, 然后通过鼠标滚轮来实现字体大小的调整, 但是前提是你要先打开相关设置。设置入口如下, 依次进入 Settings → Editor → General, 然后勾选 Change font size with Ctrl+Mouse Wheel 即可。

如图 4-4。

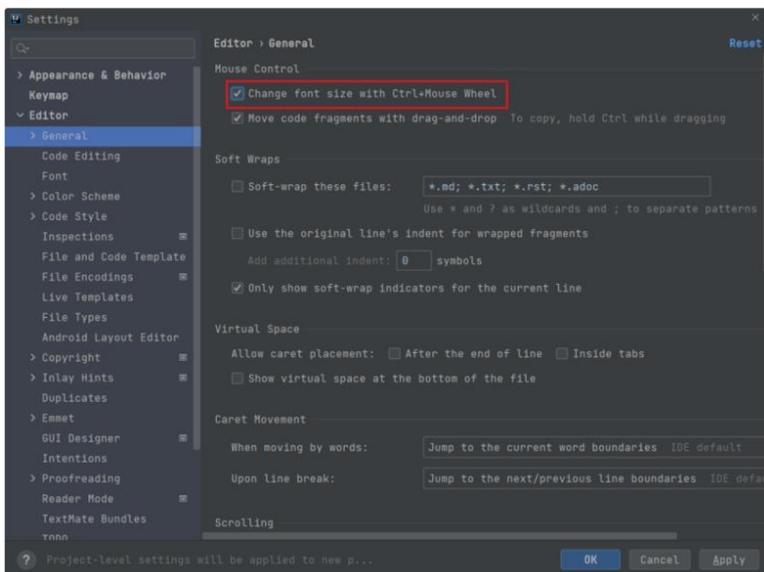


图 4-4

(4)控制台

一般情况下无需调整, 它会沿用编辑器一样的配置, 但如果需要自定义, 可以去如下入口: Settings → Editor → Color Scheme → Console Font, 然后勾选如下 Use color scheme font instead of the default 选项并进行设置即可。

如图 4-5。

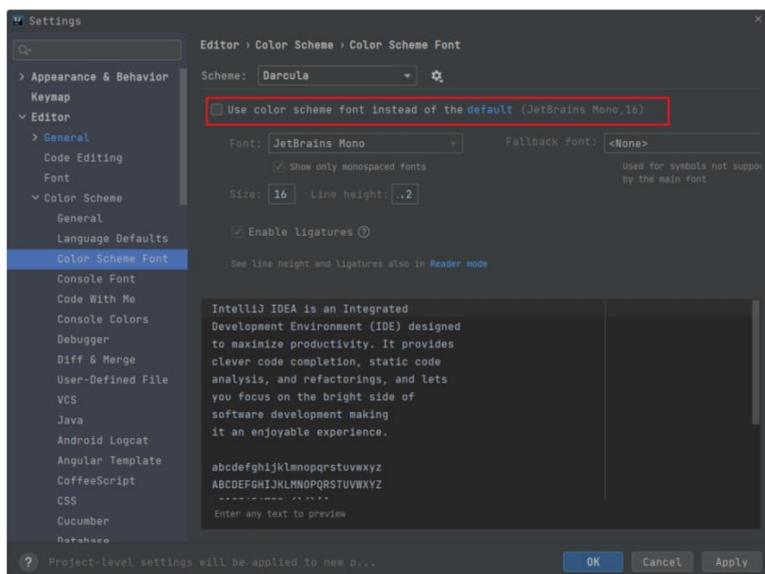


图 4-5

(5) 扁平化显示及空包隐藏

有时候，我们项目的包结构会变成如下样式，有时会造成一定的不便，此时是因为开启了 Flatten Packages 所导致。如图 4-6。

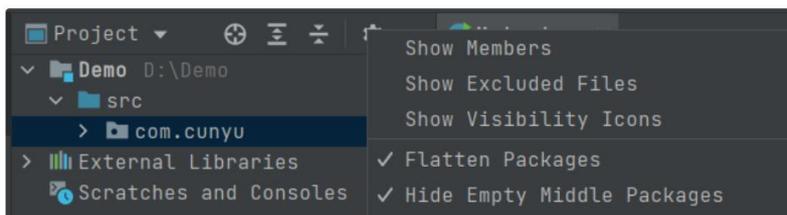


图 4-6

如果我们关闭此功能，则项目包结构如图 4-7 所示，其中 Compact Middle Packages 表示是否隐藏空包。

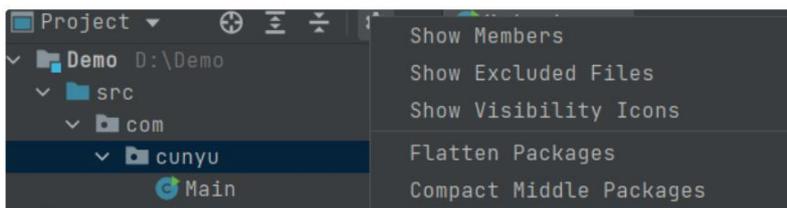


图 4-7

(6) 打开文件多行显示

平常使用 IDEA 时，一旦我们打开过多的文件时，默认会堆积在一行显示，就像浏览器打开了多个标签一样，此时需要通过右侧箭头筛选的方式来选择其他文件。为了解决这一问题，让我们打开的所有文件能以多行显示的方式方便查看。设置入口：依次打开 Settings → Editor → General → Editor Tabs，然后将 Show tabs in 勾选为 Multiples rows。

如图 4-8。

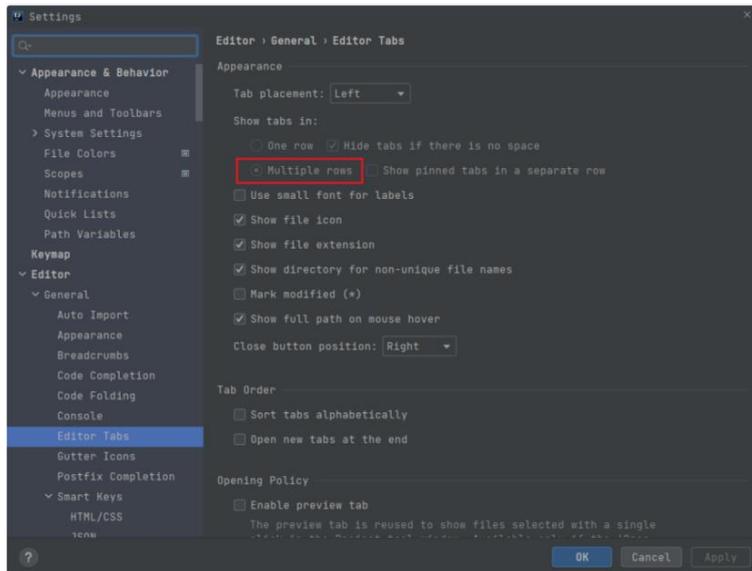


图 4-8

(7)行号和方法分隔符

日常为了方便定位 Bug 所在位置，我们通常需要打开代码所对应的行号，方便我们查找。另外，如果我们代码中方法较多时，最好是在每个方法之间加一个分隔符，方便我们知道一个方法的开始和结束。而这两个功能，IDEA 都是提供的。设置入口：依次打开 Settings → Editor → General → Appearance，然后勾选 Show line numbers 和 Show method separators。

如图 4-9。

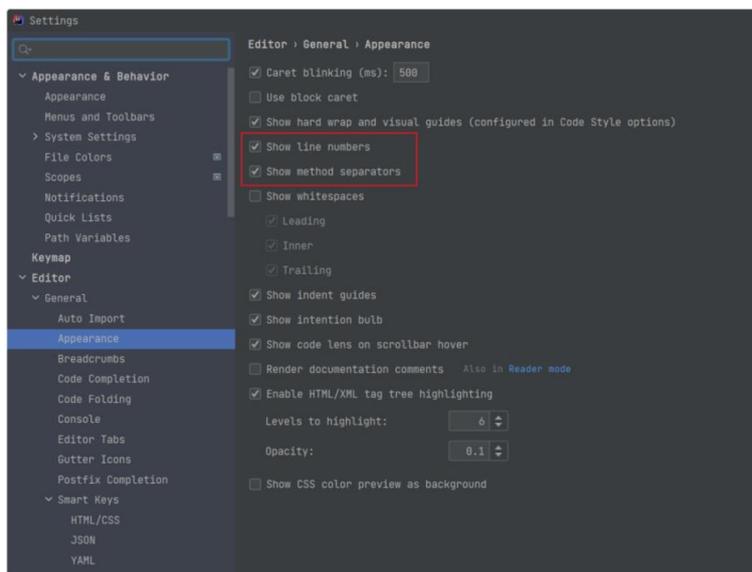


图 4-9

(8)自动导包

日常工作中，我们会经常用到第三方的包，每次都需要我们通过手动 (Alt + Enter) 的方式来导入，但实际上 IDEA 也是提供自动导包功能的，设置入口如下：Settings → Editor → General → Auto Import，将 Insert imports on paste 设置为 Always，再勾选 Add unambiguous imports on the fly 以及 Optimize imports on the fly 即可。

如图 4-10。

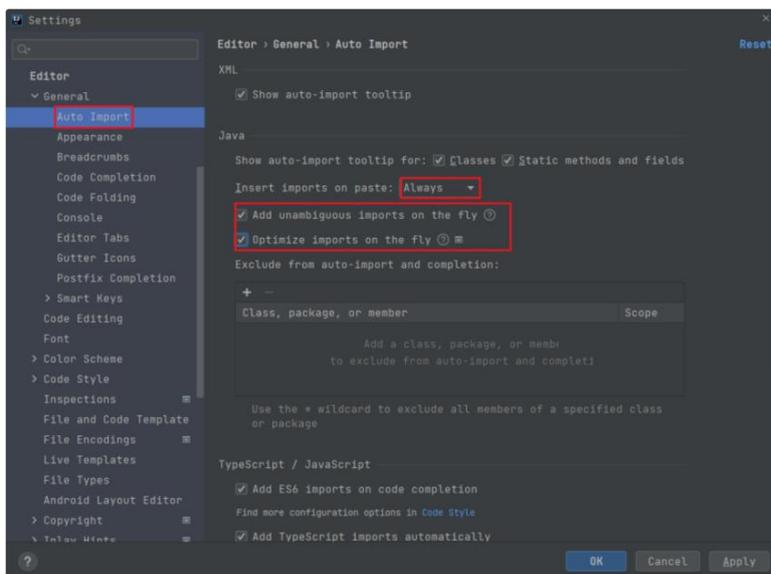


图 4-10

(9)忽略大小写

通常 IDEA 在导入类时大小写是全匹配的，这样一样就极大影响了我们编码的速度。此时我们就可以通过设置 IDEA，让其忽略大小写字母，提高效率。设置入口如下，依次进入 Settings → Editor → General → Code Completion，取消勾选 Match case。

如图 4-11。

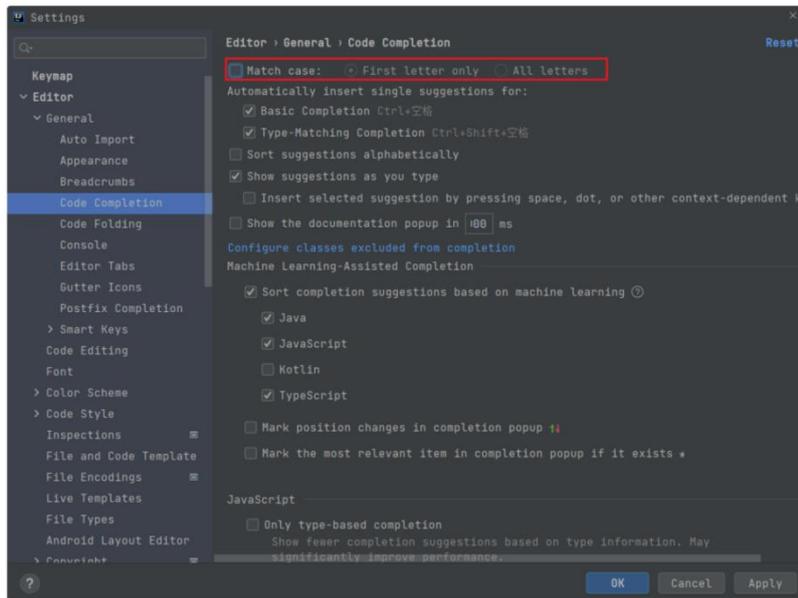


图 4-11

(10)自动编译

IDEA 默认是不会对代码自动编译的，需要我们手动配置。设置入口：依次打开 Settings → Build,Execution,Deployment → Compiler，然后勾选 Build project automatically 和 Compile independent modules in parallel。

如图 4-12。

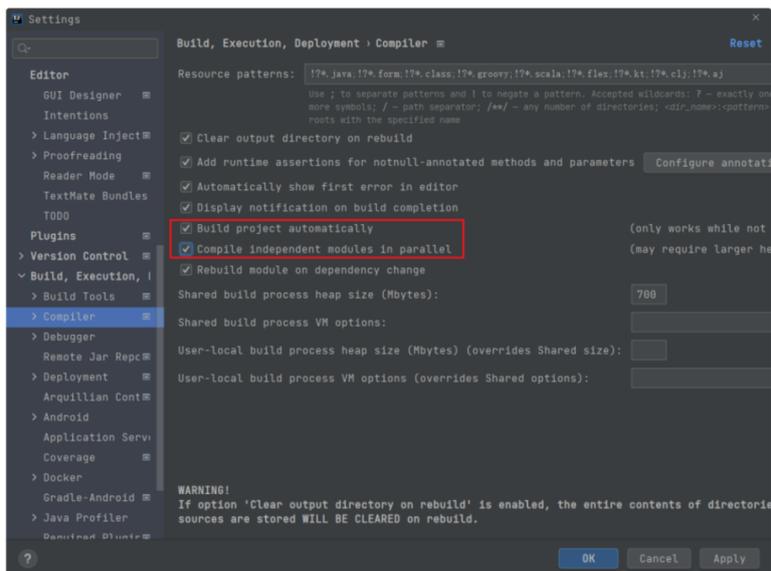


图 4-12

(11) 编译器设置

这里推荐使用 Javac 编译器，然后编译器版本最好保持和你本地使用的 JDK 版本相同。因为我本地使用的是 JDK11，所以这里版本我设置的是 11，这里最好根据你自己的实际情况进行设置。设置入口：依次打开 Settings → Build, Execution, Deployment → Compiler → Java Compiler，然后将 Use compiler 设置为 Javac，并将 Project bytecode version 设置为你本地使用的 JDK 的版本。

如图 4-13。

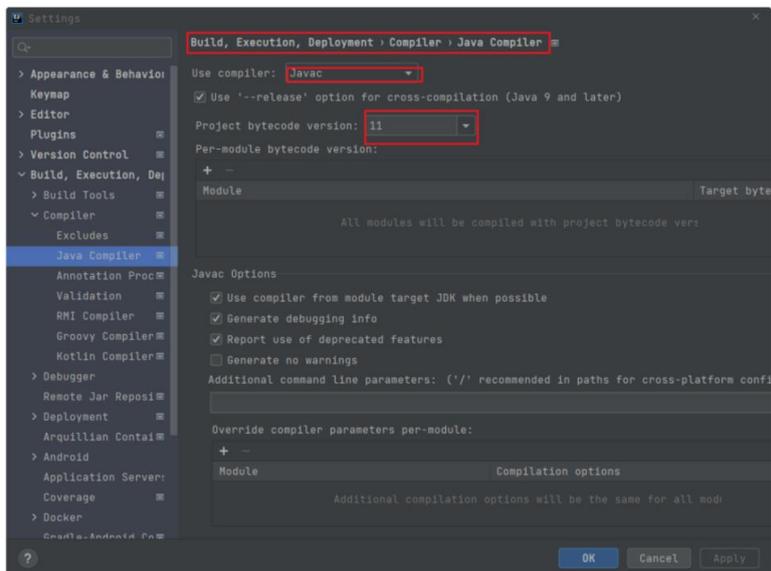


图 4-13

(12) Maven 设置

默认情况下，IDEA 是由绑定的 Maven 的，如果我们设置自己所使用的 Maven。需要设置的主要有三点：

- 1 Maven home path: 即自己本地的 Maven 所在目录；
- 2 User settings file: 即自己本地 Maven 环境下的 settings.xml 所在目录；

3 Local repository: 即自己本地的 Maven 仓库;

设置入口: 依次打开 Settings → Build, Execution, Deployment → Build Tools → Maven, 然后设置以上三点即可。

如图 4-14。

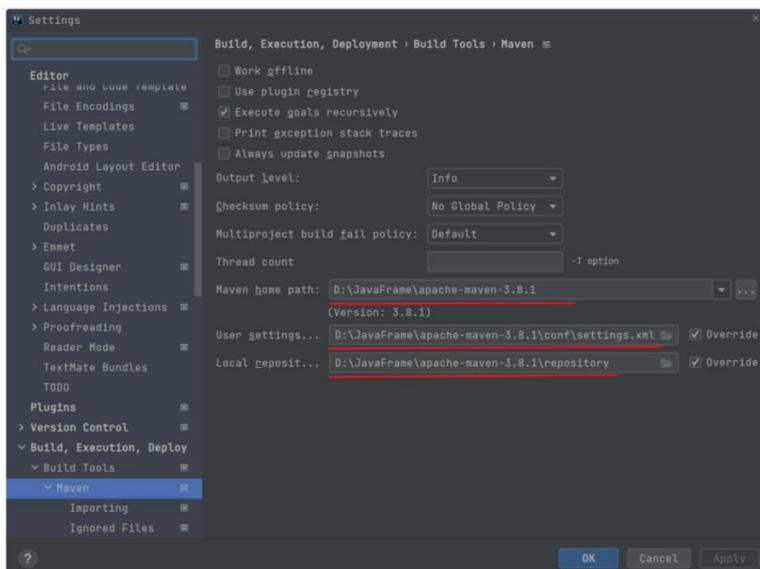


图 4-14

(13)类模板

当我们创建一个类时, 希望每个类的前边都已经一些注释的代码, 用于标识这是我们自己所写的代码。此时就需要到 Settings → Editor → File and Code Templates, 然后在右边的 Includes → File Header 右边的方框中, 填入如下模板。设置完成之后, 以后我们每次创建新的类文件时, IDEA 都会在文件的开头添加如下信息。

如图 4-15, 4-16。

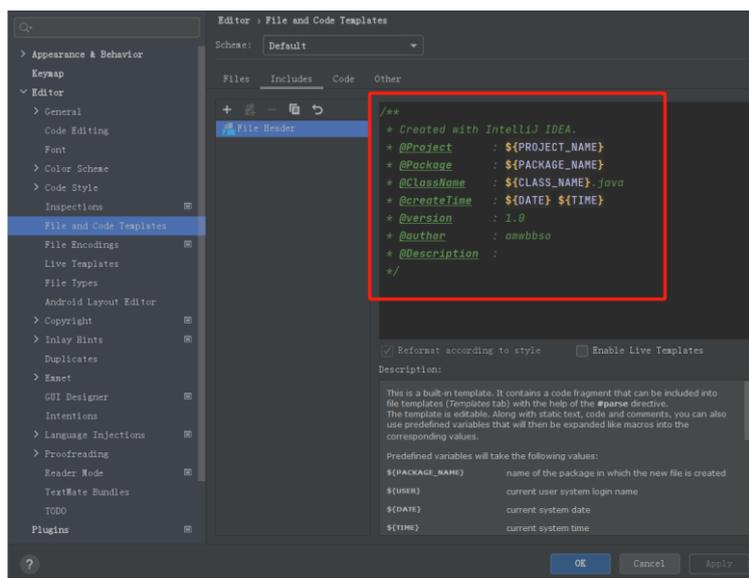


图 4-15

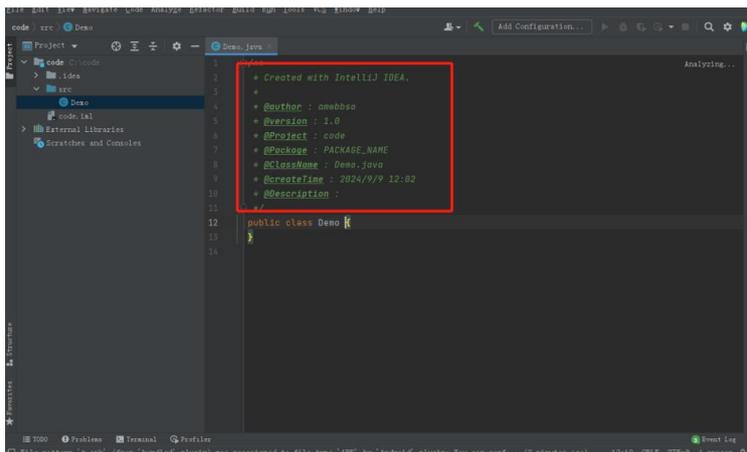


图 4-16

(14)导入导出设置

经过我们一番调教，IDEA 用起来就十分顺手了。但这时候问题就来了，要是我们换个系统或者电脑使用啥的，难道我们又得重新设置一遍么？当然不是了！IDEA 是支持导入导出我们的配置的，我们只需要将我们的设置导出，然后换电脑或者重装系统使用时进行导入即可。

导出

导出口：依次打开 File → Manage IDE Settings → Export Settings…。

如图 4-17。

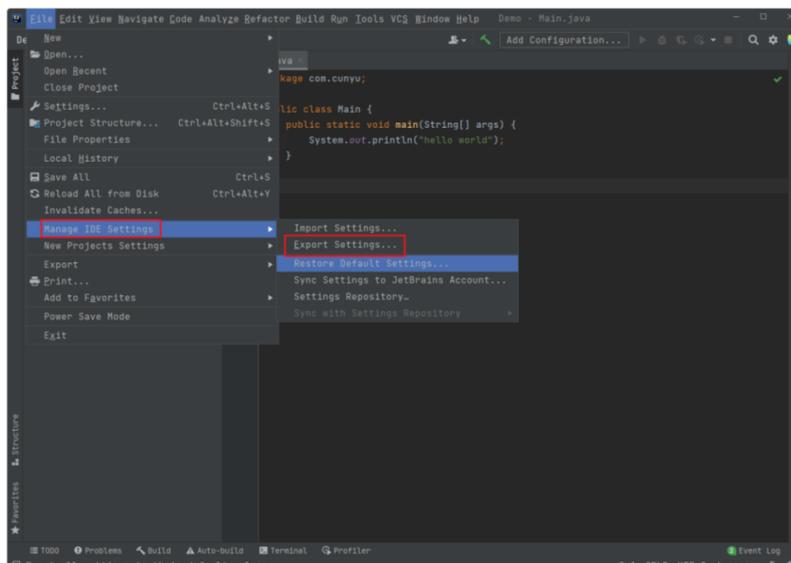


图 4-17

导入

导入入口，依次打开 File → Manage IDE Settings → Import Settings…。

如图 4-18。

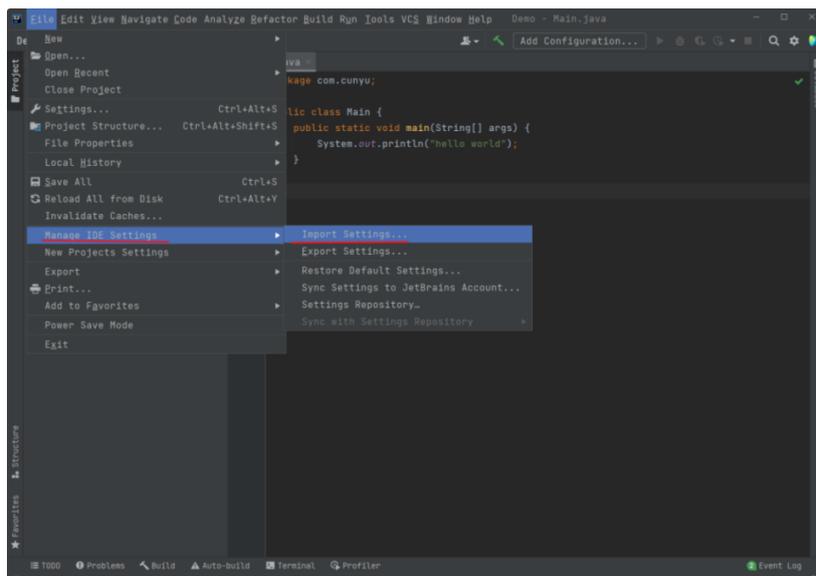


图 4-18

七、实验考核

本实验考核采用【实验随堂查】方式开展。

每个实验完成后，在实验课上通过现场演示的方式向实验指导教师进行汇报，并完成现场问答交流。

每个实验考核满分 100 分，其中实验成果汇报 60 分，现场提问交流 40 分。

实验考核流程：

- (1) 学生演示汇报实验内容的完成情况，实验指导老师现场打分。
- (2) 指导老师结合实验内容进行提问，每位学生提问 2-3 个问题，根据回答的情况现场打分。
- (3) 实验考核结束后，进行公布成绩。

八、创作说明

本实验指导书由河南中医药大学信息技术学院互联网技术教学团队与河南方和信息科技有限公司联合创作。

作者：黄子杰（河南方和信息科技有限公司）

审核：阮晓龙（河南中医药大学信息技术学院）

排版：刘昊鑫（河南中医药大学智能医学工程专业 2023 级）