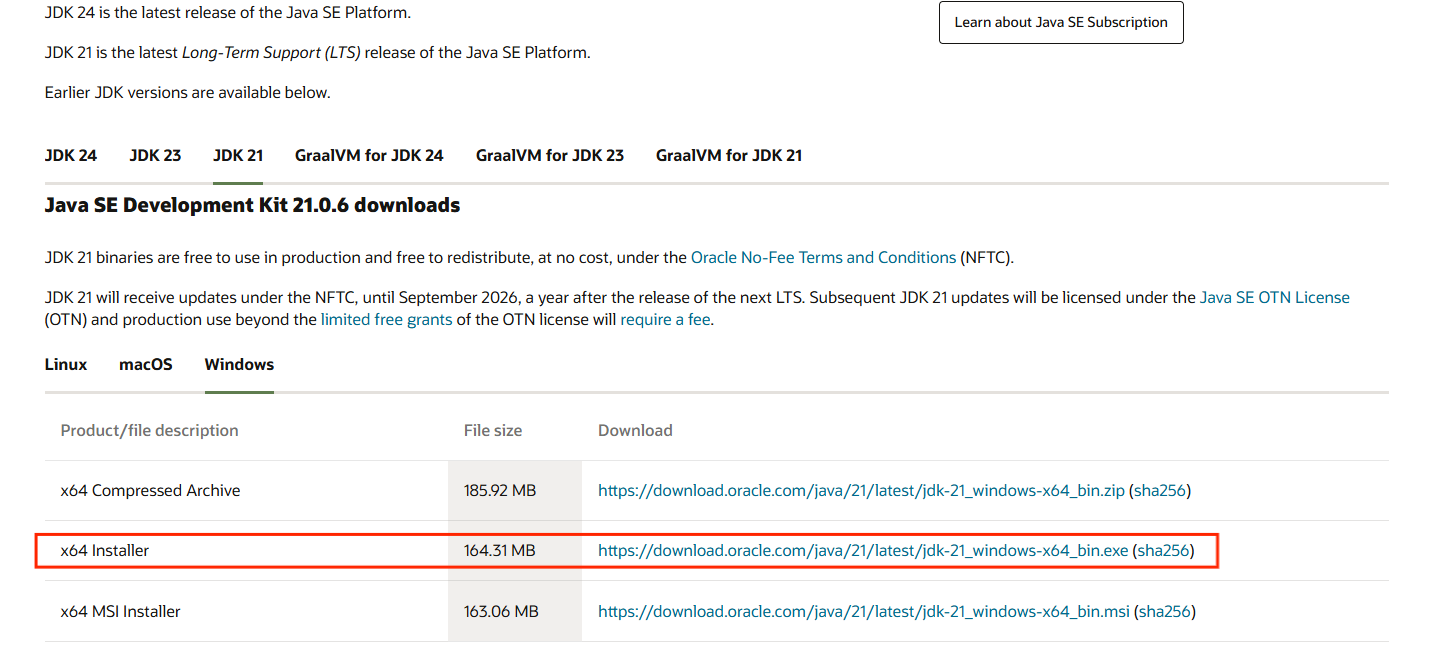
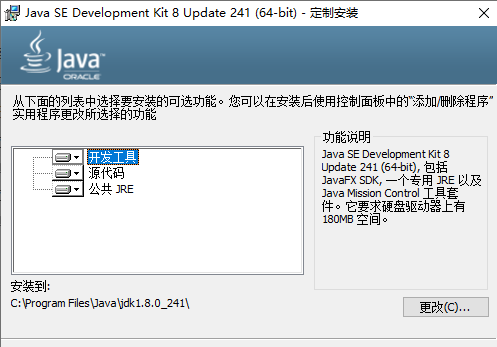
1. **配置环境**

1）下载JDK

进入Oracle官网：<https://www.oracle.com/java/technologies/downloads/#jdk21-windows>



一路next点到底即可（**过程中出现安装JRE的弹窗请点同意**）

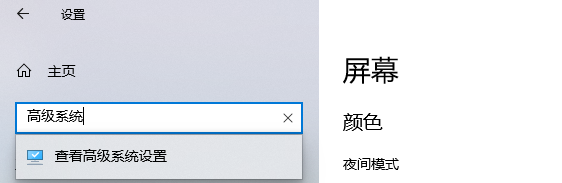


现在JDK安装包可以自动配置环境变量，如果不行就按下面手动配置一下（没有问题直接看下一步就行）

A.找到并记住JDK安装目录，如：C:\Program Files\Java\jdkxxx.xxx\

B.右键点我的电脑->属性->高级系统设置->高级->环境变量

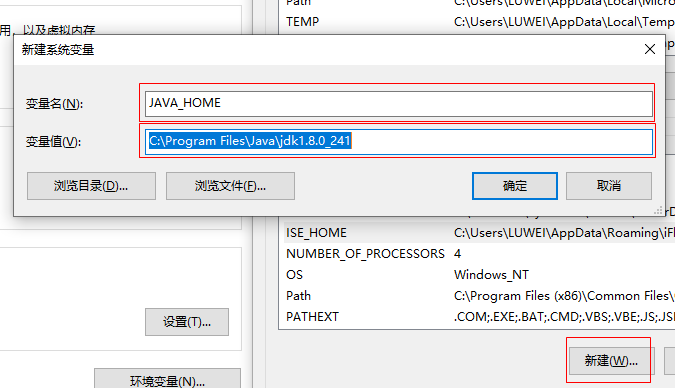
或者如下图所示直接在设置里搜索



点击环境变量

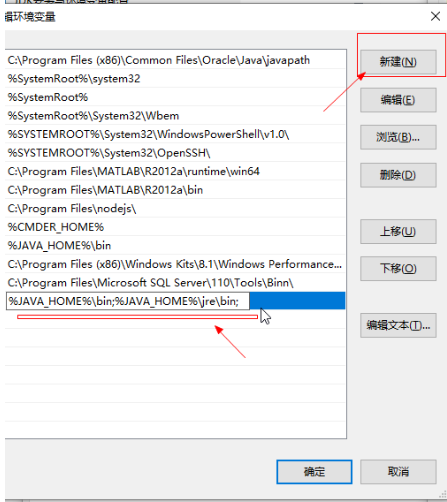


在系统变量中点新建，弹出的窗口中，变量名写JAVA\_HOME，变量值就是刚才的安装目录



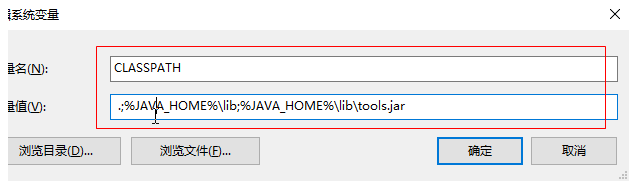
C.在同一个地方找到Path变量，点编辑，然后新建，输入

%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;



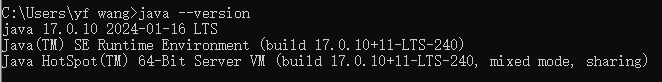
1. 再在系统变量中新建一个CLASSPATH变量，变量值为

.;%JAVA\_HOME%\lib;%JAVA\_HOME%\lib\tools.jar



到这里环境变量应该就配置完了

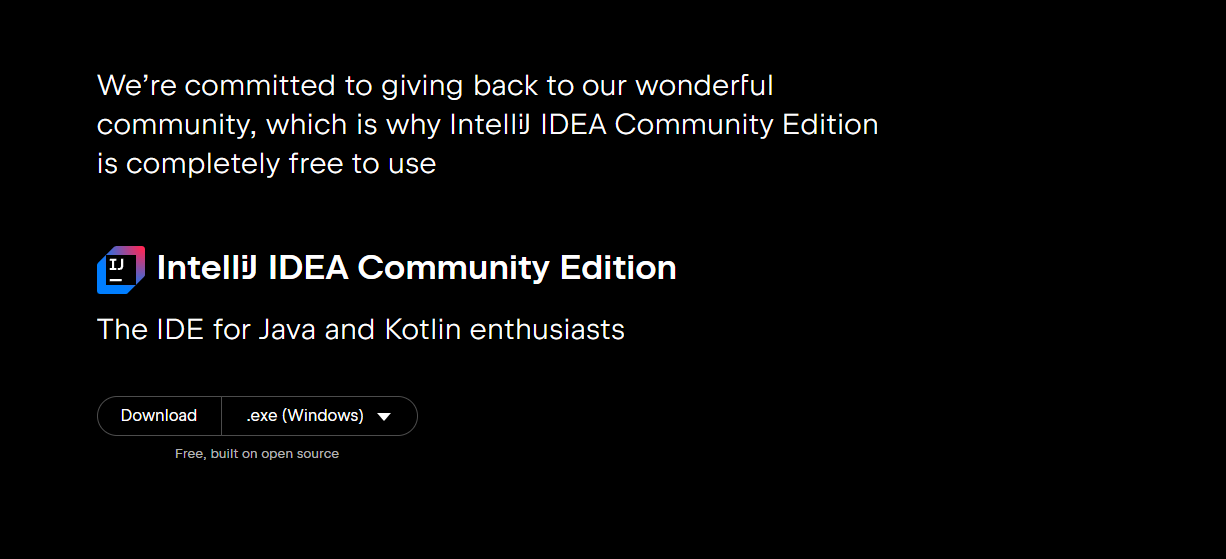
按Win + R打开cmd，输入java --version



出现上述说明安装成功

1. 下载IDEA

进入官网：<https://www.jetbrains.com/idea/download/?section=windows>

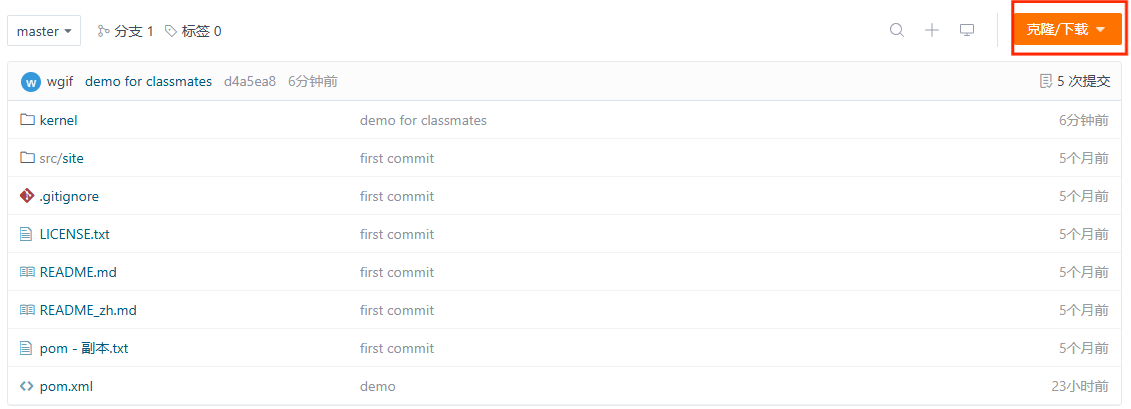


下拉找到community版本，下载并安装。

3）下载CrowdOS

去gitee下载CrowdOS的demo：<https://gitee.com/wang-yifan10086/crowd-os_-demo>

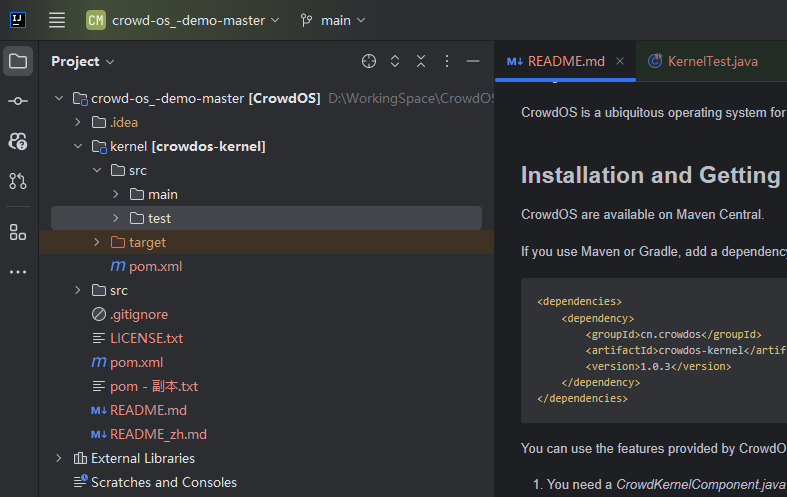
（如果熟悉git使用的话可以直接克隆）



点击克隆/下载，然后下载zip到本地



解压之后，用IDEA打开即可！

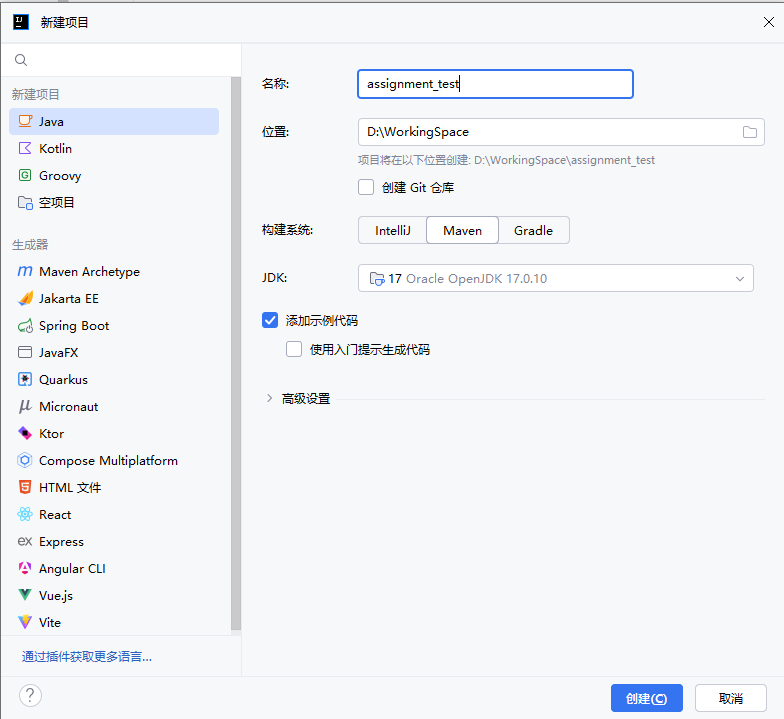


1. **简单任务——熟悉java与面向对象**

介绍：让我们通过一个简单的任务来重新熟悉一下java！我们先自己简单模拟一下群智感知中任务分配的基本过程！

要求：

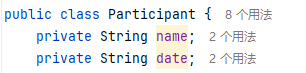
1. 在IDEA新建一个java项目



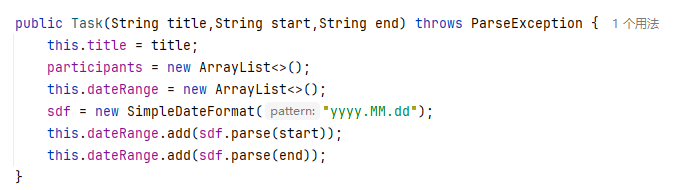
1. 创建Task类，其中包含任务名称、参与者、任务进行时间段三个属性。

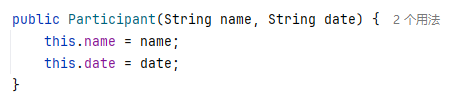


1. 创建Participant类，其中包含名字、空闲时间两个属性，以及toString方法（用来输出名字）。



1. 为两个类创建构造方法，初始化属性。





1. 为Task类设计canAssinTo()方法用来判断参与者能否参与任务（空闲时间满足任务进行时间即可）；addParticipant()方法将满足条件参与者添加到任务的参与者列表中。

提示：用date.compareTo方法对日期进行比较

1. 为Task类设计getParticipants方法，返回参与者列表（或者字符串，随便你喽）。
2. 创建一个主函数，在其中创建新的参与者与任务，进行任务分配吧！



1. **任务2——熟悉CrowdOS**

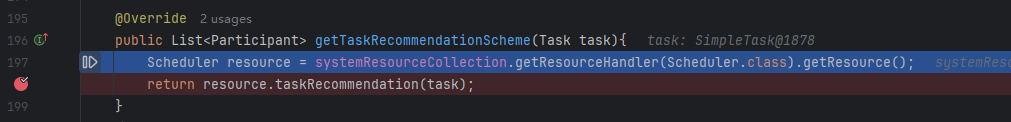
介绍：刚刚我们自己模拟了简单的任务分配流程，接下来看看在CrowdOS系统中，任务分配是如何进行的吧！

要求：

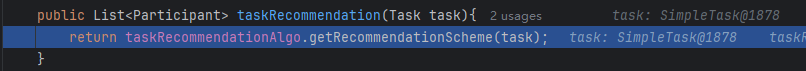
1. 用IDEA打开CrowdOS\_Demo
2. 依次进入kernel\src\test\ cn.crowdos.kernel\KernelTest
3. 模板中已经创建好了2个TimeParticipant并注册到kernel，并且耶创建并注册了一个简单的任务。请大家依葫芦画瓢的再创建5个TimeParticipant、1个SimpleTask并注册到kernel，然后运行getTaskRecommendationScheme方法，看看任务分配情况。
4. 大家自行打上断点，跟踪程序运行过程，了解CrowdOS分配任务的初步逻辑，为下一个任务做好准备。

提示：分别在下面几个地方打上断点

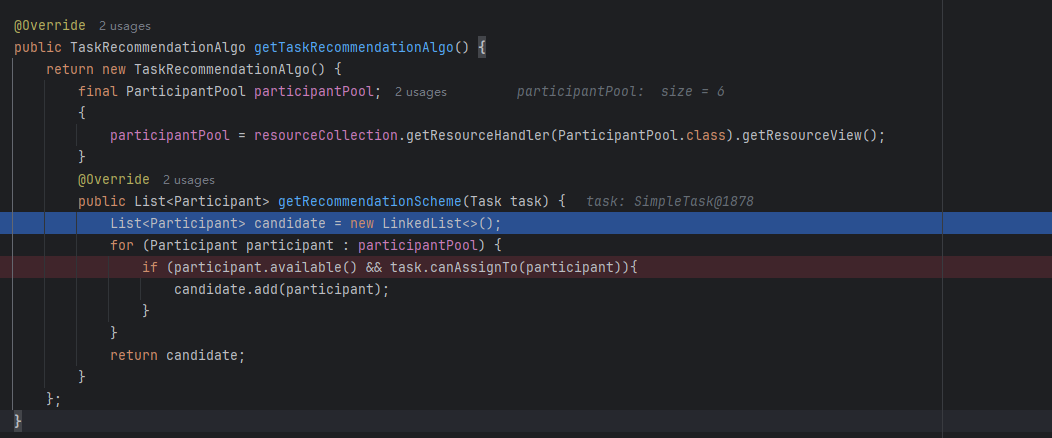
1. Kernel中的getTaskRecommendationScheme方法



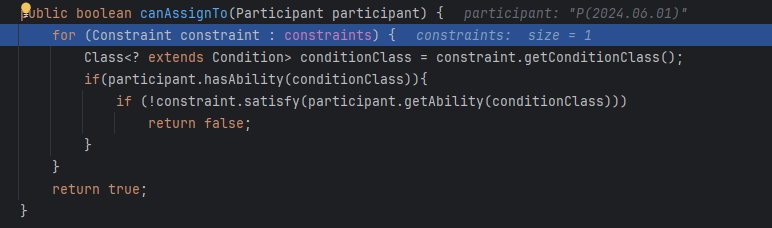
1. Scheduler中的taskRecommendation方法



1. AlgoFactoryAdapter中的getRecommendationScheme方法



1. AbstractTask中的canAssignTo方法



1. 探究问题：

如图所示，为什么任务要求时间是”2024.6.1-2024.6.2”,推荐结果只有”P(2024.6.1)”1个参与者？试着找出对应代码段！



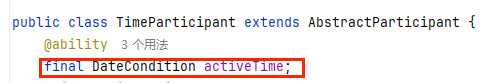


1. **任务3——限制条件为POI的任务分配**

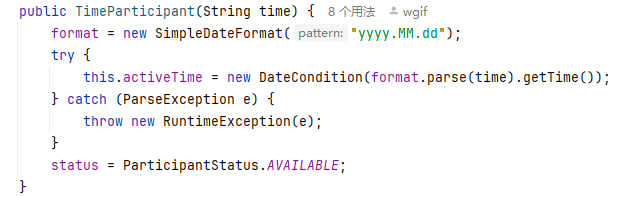
介绍：群智感知中，有很多任务都是基于POI（position of interest）进行任务分配的，请大家也试着创建一个简单的任务，规定一个任务范围，然后看看CrowdOS能不能将任务分配到范围内的参与者吧！

要求：

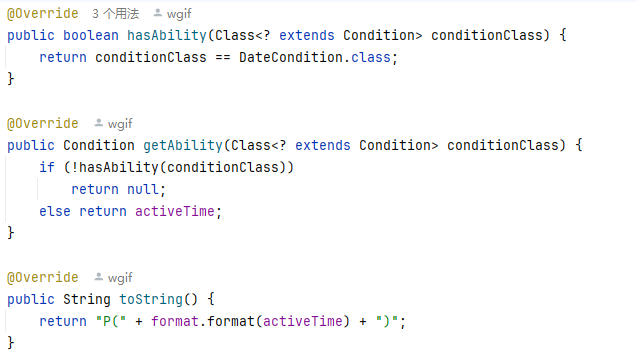
1. 在Kernel\src\main\java\cn.crowdos.kernel\目录下中找到Condition和Constraint接口的定义，并思考Condition与Constraint的关系
2. 在kernel\src\test\ cn.crowdos.kernel\common中，参照TimeParticipant的形式创建新的参与者类PoiParticipant，思考并确定能力应该是什么。



1. 参照TimeParticipant的形式，在构造方法中，对能力属性和参与者状态进行初始化



1. 参照TimeParticipant的形式，覆写父类的三个方法



1. 参照KernelTest的形式，新建一个ParticipantTest,同样也是创建参与者，创建任务，然后进行任务分配吧！
2. **附加任务1——如果你成功完成上面的任务**

如果你觉得以上都是小菜一碟，那不妨试着自己创建一个任务，自定义更复杂constraint（比如同时限制时间和地点），并试着进行新的任务分配！

1. **附件任务2**

**不会吧**！您竟然还觉得简单！那不如告诉你，其实我们之前调用的getTaskRecommendationScheme()方法并不是真正的任务分配，它只是判断参与者的condition是否满足任务的constraint。

您可以试着在注册任务之后，加入kernel.algoSelect("GGA\_I");来设定任务分配算法，并调用getTaskAssignmentScheme()方法，这个方法会用算法工厂中的任务分配算法进行真正的任务分配，您可以试着看看算法工厂的运行逻辑（我们还在GGA\_I算法中留下了一个bug，看看能不能找到！）