



链滴

JAVAFX(一) java Module 模块化简介

作者: [xiaodaojava](#)

原文链接: <https://ld246.com/article/1588561865796>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



本文主要知识点: 1. 基于jdk14的java模块化, 2. jlink的使用

前言

看过小刀最近朋友圈的小伙伴应该都知道, 最近小刀在写一个桌面的工具包, 可以连k8s, 可以做图床等, 其中用到的技术栈就是 jdk14+javaFX.

JDK14(Module)+javaFx

JDK14(Module)

选用JDK14, 主要是因为其中的jpackage命令, 这个命令可以帮我们把java程序打包成win中的exe, 包成mac 中的dmg, pkg. 这样使用者就不需要再单独安装jdk/jre了, 加上java9之后java的module化, 可以最大程度上的精简jre, 虽然是精简了,但实际上,大家可以看到, 每次我在群里面发的包,还是接近10 MB了.

javaFX

java身为一个跨平台语言,我们对其更多直观的感受是在win/mac 上写了一个springboot的程序, 然后到win/linux服务器上去运行. 不仅仅是服务端应用, 桌面端应用也可以这样跨平台. 用上面用jpackag 打包之后的应用, 在不同的平台上打的包, 可以在不同的平台上运行. 虽然在UI/字符显示上会有些许差别但是差别不怎么大. 可以通过接口-不同平台实现类的方式进行处理.

系统配置要求

本文及后续文章系统要求:

jdk14(为了使用jpackage命令),

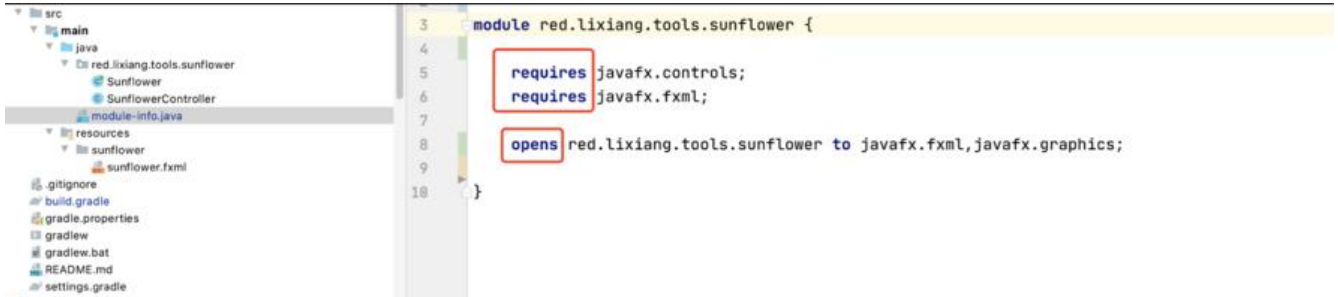
idea社区版2020.1或以上(这个版本开始支持jdk14),

gradle 6.3及以上(gradle 6.3开始支持jdk14)

小刀的交流群(一个人踩坑? 踩的想砸电脑.....)

java Module

模块化和非模块化的区别,在代码的体现上就是module-info.java的区别



如上图所示: 该文件的关键在于三个关键字(requires, opens, export), 因为我们这个项目已经是最上模块, 不需要为别的模块提供方法, 所以没有用到export关键字。

requires: 我们这个模块(模块名: red.lixiang.tools.sunflower) 需要javafx.controls和javafx.fxml模块

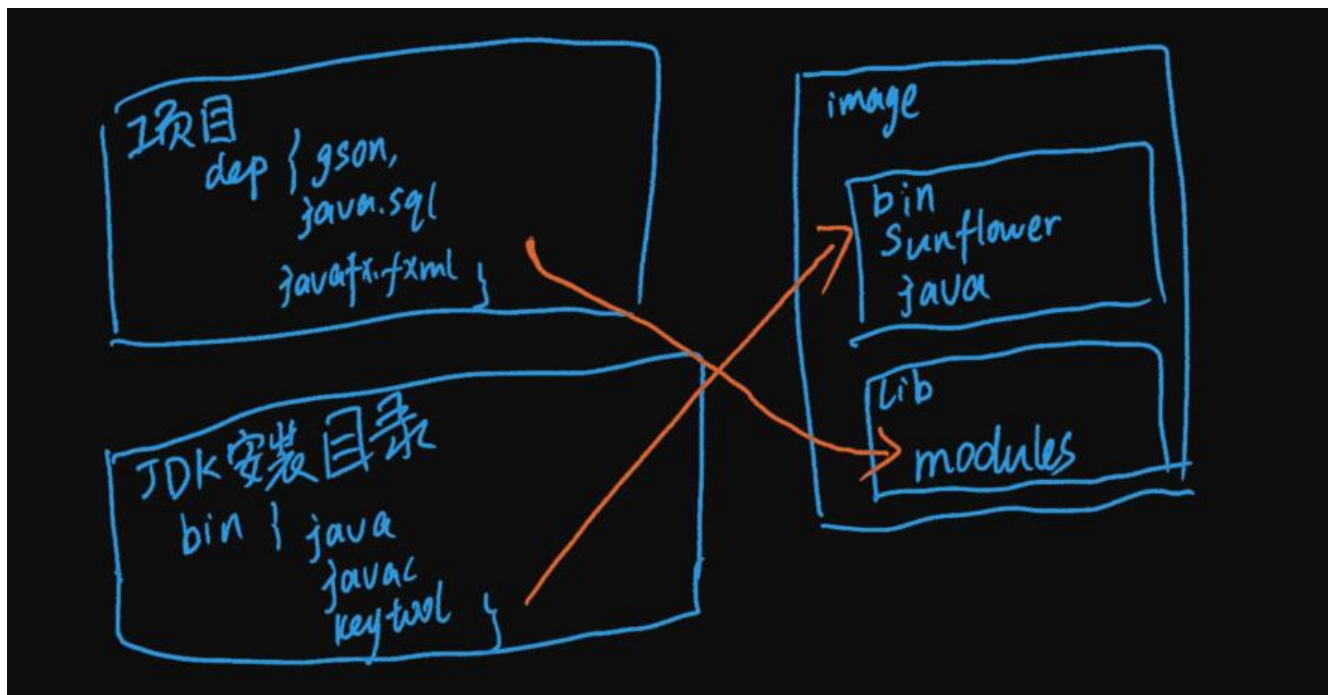
opens: 主要是反射使用,javafx.fxml可以反射red.lixiang.tools.sunflower中的类,然后可以对应实例化或者调用实例中的方法

Jlink

在java 模块化中, jlink命令是核心,其作用就是把jdk中我们需要模块抽出来, 把我们自己引入的模块给包一起,然后生成一个image(可运行的镜像), 这个有gradle插件帮我们完成. 我们只需要知道这个命令用即可,生成之后的文件如下:



我们可以在lib文件夹中看到modules这个文件, 就是合并之后的模块依赖, 这时候, 要运行这个程序, 只要运行bin目录下的Sunflower/Sunflower.bat 就可以了.



jlink的作用,我们可以用上图表示, 把项目的requires的打包到lib中的modules中, 把jdk中相关的命令包到 image文件夹中.

后语

大家对jdk14, jlink, module有什么想法? 欢迎一起讨论~~