

Java mini cad报告：

设计理念

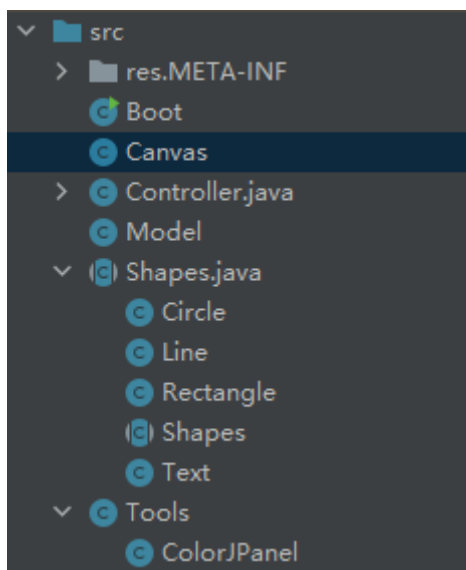
- 本程序由java实现，使用了swing等库
- 采用MVC模型，程序内部分为几个大类
- 程序严格执行MVC模型和面向对象的程序设计思路，各个类之间的通信全部靠调用方法实现，相互之间耦合度极低。

功能描述：

- 支持直线绘制，矩形，圆形绘制，字符串绘制
- 支持鼠标拖拽移动图形位置
- 支持鼠标单击图形选中，选中后图形边线会变粗或者字符串变为斜体加以提示：
- 按下+可以让图形边线变粗/字符串字体变粗
- 按下-可以让边线变细/字符串字体变细
- 按下del或者d可以删除此图形

实现描述：

程序内部分为几个大类：



Shapes：所有形状类的父类

- 抽象类
 - 实现了Serializable接口
 - 属性：
 - Font——边线大小
 - color——图形颜色
 - Circle、Rectangle、Line、Text：继承了Shape类，并且还保存了各自的形状参数各自封装了自己的API

Canvas：实现程序左侧的画布

- 继承自JPanel
- 注册了画布上的各种事件响应对象，比如mouseClick、mouseDrag、KeyPressed.....

Tools：实现程序右侧的工具栏

- 继承自JPanel
- 注册了工具栏的各种事件响应对象

Controller：实现控制

- 属性：
 - 当前选中的颜色
 - 当前选中的形状对象（画布上的Shapes对象
 - 当前正在绘制的形状对象（和上一个属性不完全一致
 - 当前是否正在绘制形状
 - 当前选中的形状类型（下笔会画出这种形状
- 封装了一系列API，当工具栏上有事件发生时，响应函数会调用此类的API修改控制信息的状态。

Model：存储画在画布上的各种形状对象

- 属性：
 - ArrayList<Shapes> list
- 封装了add, del, getList的方法。

使用描述：

- 进入界面后，选择右侧工具栏中的某一种图形类型，**设定下次绘画的图形**
- 设定下次绘画的图形后，在画布上点击拖拽可以画出圆形、直线、矩形/点击画布上某一位置在此处放入一个字符串。
- 再次选中同一个图形(点击右侧相同的button)可以取消绘图状态，此时鼠标在画布上点击，拖拽将不会绘画，
- 在未选中下次绘画的图形时，可以点击选中已经画上的图形元素。也可以选择其它的形状（点击右侧其它的形状button）改变下次绘画的形状。
- 绘画之前点击色盘上的颜色，可以改变笔的颜色。
- 在没有选中绘画图形的状态下选择画布上的某一个元素可以将它置为选中状态：
 - 此时按下键盘上的 '+'、'-'、'D'（按不按shift无所谓）或者DEL键可以进行相应的操作
 - '+'、'-'会改变图形边线粗细/string的字体大小
 - D/DEL键可以删去这个元素
 - 此时也可以点击右侧的色盘改变选中图形/字符串的颜色，但不会改变笔的颜色。
 - 选中图形/字符串之后才可以进行鼠标拖拽改变它的位置
- 点击save可以将画布上的一切保存到文件中。保存不影响当前画布的状态。
- 点击load可以选择某个保存的画布并且恢复画布和保存时刻的控制状态——包括画布上的所有图形及其颜色、位置等，还有保存时刻笔的颜色，保存时刻选中的形状等等。
 - load操作成功后会覆盖之前的画布，请注意保存。
 - 附带文件cad为一个已经保存过的画布，可以直接load。

