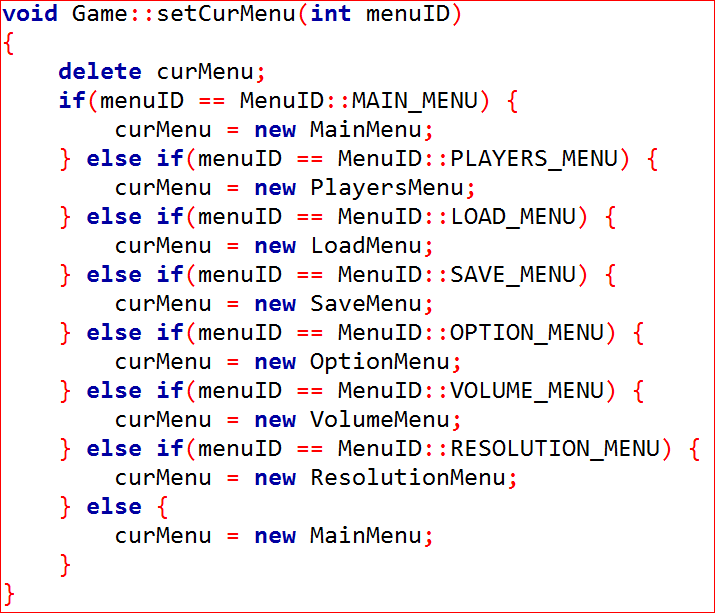
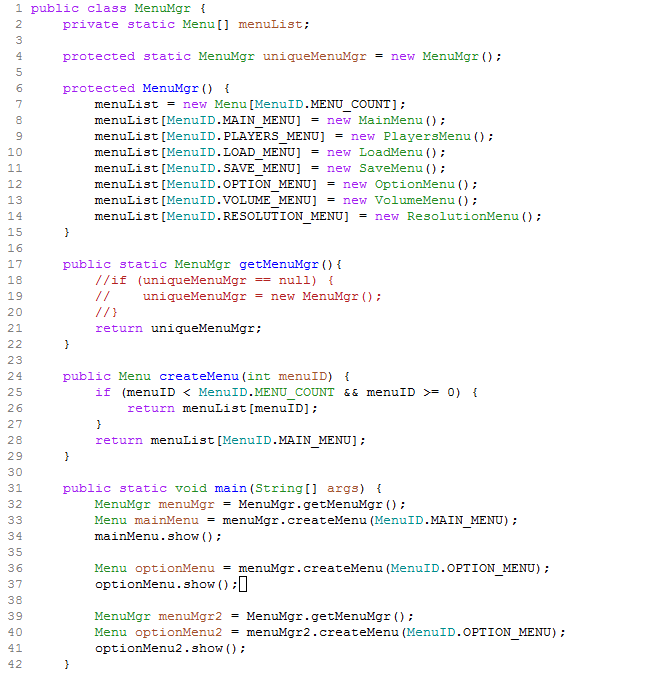
作业01 创建型(以准备完成中的代码为基础)

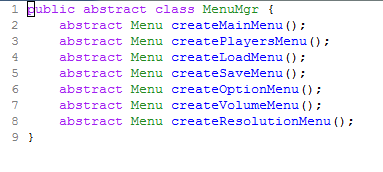
1. 在game.cpp中，存在代码如图  
   

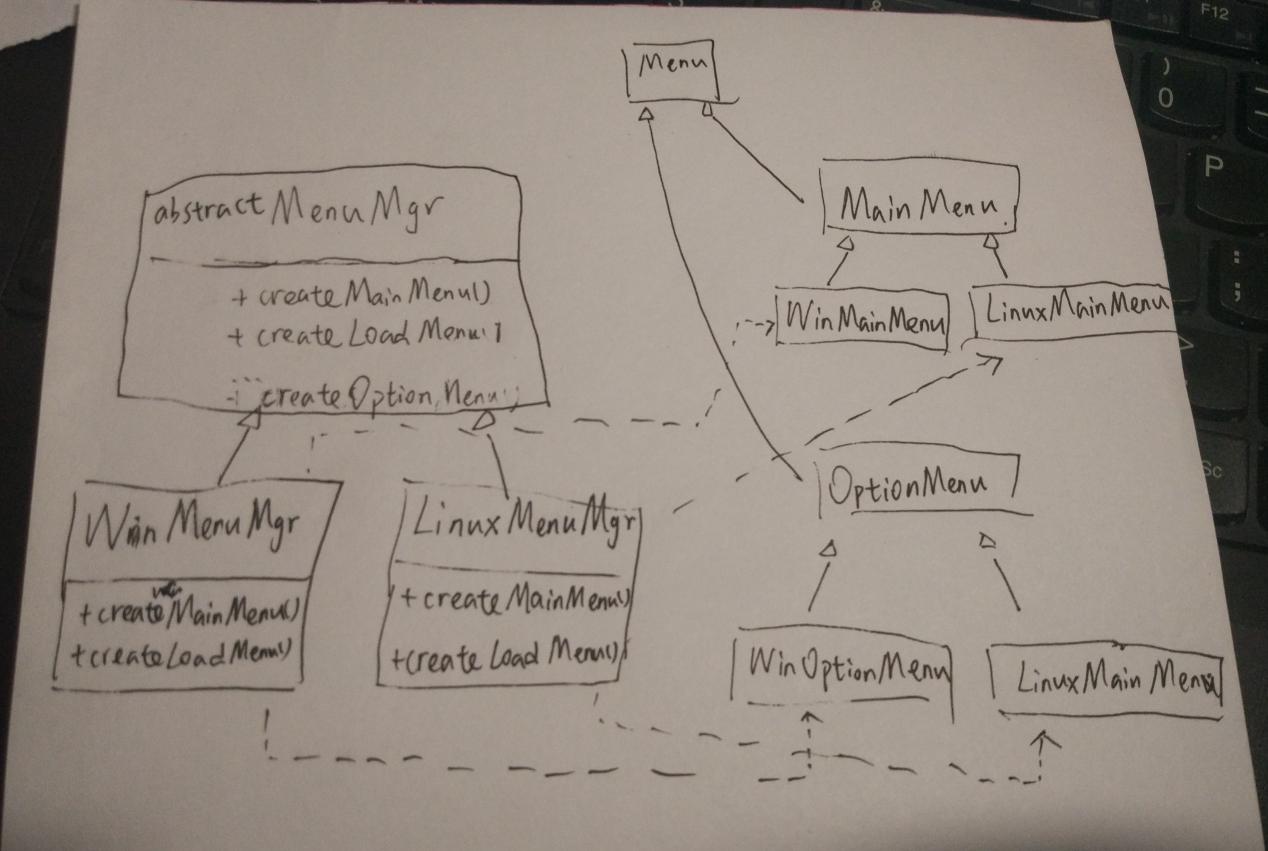
setCurMenu中需要频繁创建销毁菜单，显然是没有必要的。请使用多例模式设计并实现MenuMgr类，使得各菜单至多只需要创建一次和销毁一次，并保持原程序功能不变。

1. 对于1中的MenuMgr类，没有必要实例化多个对象，请在1的基础上对MenuMgr本身继续应用单例模式。
2. 若已知某些菜单可能会变化，例如SaveMenu可能会有其它样式或内容。请对2中的MenuMgr继续应用工厂模式，使得可以方便地更换成新的SaveMenu菜单。

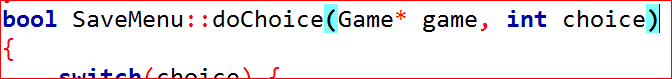


1. 若已知未来会因为某些原因（如平台移植、版本升级的），全部菜单可能会整体替换掉，请对MenuMgr应用抽象工厂模式，使得可以方便地用新菜单组替换原有菜单组。





1. 在各Menu中，doChoice函数，都需要参数game。



从代码语义可以看出，虽然需要频繁传递game参数，但这个game都是同一个对象，而且整个程序中，有且只应有一个game。请用单例模式重新设计Game类，使得doChoice函数不再需要显式传递game参数。



1. 若Map由多种Block组成，各种Block有不同的游戏效果，如MoneyBlock带有一些钱，玩家经过时，获得这些钱；对于TripBlock，玩家正好走到这种Block时，金钱和生命都受到损失；BarBlock，则当玩家经过时，会被阻挡而本轮不能继续前进；前进到SlideBlock时，会继续前进2格等。以上四种Block分别表示为ABCD。
   1. 当前代码，在用户选择Play菜单项时，只是显示字符串Playing，而现需要显示由各Block组成的Map。若Map为

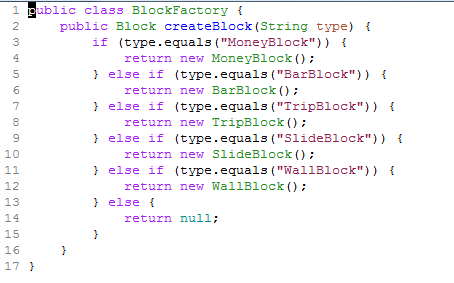
A A A A A  
 B B  
 C C

D D

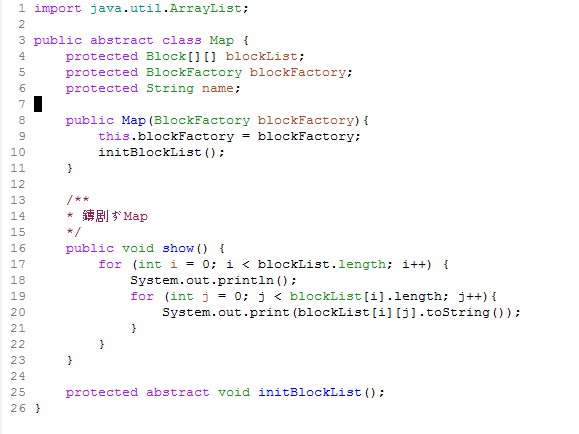
A A A A A

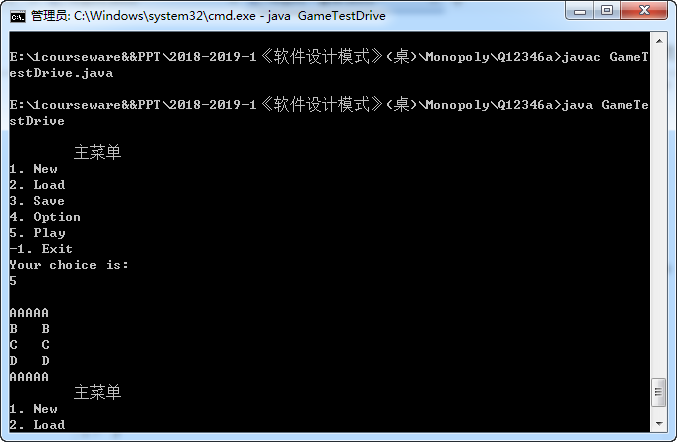
使用工厂方法模式将Block的创建独立出来并创建定义并实现Map及相关类。

Block工厂

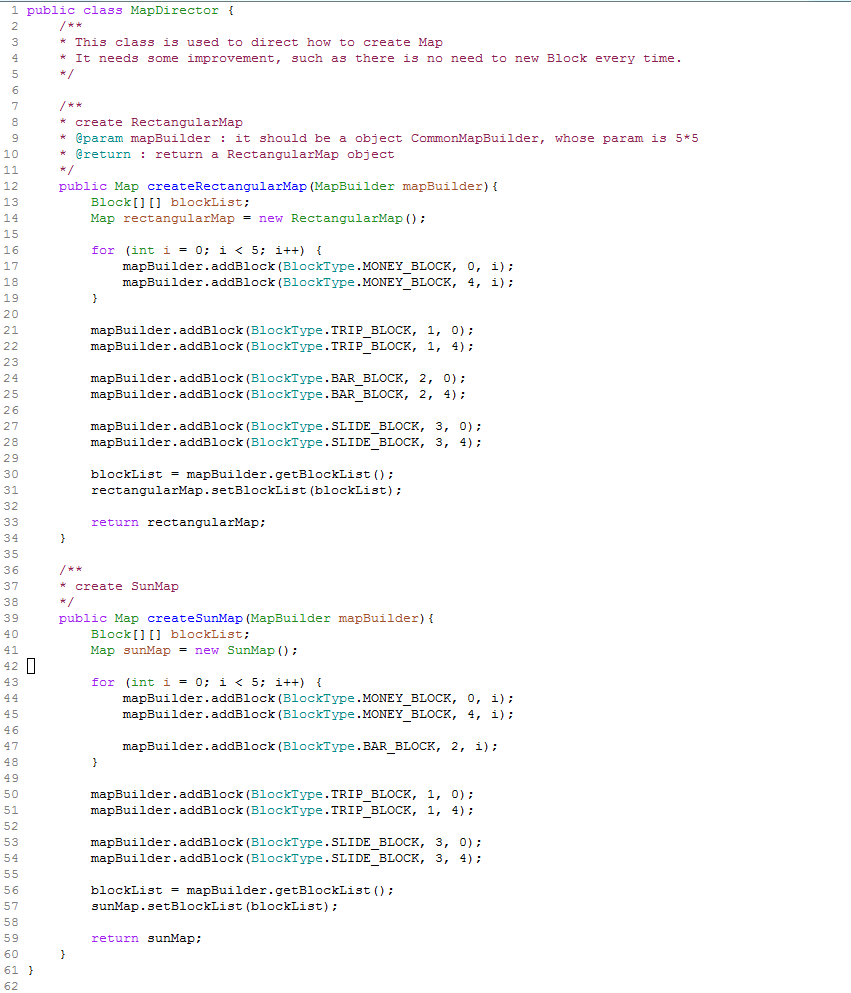


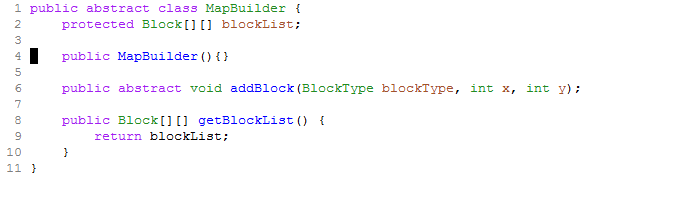
Map类

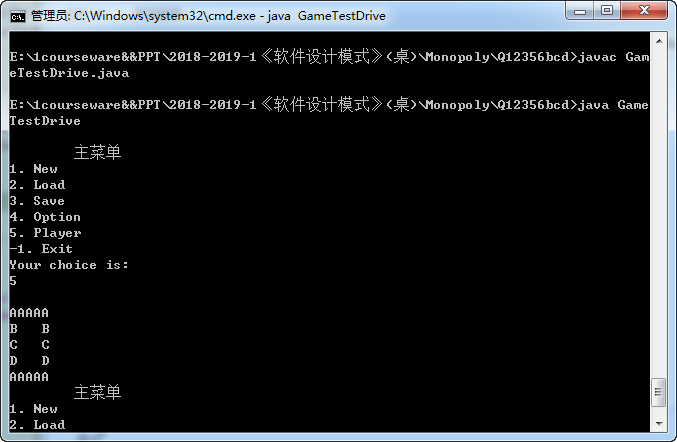




* 1. 请使用生成器模式重新创建定义并实现Map及相关类。







* 1. 本例中，director和builder哪个部分可以退化？

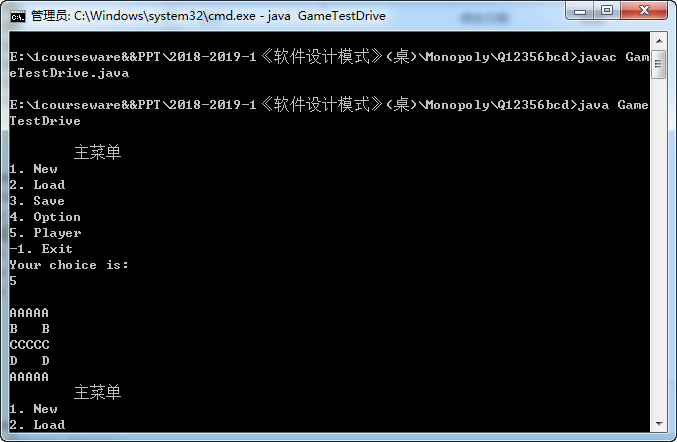
builder可以退化

* 1. 扩展MapDirector的类，使得可以方便地变更Map为：

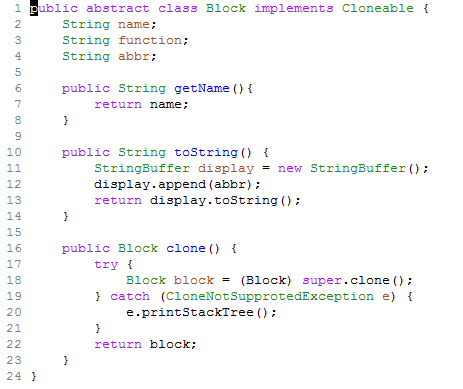
A A A A A  
 B B  
 C C C C C

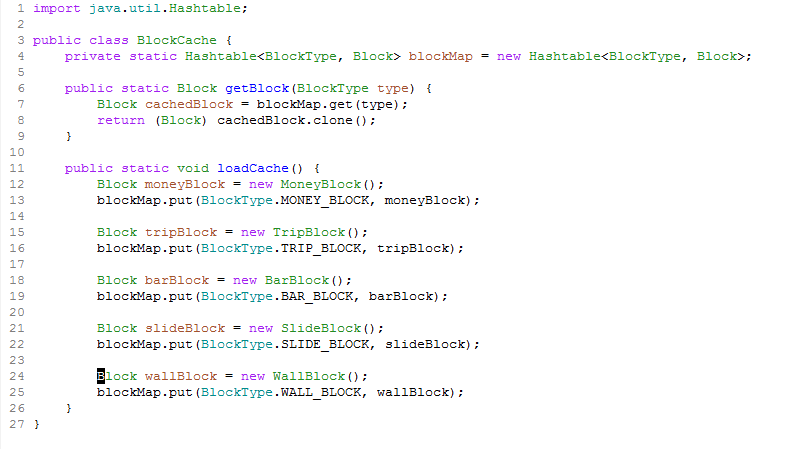
D D

A A A A A



* 1. 改用原型方法设计各Block类；





* 1. 使用map的生成器和Block的原型管理器，实现**动态改变**”零件”,但使用相同”布局”,构建新Map(如)。

E E E E E  
 B B  
 F F F F F

D D

E E E E E

