——****

**生物医学工程专业**

**实验报告**

**实验课程： 计算机图形学与可视化**

**实验题目： 第二次实验**

**班级： 生医2202 姓名： 李卓远**

**学号： 20227291 同组人：**

**实验日期： 2023.11.4 指导教师： 代茵**

**实验成绩（教师签字）：**

**医工学院实验教学中心制**

|  |
| --- |
| 1.采用中点法绘制圆  实验代码  #include **<**iostream**>**  #include **<**graphics**.**h**>  void** midCircle**(int** x0**,** **int** y0**,** **int** R**,** COLORS color**=**RED**){**     **int** x**=**0**,** y**=**R**;**     **double** d **=** 1.25 **-** R**;**     **while(**x **<** y**){**         **if(** d **<** 0**)**             d **=** d**+** 2**\***x **+**3**;**         **else** y**--,** d **=** d **+** 2 **\*** x **-** 2 **\*** y **+** 5**;**         x**++;**                  putpixel**(**x**+**x0**,** y**+**y0**,** color**);**         putpixel**(-**x**+**x0**,** y**+**y0**,** color**);**         putpixel**(**x**+**x0**,** **-**y**+**y0**,** color**);**         putpixel**(-**x**+**x0**,** **-**y**+**y0**,** color**);**         putpixel**(**y**+**y0**,** x**+**x0**,** color**);**         putpixel**(**y**+**y0**,** **-**x**+**x0**,** color**);**         putpixel**(-**y**+**y0**,** x**+**x0**,** color**);**         putpixel**(-**y**+**y0**,** **-**x**+**x0**,** color**);**     **} } int** main**() {**     initgraph**(**800**,**800**);**       *//* *初始化*     midCircle**(**400**,** 400**,** 200**,** WHITE**);**   *//* *画圆*     getch**();**             closegraph**();**            **return** 0**; }**  程序运行    2.采用Sutherland-Cohen 矩形裁剪  实验代码  #include **<**iostream**>** #include **<**graphics**.**h**>** #define LEFT 1 #define RIGHT 2 #define BOTTOM 4 #define TOP 8  **class** Point **{ public:**     **double** x**;**     **double** y**;**          Point**(double** xCoord**,** **double** yCoord**)** **:** x**(**xCoord**),** y**(**yCoord**)** **{} };  struct** Range **{**     **float** left**;**     **float** bottom**;**     **float** right**;**     **float** top**; };  int** clipEncode**(**Point point**,** Range range**)** **{**     **int** code **=** 0**;**     **if** **(**point**.**x **<** range**.**left**)** code **|=** LEFT**;**     **if** **(**point**.**x **>** range**.**right**)** code **|=** RIGHT**;**     **if** **(**point**.**y **<** range**.**bottom**)** code **|=** BOTTOM**;**     **if** **(**point**.**y **>** range**.**top**)** code **|=** TOP**;**     **return** code**; }  void** LineClip**(**Point p1**,** Point p2**,** Range range**)** **{**     setcolor**(**GREEN**);**     **int** codeP1 **=** clipEncode**(**p1**,** range**);**     **int** codeP2 **=** clipEncode**(**p2**,** range**);**          **while** **(true)** **{**         **if** **(!(**codeP1 **|** codeP2**))** **{**             line**(**p1**.**x**,** p1**.**y**,** p2**.**x**,** p2**.**y**);**             **return;**         **}**         **if** **(**codeP1 **&** codeP2**)** **return;**                  **int** codeOut **=** codeP1 **?** codeP1 **:** codeP2**;**         **double** x**,** y**;**                  **if** **(**codeOut **&** TOP**)** **{**             x **=** p1**.**x **+** **(**p2**.**x **-** p1**.**x**)** **\*** **(**range**.**top **-** p1**.**y**)** **/** **(**p2**.**y **-** p1**.**y**);**             y **=** range**.**top**;**         **}** **else** **if** **(**codeOut **&** BOTTOM**)** **{**             x **=** p1**.**x **+** **(**p2**.**x **-** p1**.**x**)** **\*** **(**range**.**bottom **-** p1**.**y**)** **/** **(**p2**.**y **-** p1**.**y**);**             y **=** range**.**bottom**;**         **}** **else** **if** **(**codeOut **&** RIGHT**)** **{**             y **=** p1**.**y **+** **(**p2**.**y **-** p1**.**y**)** **\*** **(**range**.**right **-** p1**.**x**)** **/** **(**p2**.**x **-** p1**.**x**);**             x **=** range**.**right**;**         **}** **else** **if** **(**codeOut **&** LEFT**)** **{**             y **=** p1**.**y **+** **(**p2**.**y **-** p1**.**y**)** **\*** **(**range**.**left **-** p1**.**x**)** **/** **(**p2**.**x **-** p1**.**x**);**             x **=** range**.**left**;**         **}**                  **if** **(**codeOut **==** codeP1**)** **{**             p1**.**x **=** x**;**             p1**.**y **=** y**;**             codeP1 **=** clipEncode**(**p1**,** range**);**         **}** **else** **{**             p2**.**x **=** x**;**             p2**.**y **=** y**;**             codeP2 **=** clipEncode**(**p2**,** range**);**         **}**     **} }  int** main**()** **{**     Point p1 **=** Point**(**100**,** 100**);**     Point p2 **=** Point**(**600**,** 300**);**     Range myRange **=** Range**{**         180**,** 150**,** 480**,** 270     **};**          initgraph**(**640**,** 480**);**          setcolor**(**RED**);**     line**(**p1**.**x**,** p1**.**y**,** p2**.**x**,** p2**.**y**);**     setcolor**(**WHITE**);**     rectangle**(**myRange**.**left**,** myRange**.**bottom**,** myRange**.**right**,** myRange**.**top**);**     LineClip**(**p1**,** p2**,** myRange**);**          getch**();**     closegraph**();**          **return** 0**; }**  实验结果    3.种子填充法  实验代码  #include **<**iostream**>**  #include **<**graphics**.**h**>** #include **<**vector**>  using** **std::**vector**,** **std::**cout**,** **std::**endl**;  class** Point **{ public:**     **double** x**;**     **double** y**;**          Point**(double** xCoord**,** **double** yCoord**)** **:** x**(**xCoord**),** y**(**yCoord**)** **{}**     **friend** **std::**ostream**&** **operator<<(std::**ostream**&** os**,** **const** Point**&** point**)** **{**         os **<<** **"Point("** **<<** point**.**x **<<** **", "** **<<** point**.**y **<<** **")";**         **return** os**;**     **} };  void** seedFill**(**Point p0**,** COLORS colorInner**=**GREEN**){**     vector**<**Point**>** colorStack**;**     colorStack**.**push\_back**(**p0**);**     **long** **unsigned** **int** initColor **=** getpixel**(**p0**.**x**,** p0**.**y**);**     **while(!**colorStack**.**empty**()){**         Point thisPoint **=** colorStack**.**back**();**         colorStack**.**pop\_back**();**                  **if(**getpixel**(**thisPoint**.**x**+**1**,** thisPoint**.**y**)** **==** initColor**){**             colorStack**.**push\_back**(**Point**(**thisPoint**.**x**+**1**,** thisPoint**.**y**));**         **}**         **if(**getpixel**(**thisPoint**.**x**-**1**,** thisPoint**.**y**)** **==** initColor**){**             colorStack**.**push\_back**(**Point**(**thisPoint**.**x**-**1**,** thisPoint**.**y**));**         **}**         **if(**getpixel**(**thisPoint**.**x**,** thisPoint**.**y**+**1**)** **==** initColor**){**             colorStack**.**push\_back**(**Point**(**thisPoint**.**x**,** thisPoint**.**y**+**1**));**         **}**         **if(**getpixel**(**thisPoint**.**x**,** thisPoint**.**y**-**1**)** **==** initColor**){**             colorStack**.**push\_back**(**Point**(**thisPoint**.**x**,** thisPoint**.**y**-**1**));**         **}**                  **if(**getpixel**(**thisPoint**.**x**,** thisPoint**.**y**)** **==** initColor**){**             putpixel**(**thisPoint**.**x**,** thisPoint**.**y**,** colorInner**);** *//*            *Sleep(1); //*            *cout* *<<* *thisPoint* *<<* *endl;*         **}**     **} }   int** main**() {**     initgraph**(**640**,**480**);**          setcolor**(**WHITE**);**      circle**(**200**,** 200**,** 100**);**          seedFill**(**Point**(**200**,**200**),** GREEN**);**          getch**();**     closegraph**();**            **return** 0**; }**  实验结果 |