

字母画板项目

项目背景

同学们即将完成基础的程序设计课程学习，这意味着大家要告别黑框命令行，迎接各种新事物。它可以是Web应用开发，也可以是图形界面程序设计，或者是硬件电路设计。其中，图形界面程序设计（GUI, Graphical User Interface）是一门十分有趣的技术，它能为用户提供更友好易懂的界面。按钮，文本框，图标，滑条等等的基础部件组成了整个窗口界面。而这些基础部件，实际上是通过绘制各种线条来实现的。

在本项目中，我们尝试在命令行中通过绘制线条，来画出不同的图形。为了简单起见，我们以画字母作为我们的目标。

项目基本要求

请你设计一个程序，该程序至少包括以下功能：

1. 使用**继承和多态**去表示和处理四种线条（横线—，竖线|，对角线\，反对角线/）以及各种字母，字母中包含多条线段（例如，字母A包含一条反对角线/，一条对角线\，一条横线—）
2. 线条和字母会储存多个位置信息，以便画图时使用（例如，每种线条都需要储存起点和终点的坐标）
3. 主程序需要处理三种输入
 - a. 添加字母
 - b. 打印字母的位置信息（参考图1，图1仅供参考，实际展现形式不一定需要和图中一致）
 - c. 画出所有字母（参考图2，图2仅供参考，实际展现形式不一定需要和图中一致）

程序需要有形状基类，继承形状基类的4个线条类和至少4个的字母类（A，S，Y，U）

```

stat参考样例

Alpha Z(leftup:[5,5])
    Horizon Line(begX:5, endX:10, y:5)
    Anti Diagonal Line(leftdown:[5,10], rightup:[10,5])
    Horizon Line(begX:5, endX:10, y:10)
Alpha Y(leftup:[12,5])
    Diagonal Line(leftup:[12,5], rightdown:[15,8])
    Anti Diagonal Line(leftdown:[15,8], rightup:[18,5])
    Vertical Line(x:15, begY:8, endY:10)
Alpha H(leftup:[20,5])
    Vertical Line(x:20, begY:5, endY:10)
    Horizon Line(begX:20, endX:25, y:22)
    Vertical Line(x:25, begY:5, endY:10)

```

图1

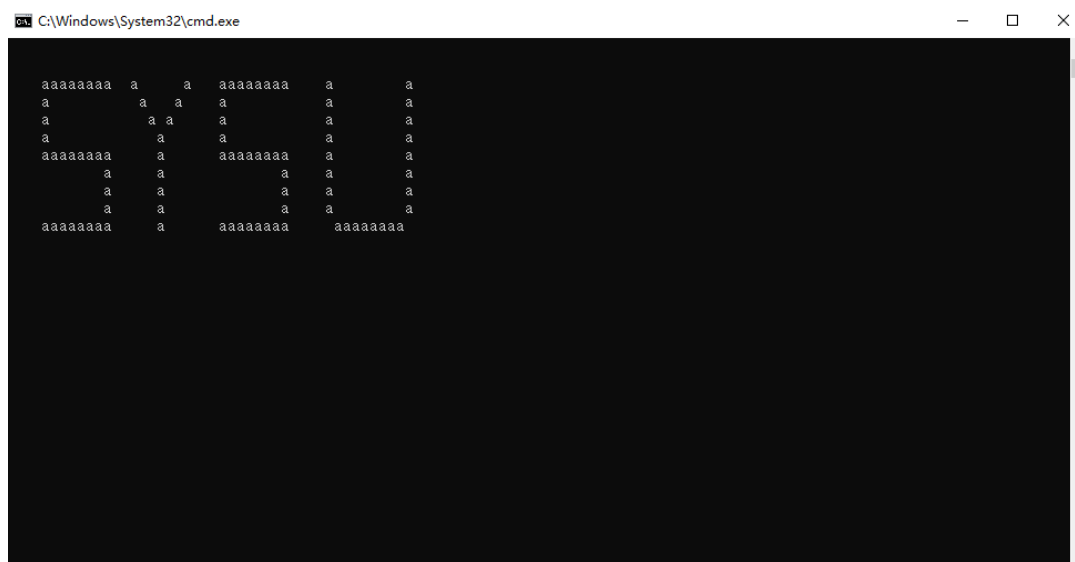


图2

项目加分项

对于项目的创新设计，我们会给予相应的加分，包括但不限于以下：

1. 设计更多的字母类；
2. 设计其他更复杂的形状（如矩形，三角形，圆形）；
3. 实现字母大小的控制。

提交内容

1. 实验代码: 包括所有源代码和可执行文件；
2. 实验报告: 请使用参考实验报告模板，命名格式为”学号1_姓名1_学号2_姓名2.pdf”；

3. README.txt: 包含小组成员的姓名和学号，一段简要的程序说明，文件夹中各个文件的简单描述以及用户友好的使用简介；
4. Alpha.exe: 可直接运行的程序；
5. 以上文件请按照如下文件结构打包成zip格式提交，命名格式为”project2.zip”。

请按以下文件结构提交：

project2.zip

- └ 学号1_姓名1_学号2_姓名2.pdf
- └ README.txt
- └ Alpha.exe
- └ Source
 - └ 源代码文件1
 - └ 源代码文件2
 - └ ...
 - └ 源代码文件n

注意事项

1. 本实验是小组实验，2人一组；
2. 请严格按照给出的格式命名所有文件，请严格按照上述文件结构打包压缩包，否则可能扣分；
3. 关于实验题目有任何问题欢迎通过邮件联系TA，邮箱：1412441716@qq.com；
4. **严禁抄袭！**

提交方式

使用matrix系统项目页提交，地址：<http://matrix.sysu.edu.cn/course/154/assignment/7105>

(Matrix系统课程页 – [文件上传题]字母画板项目)

提交时间

2019年6月10日 23:59 前

提示

该项目的实现可能需要使用到控制光标位置的函数，这里给出一段示例代码：

```
1  #include<iostream>
2  #include<windows.h>
3  using namespace std;
4
5  //Windows, Dev-C++(MinGW)
6
7  void setpos(int x,int y){
8      COORD point={x,y};
9      HANDLE handle = GetStdHandle(STD_OUTPUT_HANDLE);
10     SetConsoleCursorPosition(handle, point);
11 }
12
13 int main(){
14     setpos(10,5);
15     cout<<"HHH";
16
17     setpos(2,3);
18     cout<<"kk";
19
20     setpos(6,7);
21     cout<<"vvvv";
22
23     //system("cls");//清屏
24
25     return 0;
26 }
```