操作系统作业文档

——1753603 林寅嘉

## 选题：模拟动态分区分配

## 主界面

主界面如下图所示：

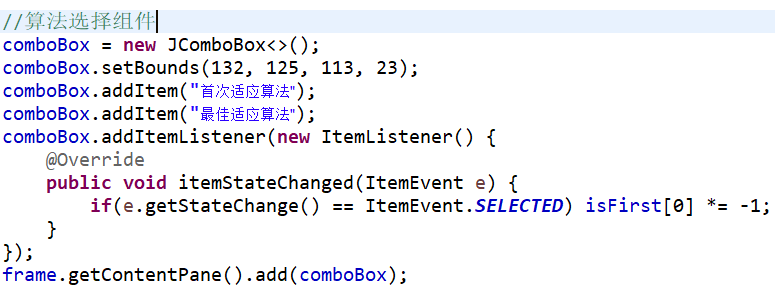
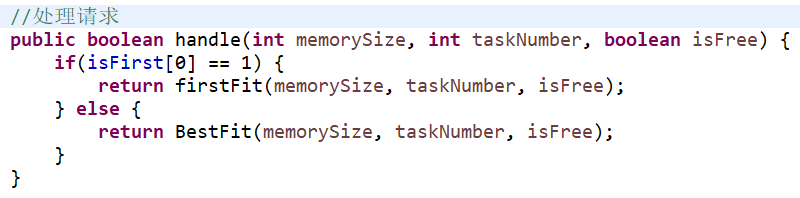


本程序可供用户自行决定所使用的算法，申请（释放）内存，任务的编号，以及申请内存的大小。当然如果用户操作过程有误，可以单击“重置“按钮实现对程序的重置。程序还提供了文本框用于显示内存的申请（释放）过程中的信息，以及一个可视化的长方形用于显示当前内存的使用情况。

## 具体功能的实现

## 算法切换

通过一个int类型的标识符来确认当且用户所选择的算法，初始值为1（代表首次适应算法）。每当用户切换算法，标识符\*-1，以此实现切换算法的功能。详细代码如下：



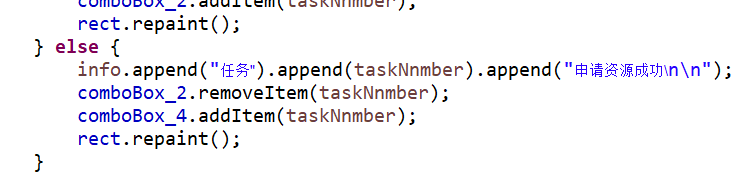
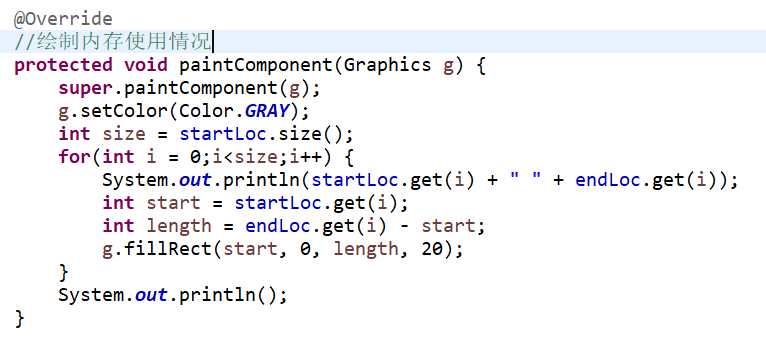
## 申请释放信息的显示

通过文本域来显示信息，每当用户发送一次请求后，都会调用一次Memory.handle()方法，该方法通过返回一个布尔类型的值判断此次请求成功与否，然后将对应的信息显示到文本域中，详细代码如下：



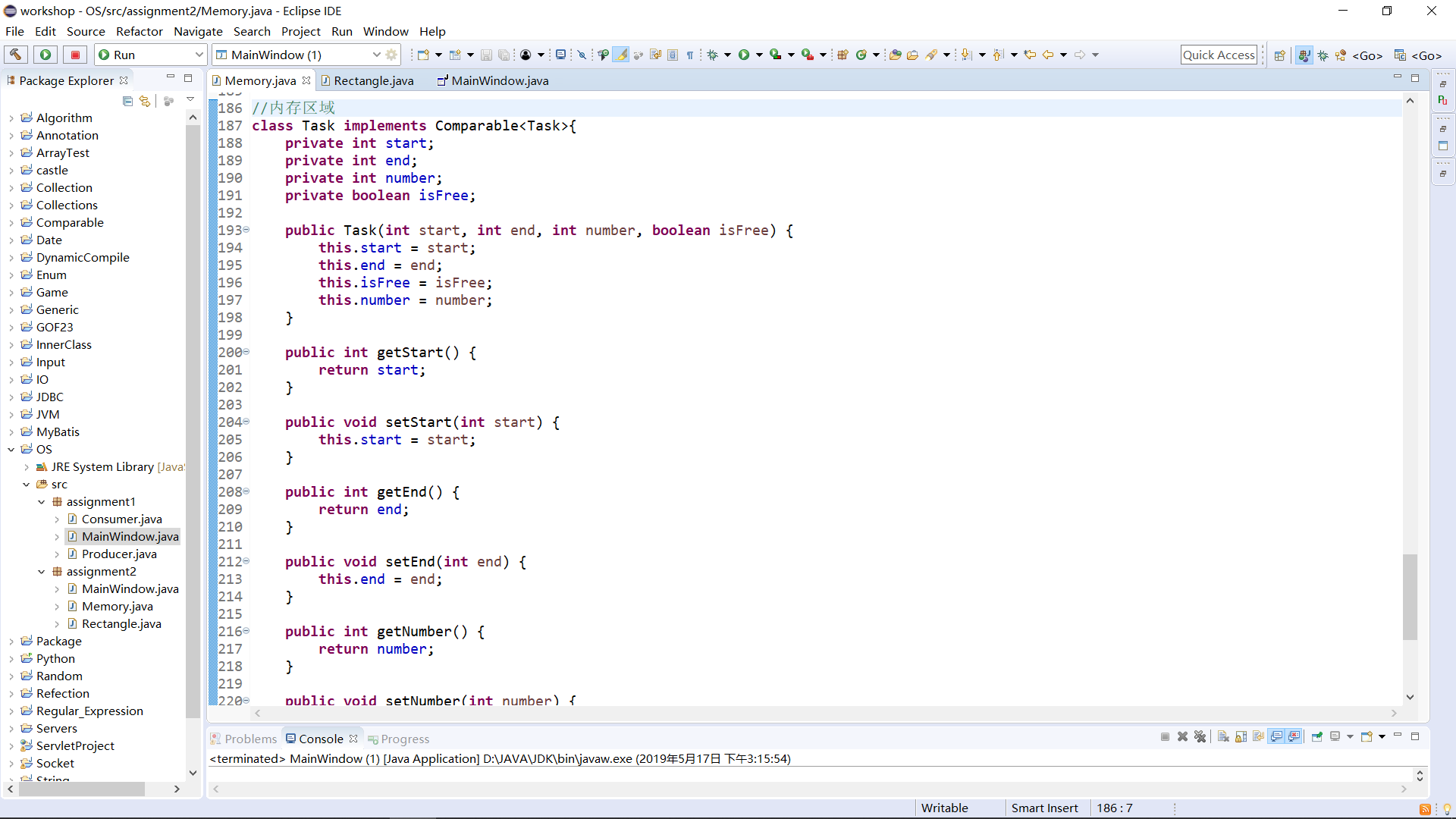
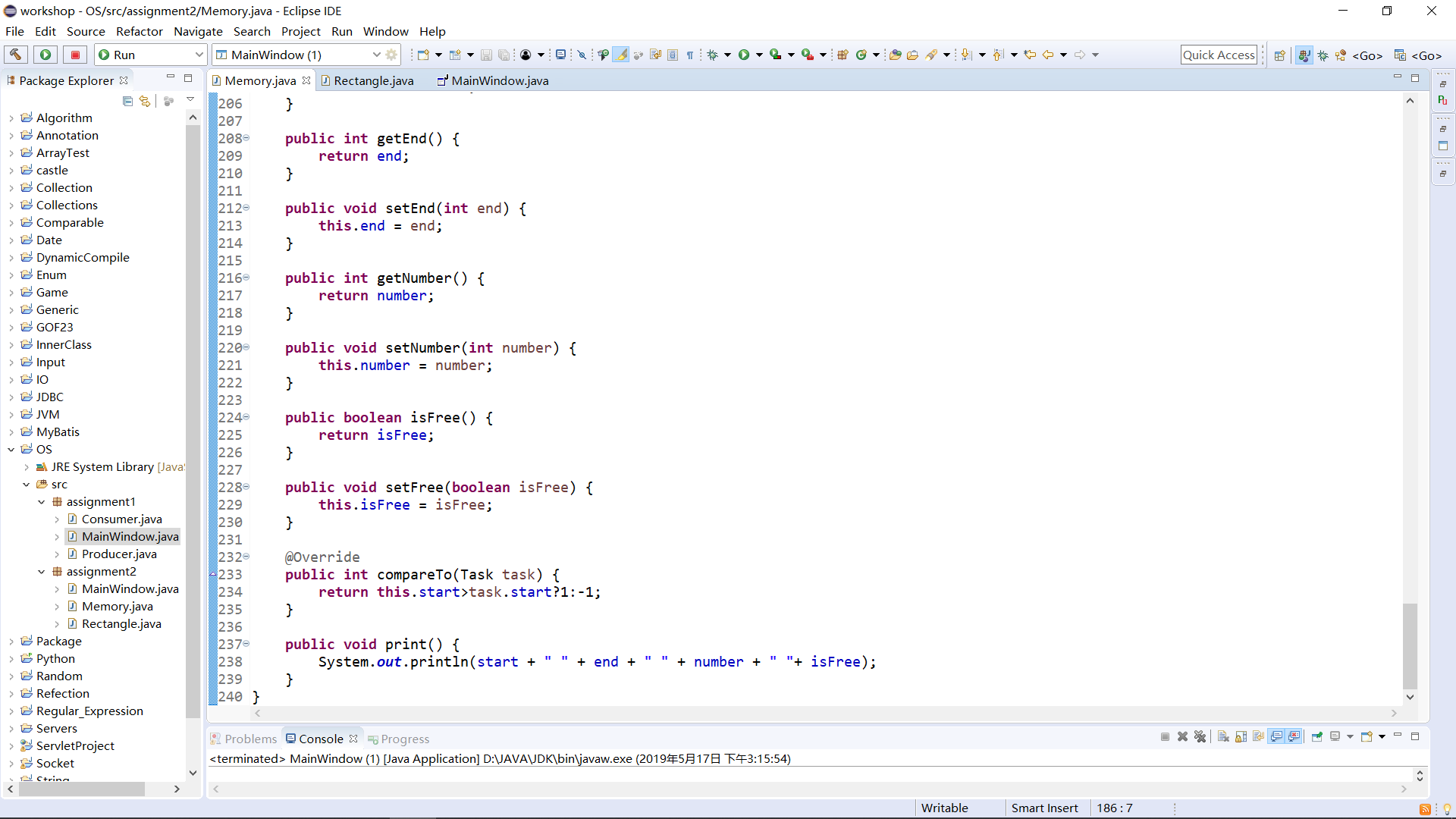
## 3.3可视化内存

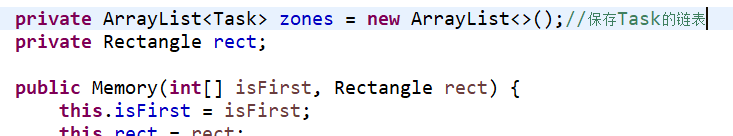
通过绘制矩形框来表示内存的使用情况，白色代表未被使用，灰色代表已被使用。内存的使用信息保存在类型为Task的链表中，将链表中isFree为false的元素的起始位置和终止位置存入startLoc，endLoc两个数组中。每次用户请求成功后，通过遍历该数组来对内存使用情况重新绘制。详细代码如下：



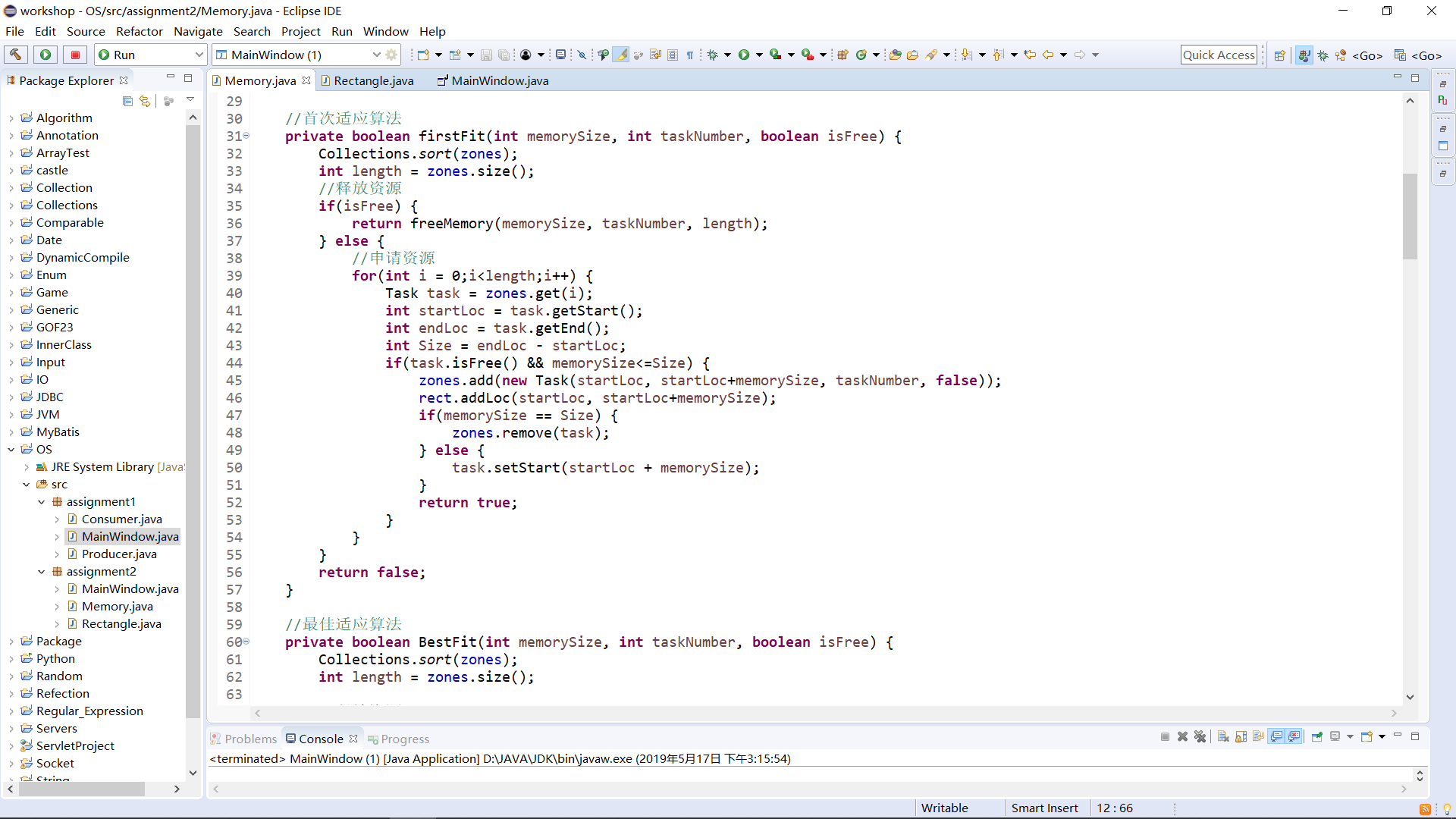
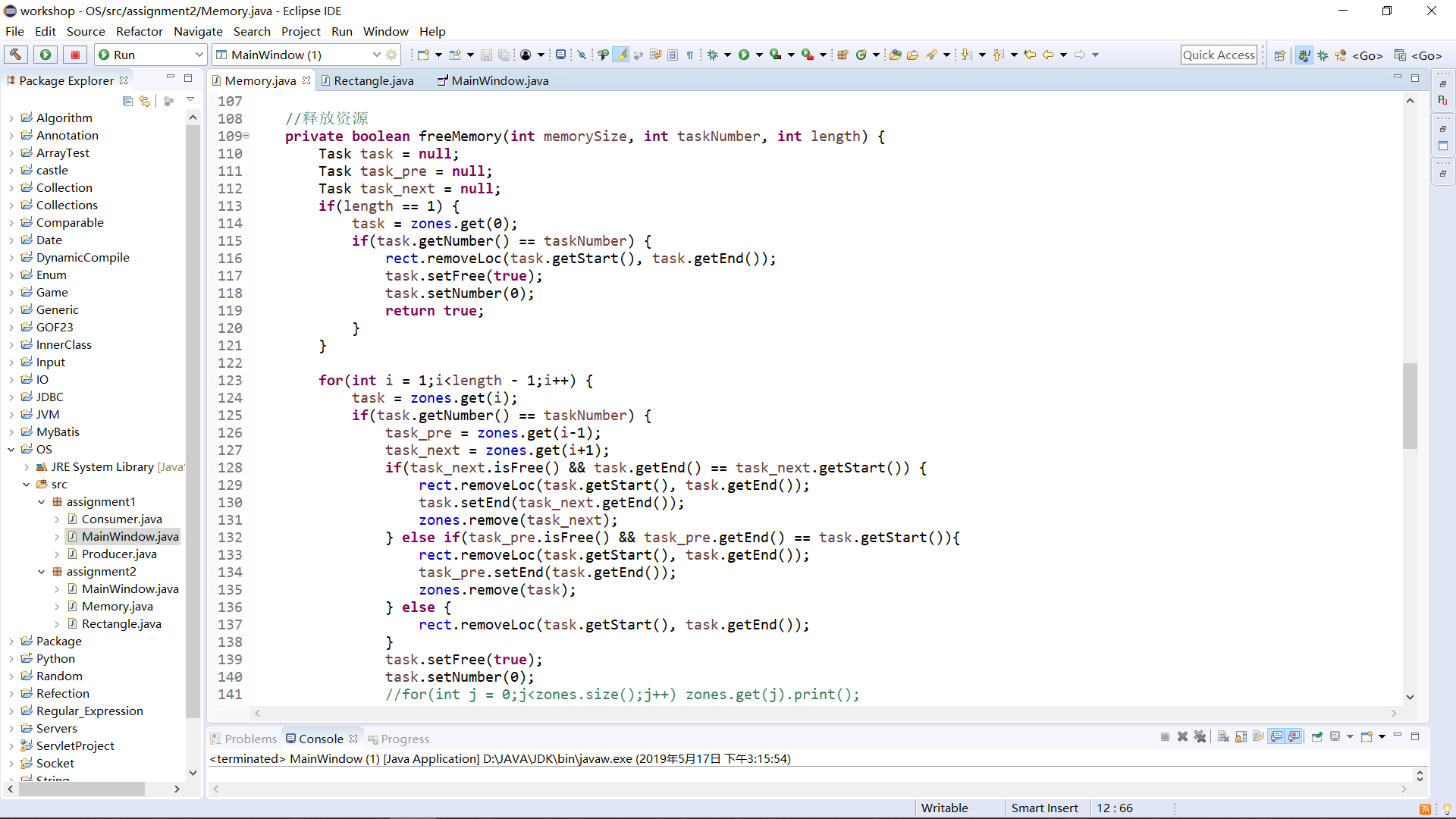
## 首次适应算法的实现

根据算法，第一次找到合适的空闲内存时，就将内存分配给该任务。为简化搜素过程，需要一个Class来保存每段内存的使用情况（包括内存的起止坐标，是否为可用以及占有其的任务编号），将这些Class保存到一个链表中，还需对其根据起始坐标进行升序排列，这样就可以简化搜索过程。Class的代码如下：



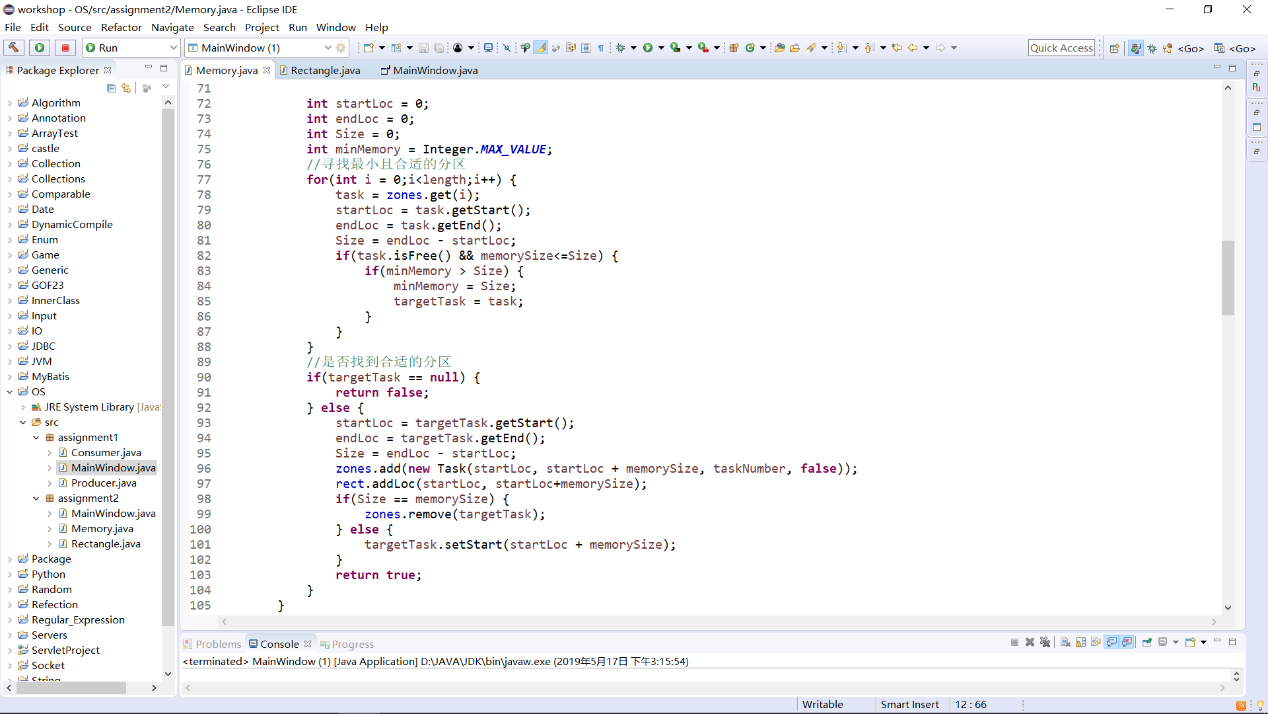


算法的具体实现：遍历zones链表，如果Task的isFree属性为true且其的内存大于等于用户申请的内存，就将其分配给相应的任务，同时将其的isFree属性设置为false，返回true，更新文本域和可视化组件。详细代码如下：



## 最佳适应算法的实现

根据算法，程序需要找到最小且符合条件的内存区域，为此申明一个Task类型的中间变量，每次找到合适的内存区域后，将其的长度与Task的长度进行比较，根据结果对Task进行相应的更新操作，在遍历zones后。Task中所保存的信息就是算法所需的信息。详细代码如下：



## 重置功能的实现

## 4 注意事项

1.请务必保证jre文件夹在可执行文件的同级目录下，否则此程序将会因为找不到Java运行环境而出现闪退的情况。

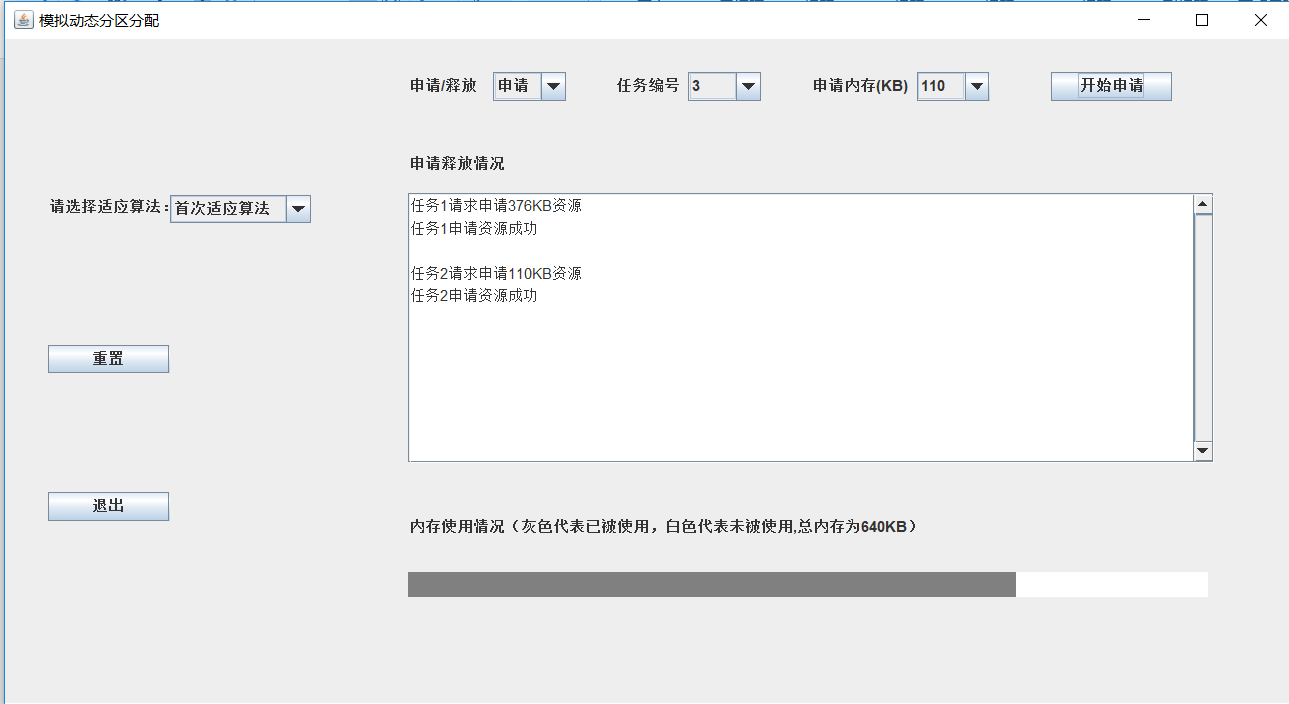
2.此程序默认每个任务在释放其所使用内存不能够再次申请内存，若想继续申请，请先释放此任务所占用的内存。

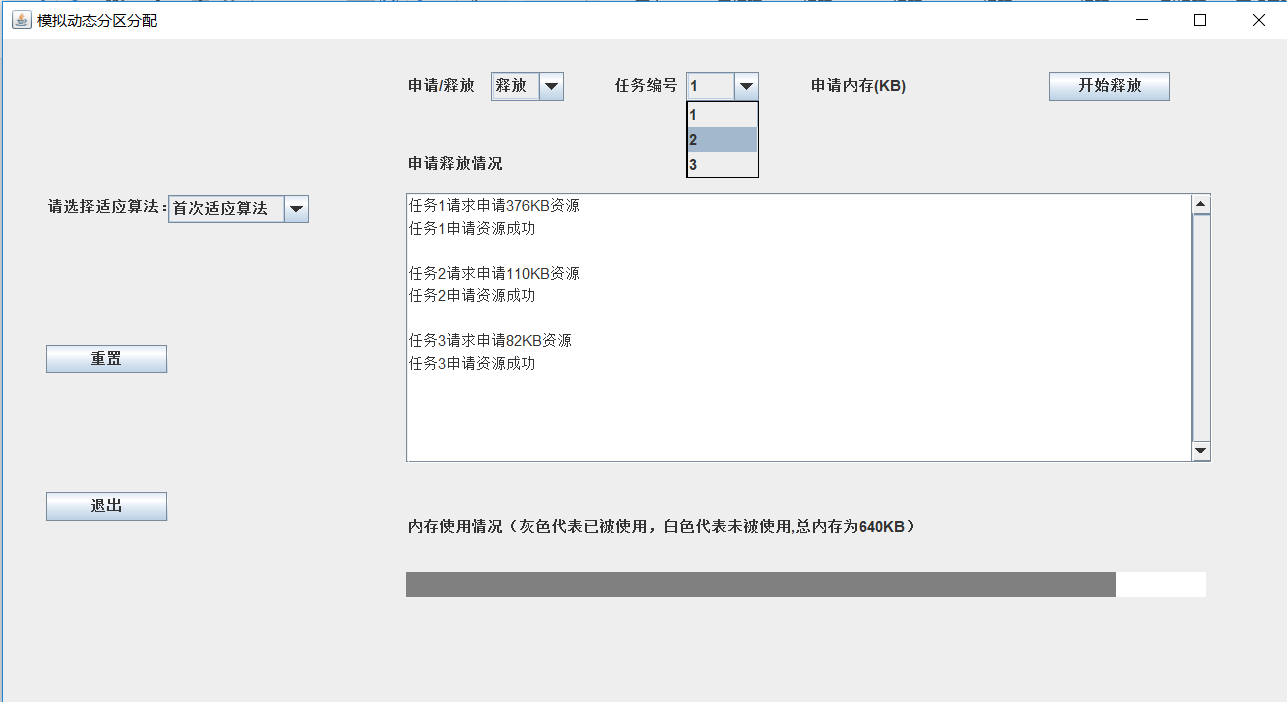
## 5 所使用的语言及工具

语言：Java

工具：eclipse，exe4j

## 6 项目截图演示

1. 申请内存：
2. 
3. 释放内存：



1. 重置：

