

实验 3 对象运行时行为的调试与性能分析

1.实验目的

(1) 通过对小段代码调试的深入分析，了解面向对象程序运行时涉及的机制，从而对面向对象程序的行为有更深刻的理解。

(2) 用 TPTP、JProfiler 等插件进行 JAVA 程序性能分析。

2. 知识要点

JVM 中有几个比较重要的内存区域，在 java 类的生命周期中扮演着比较重要的角色：

(1) 方法区：在 java 的虚拟机中有一块专门用来存放已经加载的类信息、常量、静态变量以及方法代码的内存区域，即方法区。

(2) 常量池：常量池是方法区的一部分，主要用来存放常量和类中的符号引用等信息。

(3) 堆区 (heap)：用于存放类的对象实例。

(4) 栈区 (stack)：也叫 java 虚拟机栈，是由一个一个的栈帧组成的后进先出的栈式结构，栈帧中存放方法运行时产生的局部变量、方法出口等信息。当调用一个方法时，虚拟机栈中就会创建一个栈帧存放这些数据，当方法调用完成时，栈帧消失，如果方法中调用了其他方法，则继续在栈顶创建新的栈帧。

2.1 理解 JAVA 中 class 的加载

(1) JVM 在程序运行过程中根据需要动态加载相应的类，能够按照程序的要求

创建相应的对象。

(2) 把相应类的.class 文件(可执行 bytecode)加载进内存。

2.2 对象初始化

(1) JVM 在首次加载 Java 类时，会对静态初始化块、静态成员变量、静态方法进行一次初始化。

(2) 只有在调用 new 方法时才会创建类的实例。

(3) 类实例创建过程：按照父子继承关系进行初始化。

2.3 方法调用

(1) 静态调用：类方法从 JVM 方法区调用，比如 static, final, private 等。

(2) 动态调用：每调用一个方法，当前线程创建一个栈帧，然后放入当前 stack 中，每个栈帧都包括局部变量区，操作数栈和栈帧变量区。

2.4 垃圾回收

(1) 分配内存。

(2) 保证所有正在被引用的对象还存在于内存中。

(3) 回收执行代码已经不再引用的对象所占的内存。

3. 工具讲解

3.1 使用 JAVA Profiling 工具观察和优化 JAVA 程序的性能

为了能准确地获得程序的性能信息，需要使用各种辅助工具。这些系统性能分析工具对性能瓶颈定位、系统故障排查都很有帮助。这类辅助工具也非常多，最为直接的有一些命令行工具，就在 Bin 目录下，例如 jstack\jinfo 等，还有一些

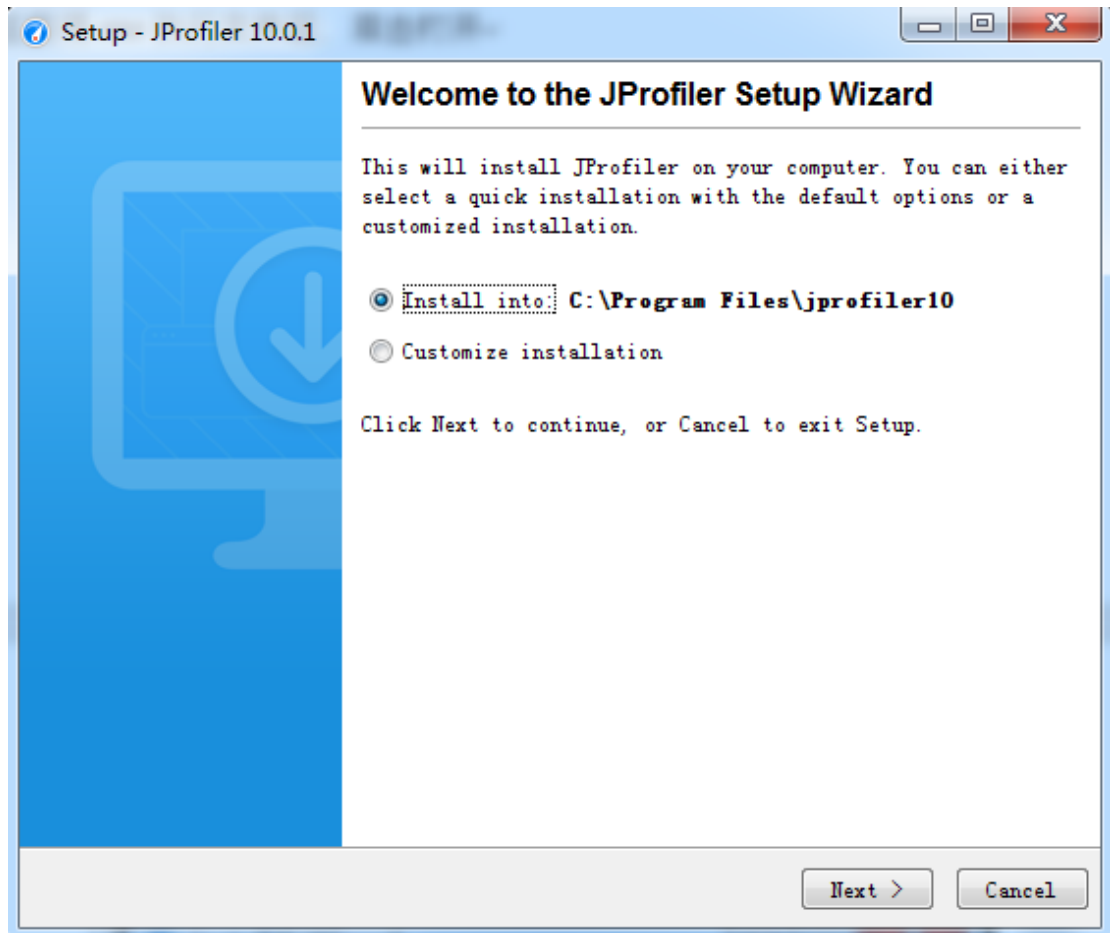
比较方便的有图形界面的，例如 JDK 自带的获取 Java 程序堆信息、线程信息的相关工具 jconsole、Visual VM 等等，也有一些需要单独下载安装的插件如 MAT、JProfiler、TPTP 等。

3.2 JProfiler 安装 (以 Eclipse 为例，IntelliJ IDEA 与 JProfiler 集成参见网址 <https://www.cnblogs.com/zhangyaxiao/p/6678385.html>)

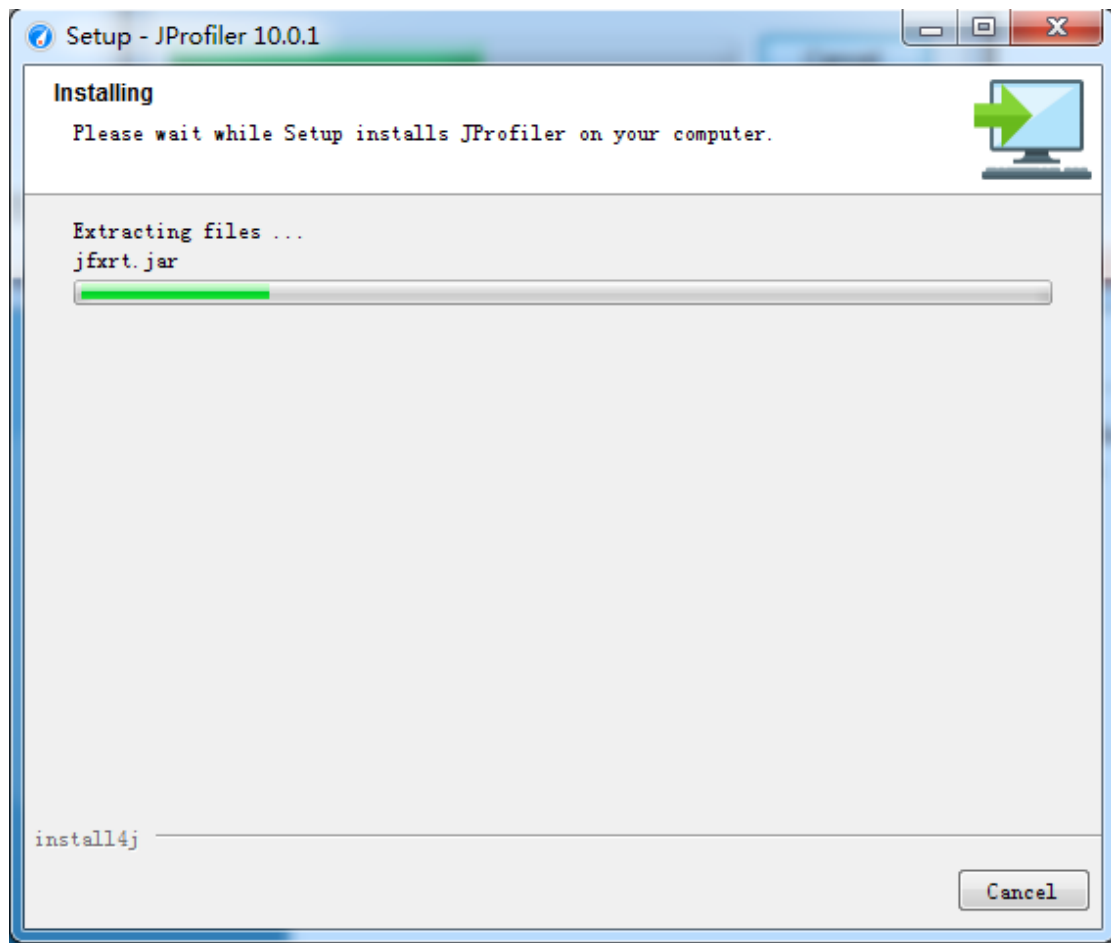
下载 JProfiler 安装文件 (<https://www.ej-technologies.com/download/jprofiler/files>) 后，双击打开。以下安装说明以 Eclipse 4.6.1 和 JProfiler10.0.1 为例，实际安装时请根据自己的版本信息进行选择。



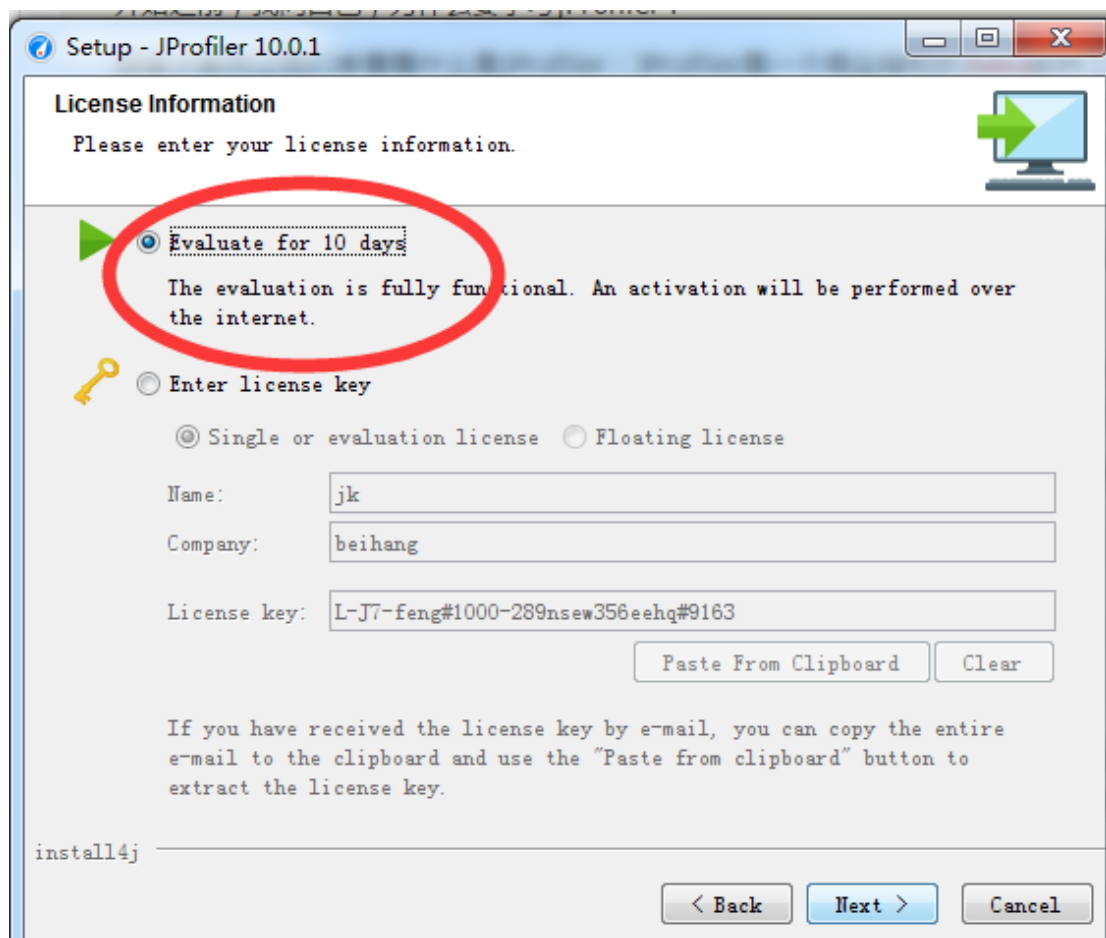
完成后选择默认安装方式。



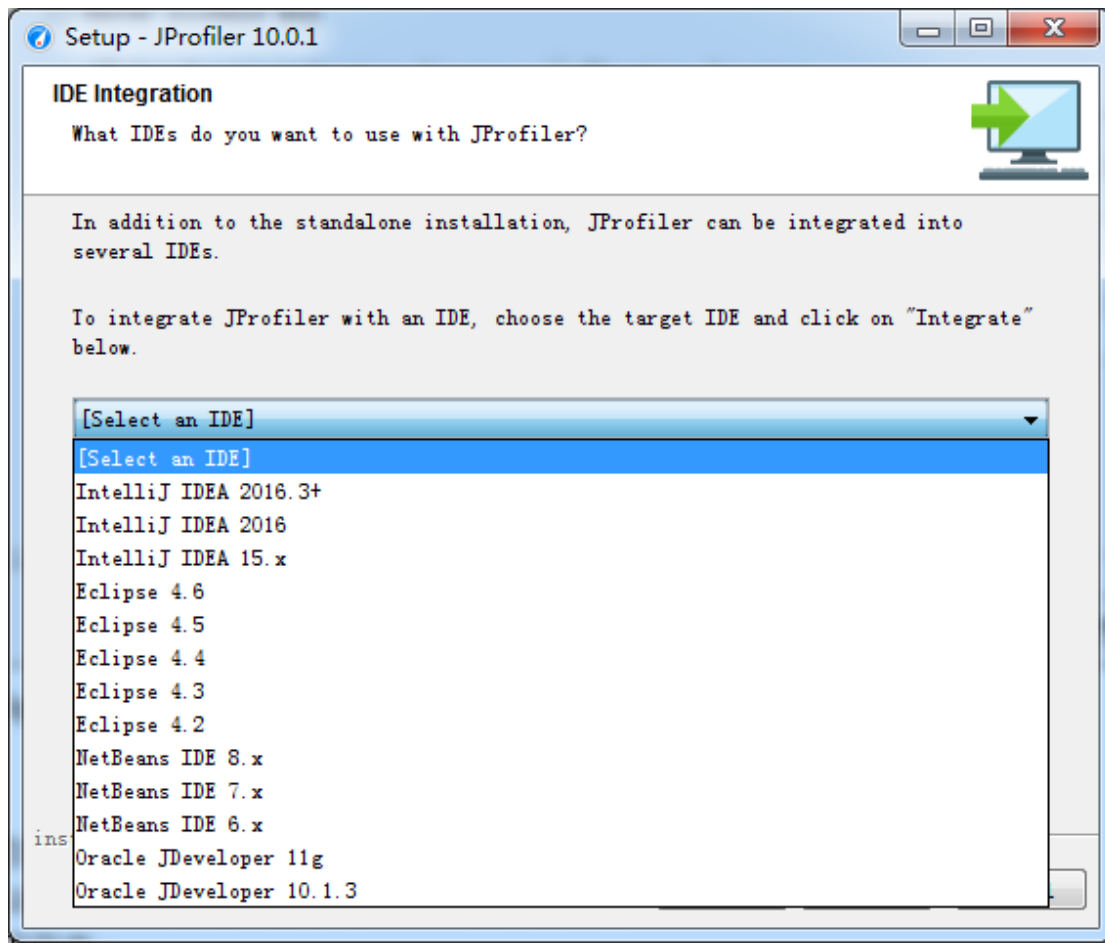
选择 Next 后等待安装完成



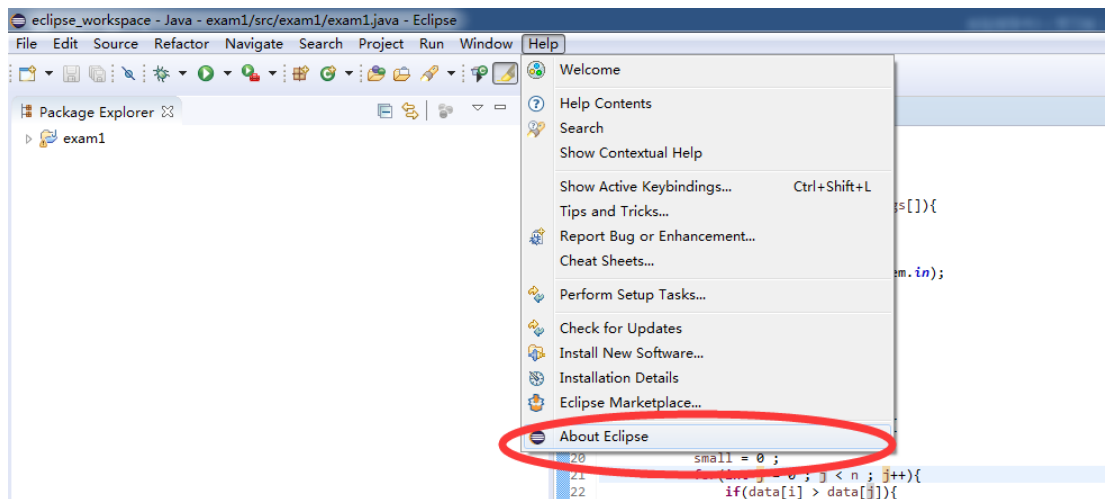
选择 10 天免费试用版，获取 license key。

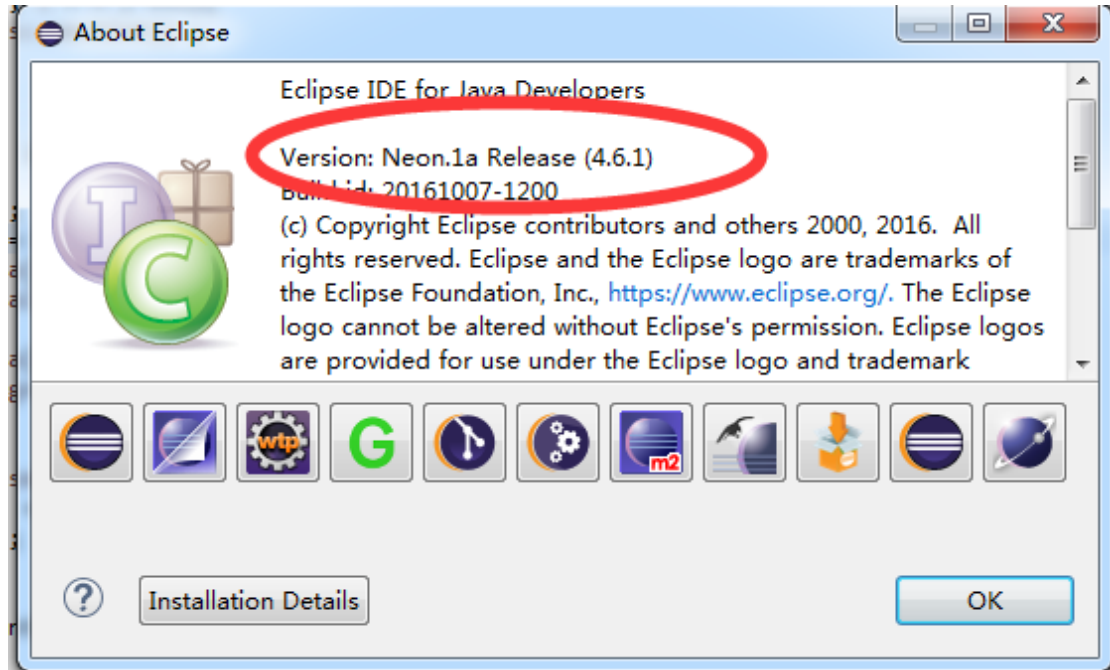


选择 Eclipse 版本

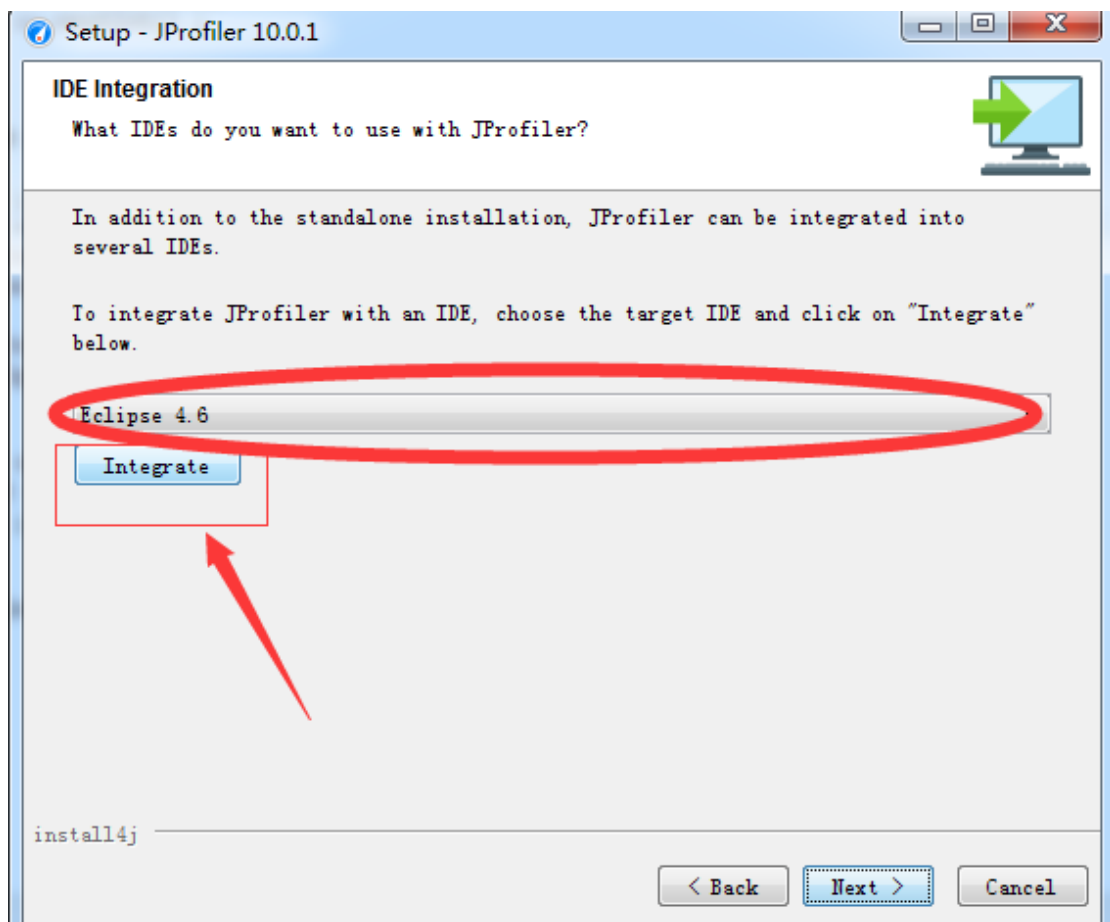


Eclipse 版本可以在如下页面进行查看



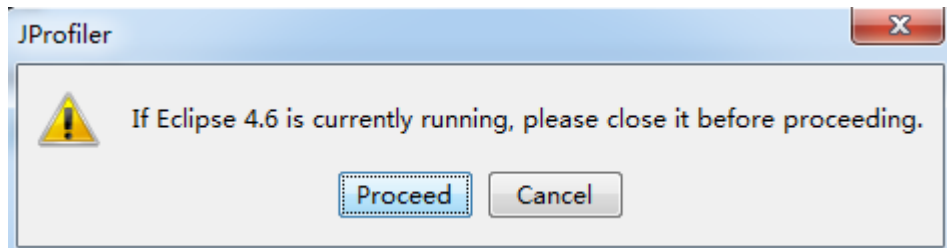


选择 4.6.1 (根据实际版本情况进行选择)

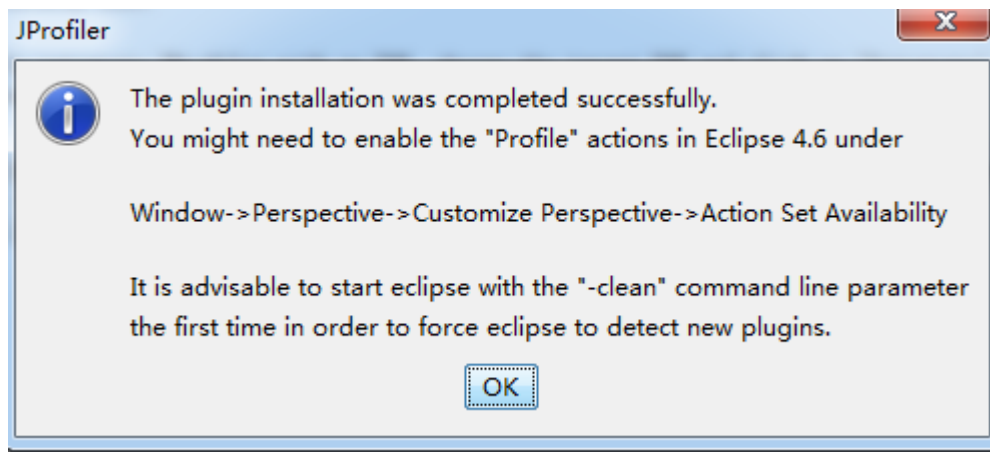


选择当前设备的 Eclipse 安装文件夹

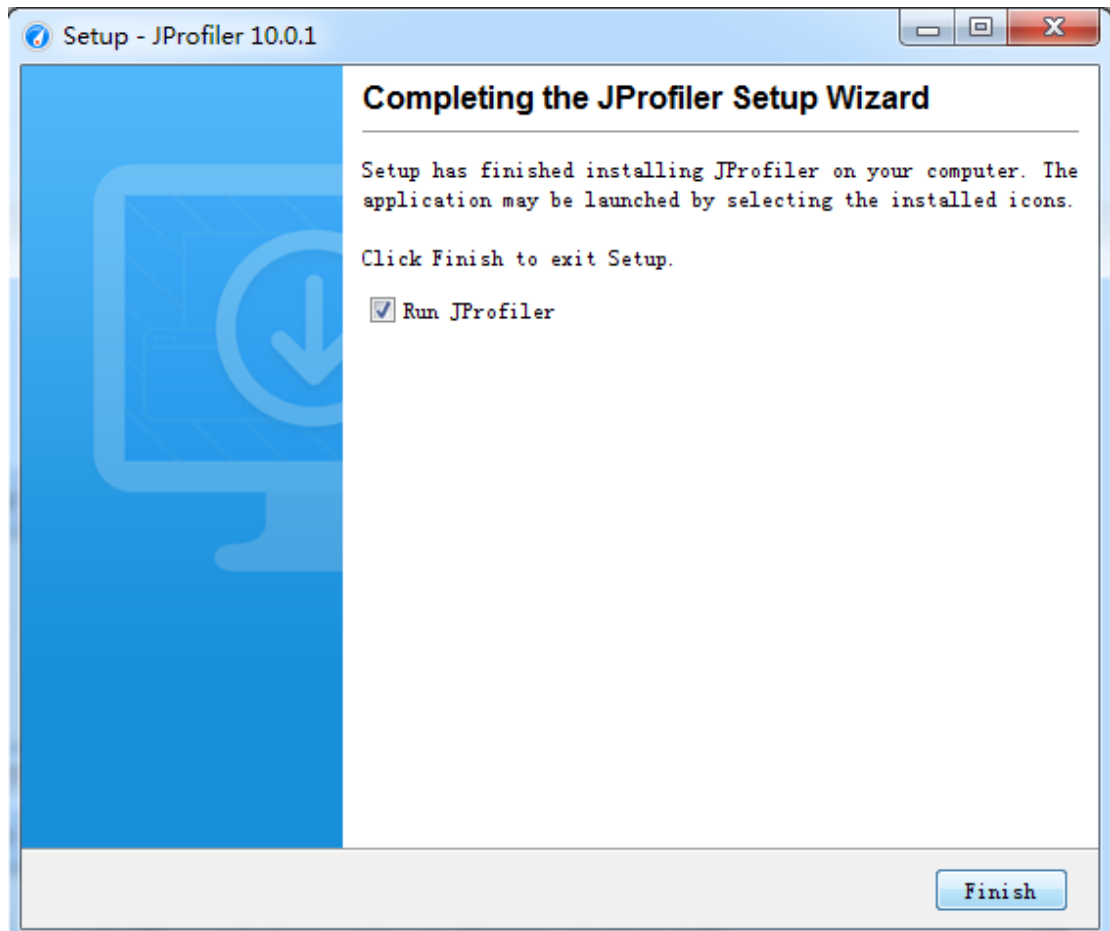
注意 : (此时应关闭正在运行的 Eclipse)



出现如下 tip 表示安装成功 (请根据提示信息进行操作):



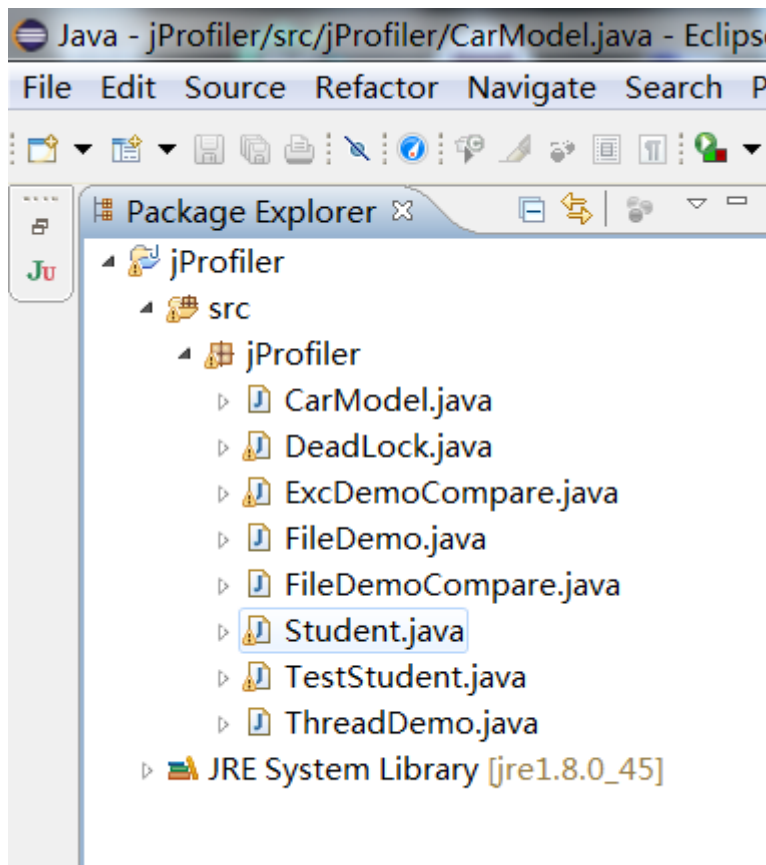
点选 OK→next



安装完成。

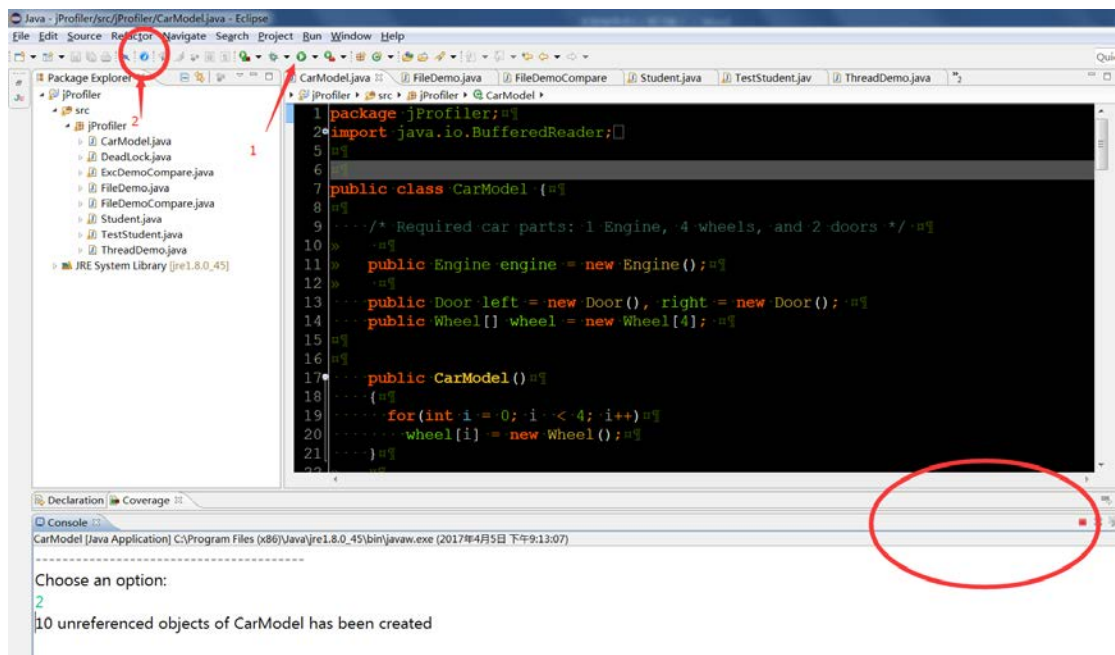
3.3 使用 JProfiler

- 新建工程

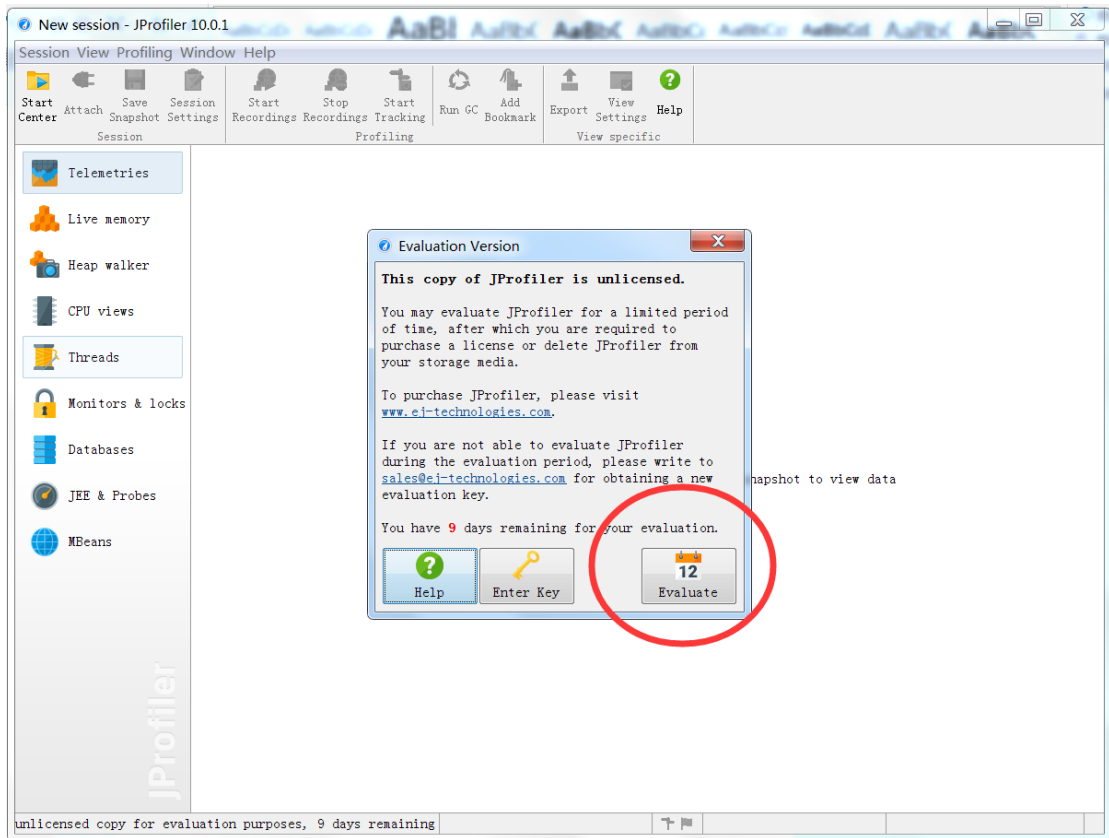


将给出的样例中的 java 文件导入工程（注意给 java 文件添加包名）

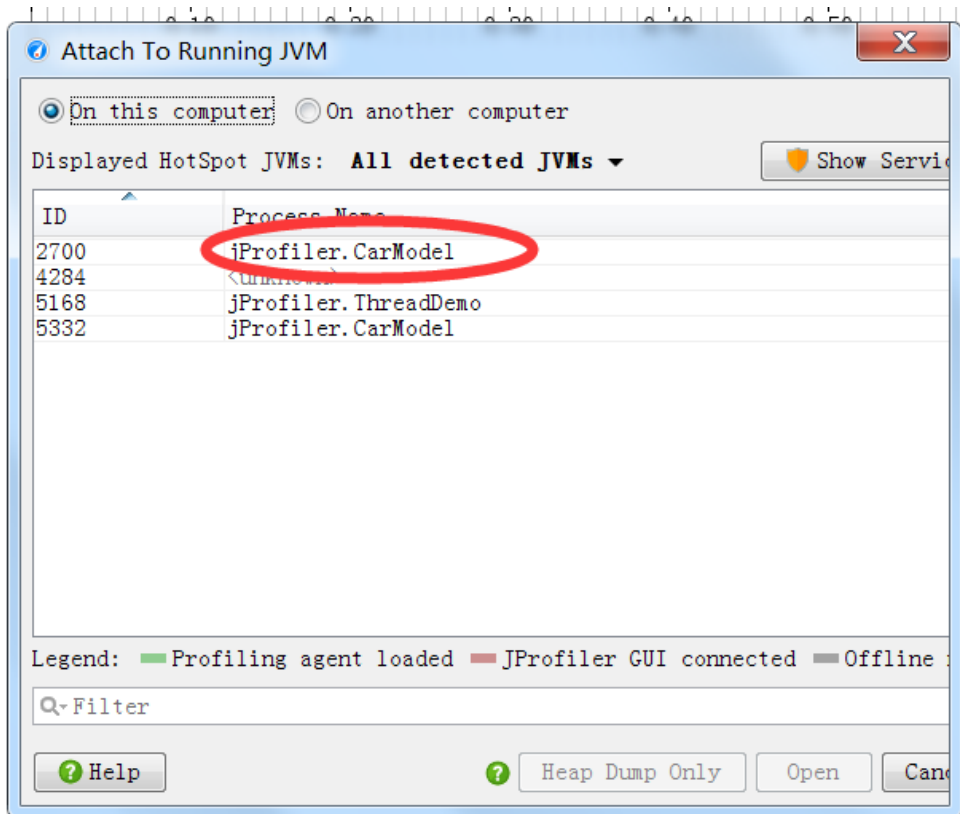
- 运行工程并打开 JProfiler



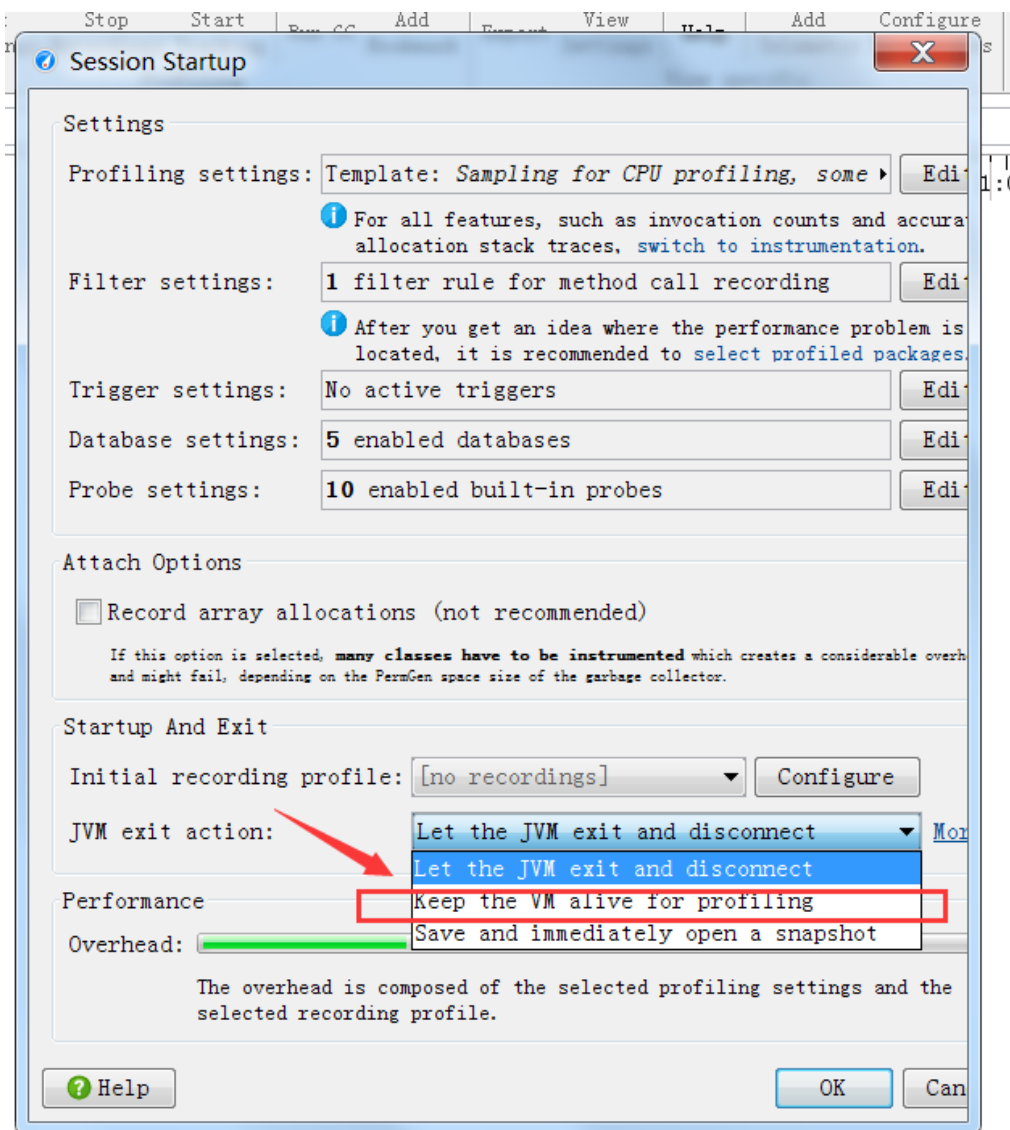
按照以下操作进行



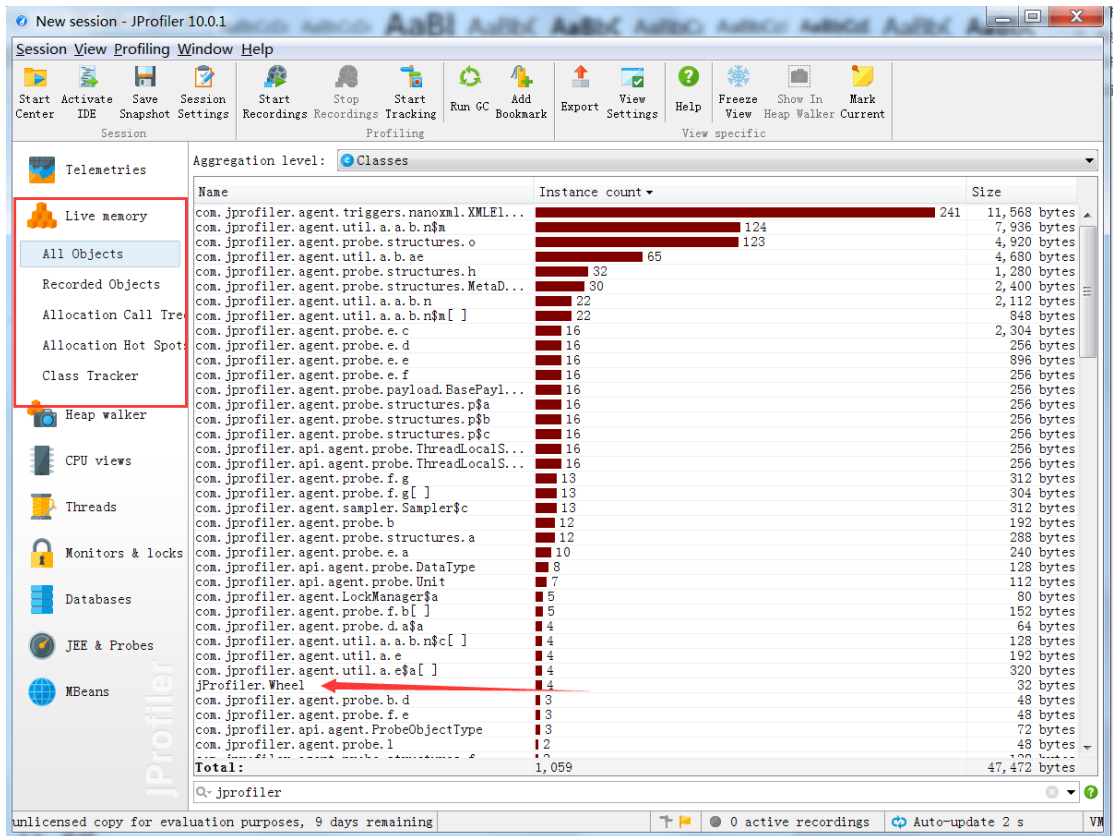
选择当前 jvm 中正在运行的目标工程



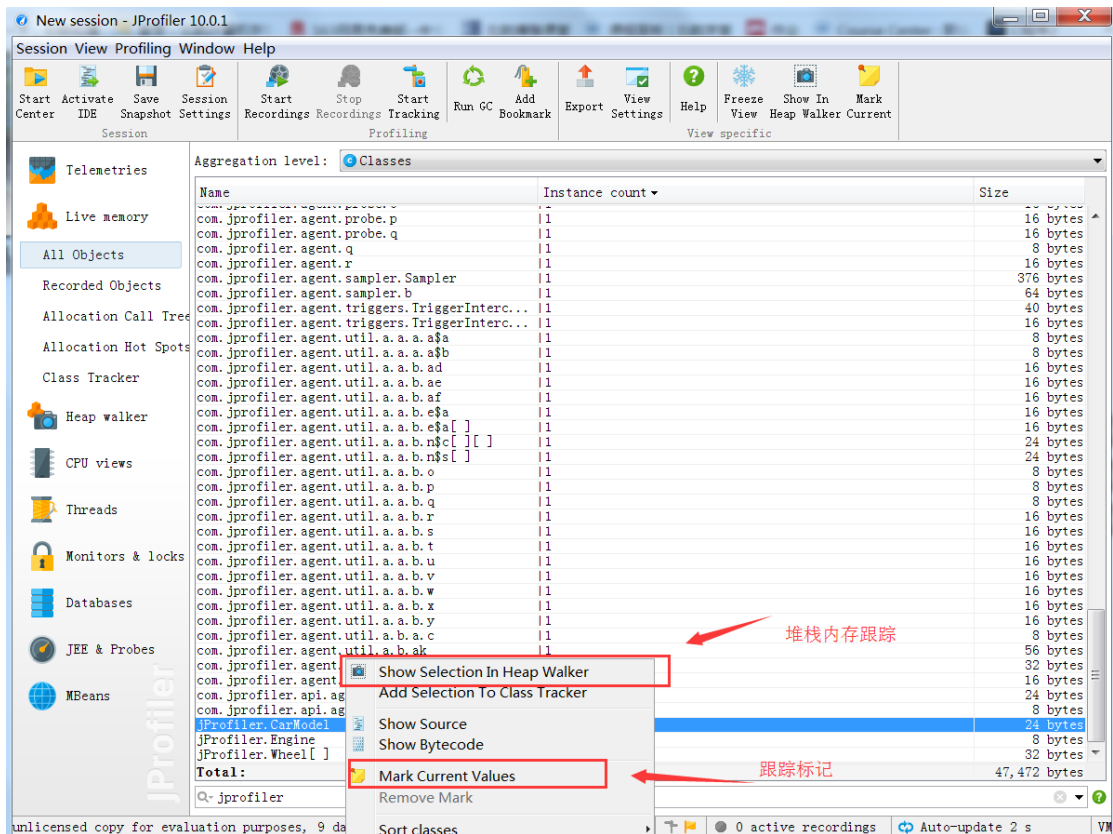
open 后，在 jvm exit action 中



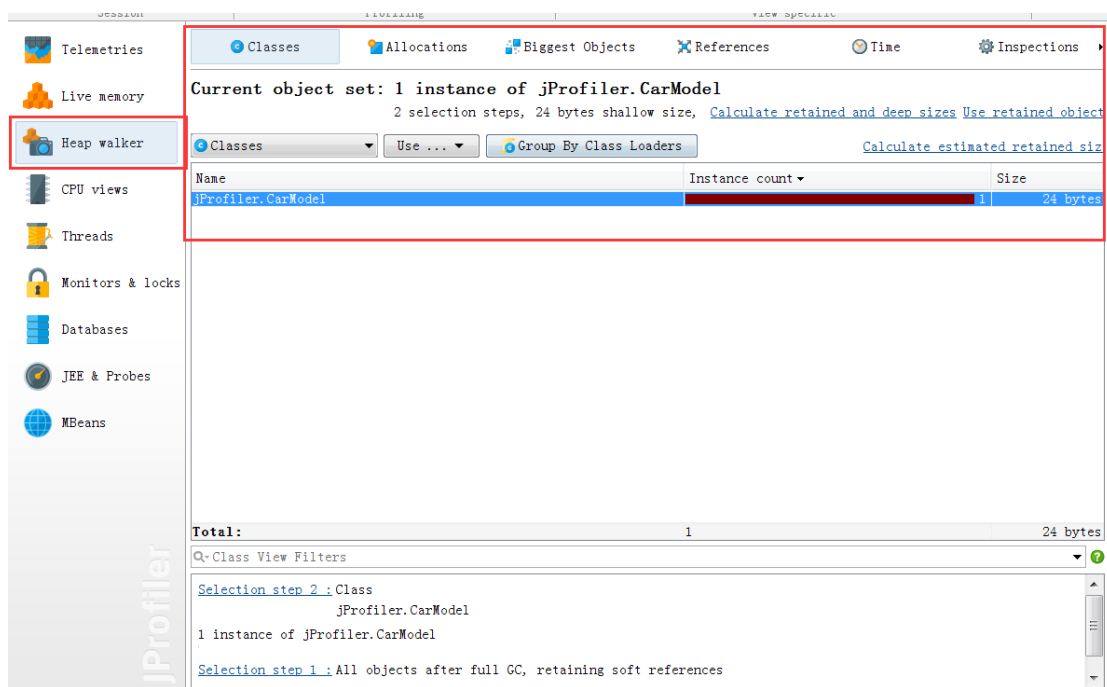
选择 Keep the VM alive for profiling 后，点击 OK，进入 JProfiler 界面如下，
整体概览



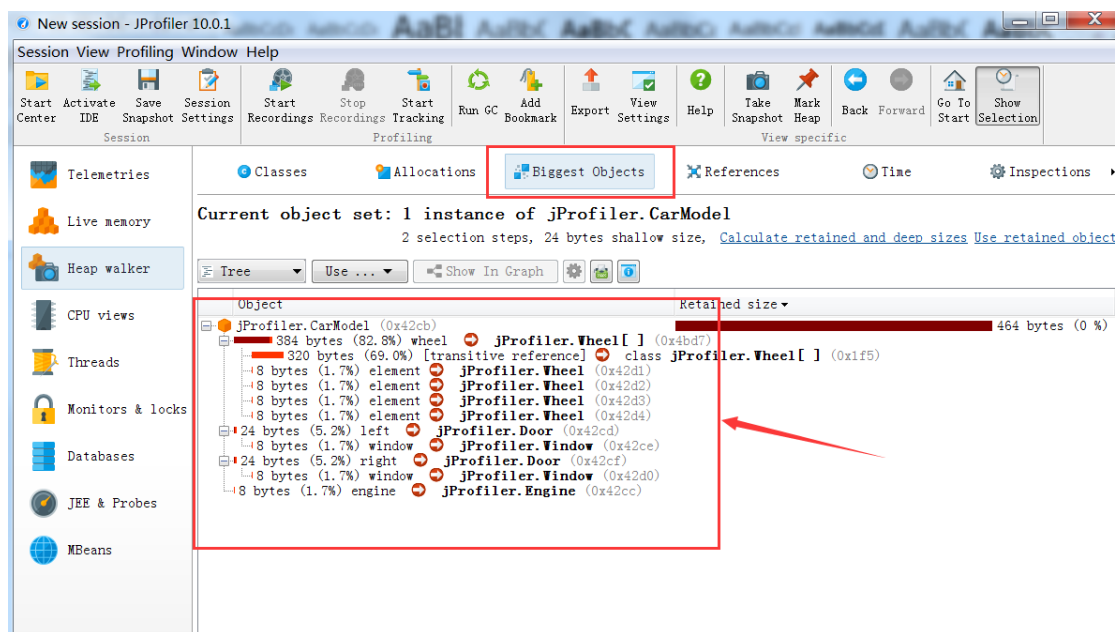
更多当前页面操作



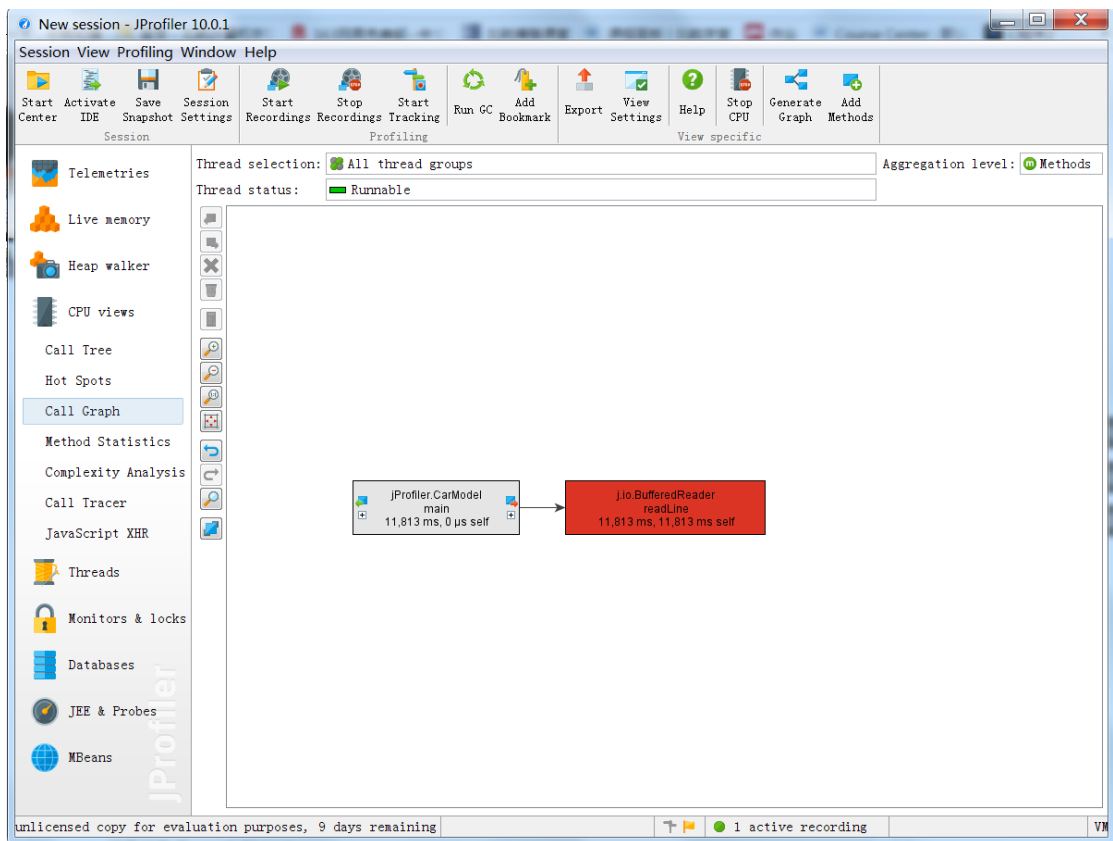
堆中信息跟踪



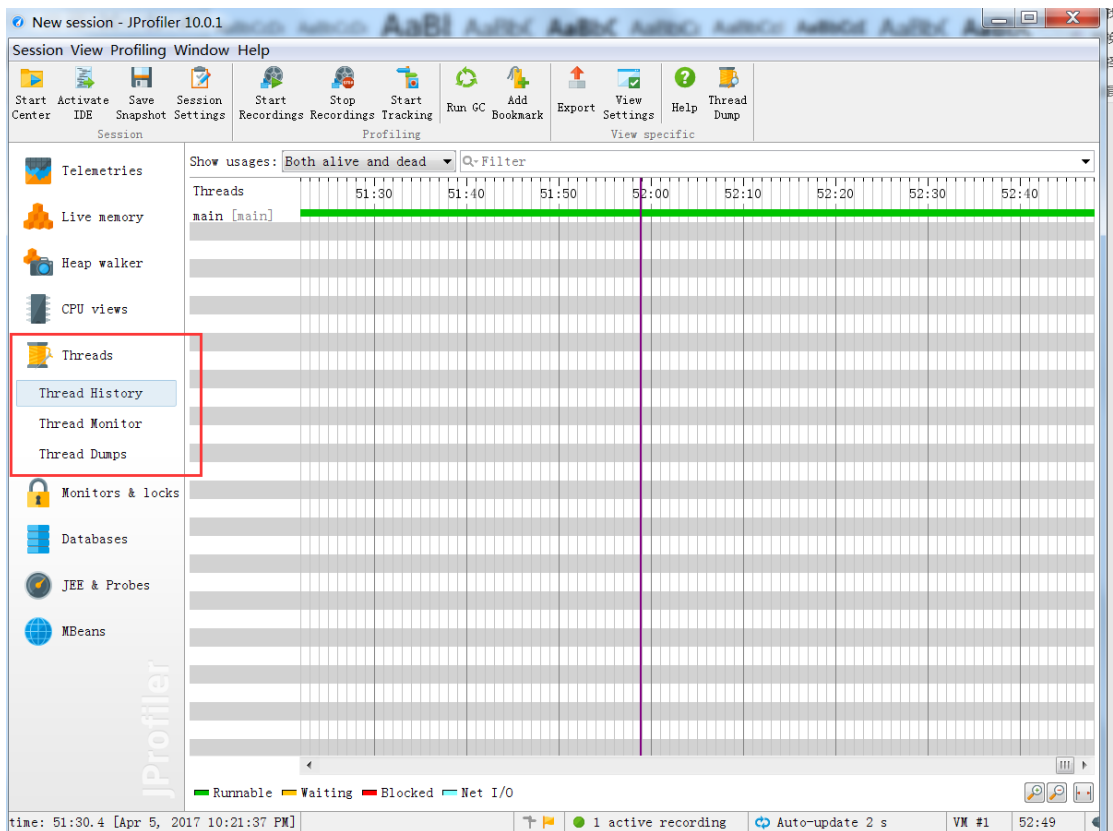
查看类和对象信息



查看调用图



查看线程信息



更多细节教程参考：

<http://blog.csdn.net/chendc201/article/details/22897999>

4. 预习要求

- (1) 在自己的笔记本电脑上安装 JProfiler , 并与 Eclipse 或 IntelliJ IDEA 进行集成 (选择其一即可)
- (2) 运行样例文件 , 查看前文所述的相关信息 , 熟悉操作步骤
- (3) 本次实验课请携带安装并集成好 JProfiler 的笔记本电脑