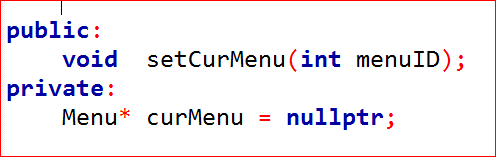
作业03 创建型+结构型+其它(以作业01中src目录中的代码为基础)

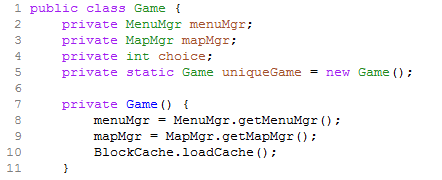
1. 在game中维护着curMenu,见下图:



在运行时，应只有一个curMenu对象。而MenuMgr作为所有menu的管理器，也可以管理curMenu。请将curMenu从Game类，移到MenuMgr中，使得运行结果不变。



Game里：

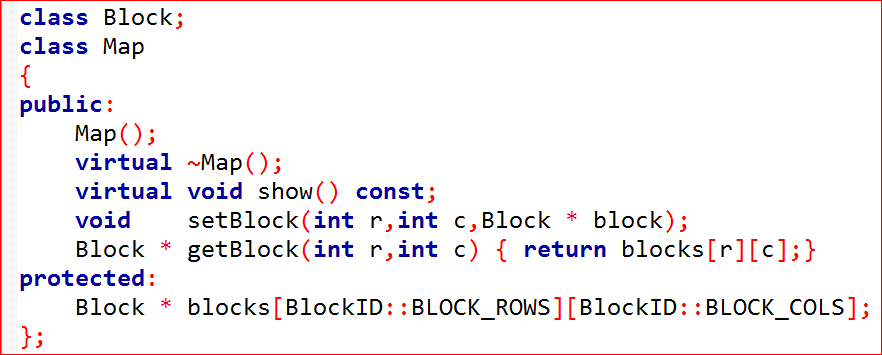




1. 道理同1. 请建立MapMgr类，并将curMap从Game类，移到MapMgr中，使得运行结果不变。



1. 查看Map.h



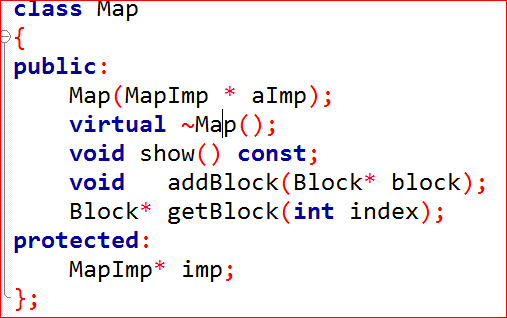
其中各Block以二维数组的形式组织。事实上，各Block的组织形式是多种多样的，如

A A A A A  
 B B  
 C C

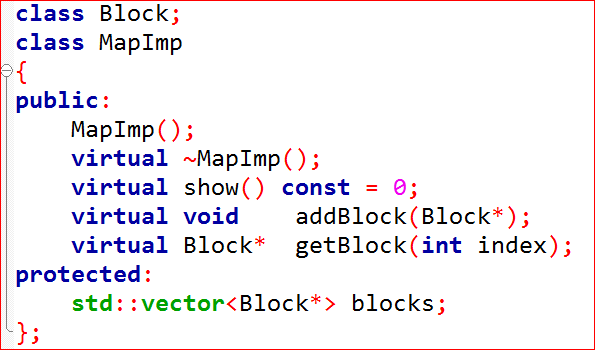
D D

A A A A A

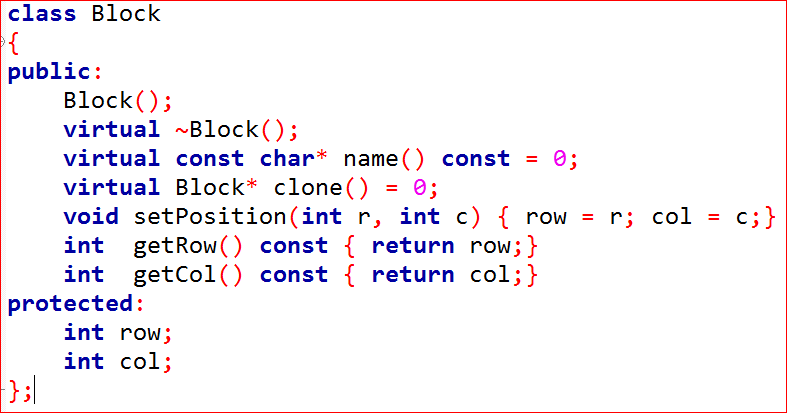
既可以用二维数组的形式表示布局结构，也可以用链表、有向图等表示，甚至各Block是通过立体的关系构成Map。请使用桥接模式分离Map的接口和具体实现。同时,为保持Map接口的一般性，改为通过线性索引的方式读取/添加Block。即Map可如下：



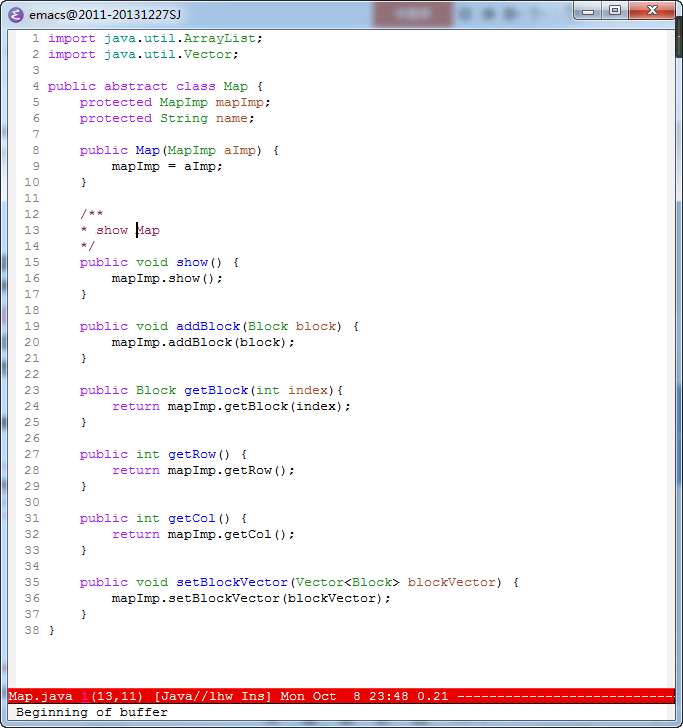
MapImp类可如下:(当需要不同的实现时，从MapImp派生即可)

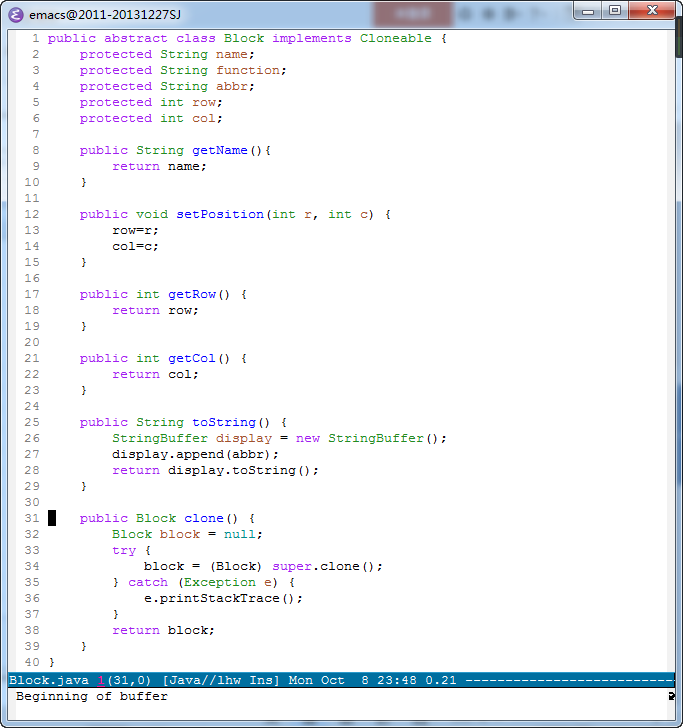


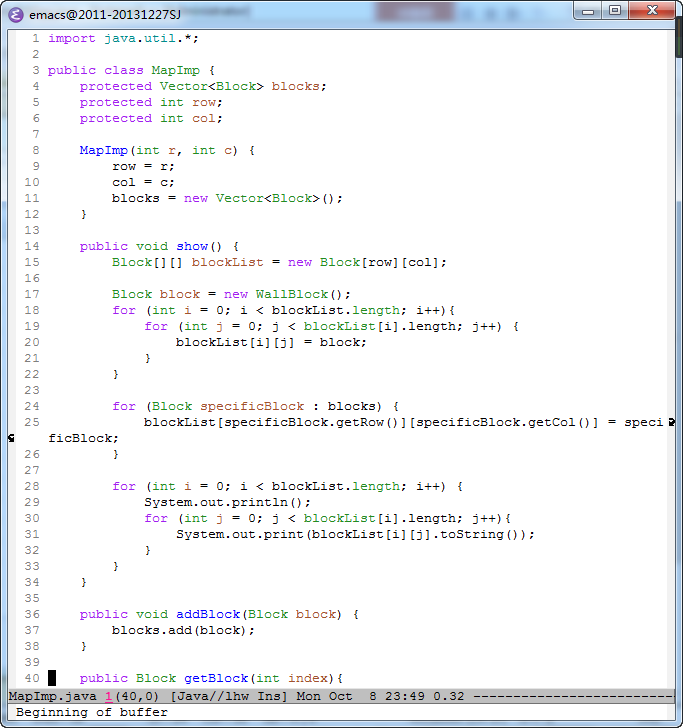
既然Map不再记录Block的二维位置信息，那么可以让Block记录自己的二维位置信息，即Block类可修改为：

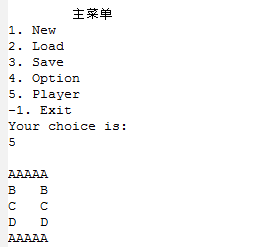


请完成相关代码的修改，使得程序功能不变。









1. 前边的代码中，地图的布局是由二维数组给出的，如：

很明显，其隐含地指出：相邻的两个非空Block是相通的，不相邻两个非空Block是不能直接连通的。但实际上，地图的布局不一定以数组形式给出，可以用其它方式给出，例如下图（地图B）中的布局，可能以下表的形式给出：

A A A A A  
 B B  
 C C

D D

A A A A A

（地图B的布局含义）

A A A A A

B B B

C C C C

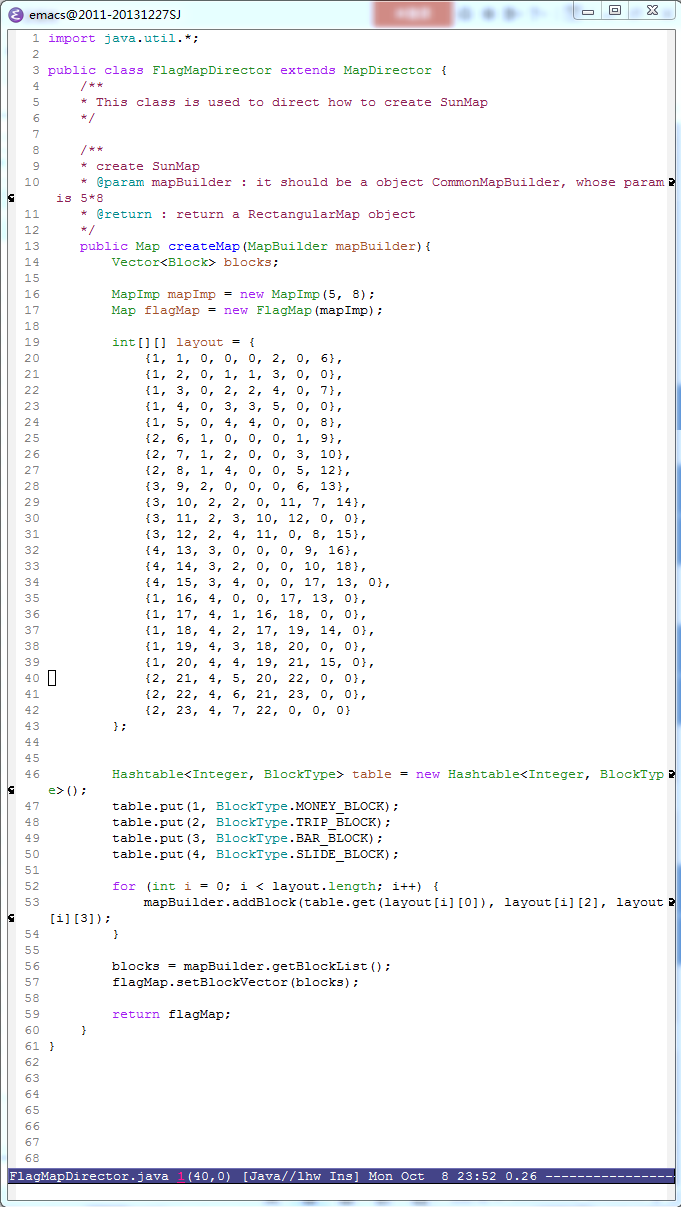
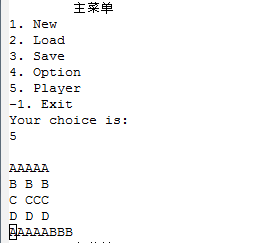
D D D

A A A A A B B B

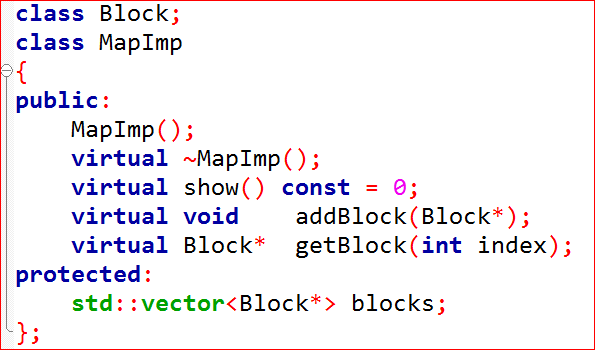
（对应地图B布局实际给出的数据表）

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **类型** | **序号** | **行** | **列** | **LEFT** | **RIGHT** | **UP** | **DOWN** |  |
| **A** | **1** | **0** | **0** | **0** | **2** | **0** | **6** |  |
| **A** | **2** | **0** | **1** | **1** | **3** | **0** | **0** |  |
| **A** | **3** | **0** | **2** | **2** | **4** | **0** | **7** |  |
| **A** | **4** | **0** | **3** | **3** | **5** | **0** | **0** |  |
| **A** | **5** | **0** | **4** | **4** | **0** | **0** | **8** |  |
| **B** | **6** | **1** | **0** | **0** | **0** | **1** | **9** |  |
| **B** | **7** | **1** | **2** | **0** | **0** | **3** | **10** |  |
| **B** | **8** | **1** | **4** | **0** | **0** | **5** | **12** |  |
| **C** | **9** | **2** | **0** | **0** | **0** | **6** | **13** |  |
| **C** | **10** | **2** | **2** | **0** | **11** | **7** | **14** |  |
| **C** | **11** | **2** | **3** | **10** | **12** | **0** | **0** |  |
| **C** | **12** | **2** | **4** | **11** | **0** | **8** | **15** |  |
| **D** | **13** | **3** | **0** | **0** | **0** | **9** | **16** |  |
| **D** | **14** | **3** | **2** | **0** | **0** | **10** | **18** |  |
| **D** | **15** | **3** | **4** | **0** | **0** | **12** | **20** |  |
| **A** | **16** | **4** | **0** | **0** | **17** | **13** | **0** |  |
| **A** | **17** | **4** | **1** | **16** | **18** | **0** | **0** |  |
| **A** | **18** | **4** | **2** | **17** | **19** | **14** | **0** |  |
| **A** | **19** | **4** | **3** | **18** | **20** | **0** | **0** |  |
| **A** | **20** | **4** | **4** | **19** | **21** | **15** | **0** |  |
| **B** | **21** | **4** | **5** | **20** | **22** | **0** | **0** |  |
| **B** | **22** | **4** | **6** | **21** | **23** | **0** | **0** |  |
| **B** | **23** | **4** | **7** | **22** | **0** | **0** | **0** |  |
|  |  |  |  |  |  |  |  |  |

若希望每个Block“知道”其相邻的各Block，请扩展3中MapDirector、MapBuilder、Block类，同时改写MapMgr中的createMap函数，使得新的程序能够根据给定数据表正确创建并显示地图（可在global.h中添加常量）。

1. 回看3中的MapImp类：



请问“MapImp类的设计中，使用了适配器模式”，这句话对吗？如果正确，回答适配了什么？采用类适配器模式可以吗？如果不正确，回答为什么？

Answer：

是对的，适配了Map类，将Map的显示等方法交给MapImp。但是个人认为由于Map的方法缺少变化，这里的适配有点名不符实。

我写的java，没办法使用多重继承，也就没办法用类适配器。