机器人足球第三次作业

程序代码：

|  |
| --- |
| #include <iostream>  #include <cstring>  #include <sstream>  #include<cmath>  using namespace std;  //点坐标信息类  class Point {  public:  //横纵坐标  double x, y;  //默认构造函数  Point() {  this->x = 0;  this->y = 0;  }  //构造函数  Point(double x, double y) {  this->x = x;  this->y = y;  }  };  //极坐标信息类  class Polar {  public:  //距离  double r;  //角度  double theta;  //默认构造函数  Polar() {  this->r = 0;  this->theta = 0;  }  //构造函数  Polar(double r, double theta) {  this->r = r;  this->theta = theta;  }  };  //获取目标坐标类  class GetCoordinate {  public:  //两条已知信息  string strf, strg;  //两个已知点  Point pf, pg;  //极坐标信息  Polar prf, prg;  //其他基本信息  double rf, rg, thetaf, thetag;  double xp, yp;  //构造函数  GetCoordinate(string strf, string strg) {  this->strf = strf;  this->strg = strg;  }  //解析操作  void init();  //计算出目标坐标  void calculate();  };  void GetCoordinate::init() {  //已知信息  Point C(0, 0), P1(-52.5, -32), P2(-52.5, 32), P3(52.5, 32), P4(52.5, -32), P5(0, -32), P6(0, 32), P7(-30, -7), P8(-30, 7), P9(30, 7), P10(30, -7), G1(-52.5, 0), G2(52.5, 0);  //处理字符串  int s1[20], s2[20];  int m1 = 0, m2 = 0;  //找出字符串1和字符串2中空格的位置  for (int i = 0; i < strf.size(); i++) {  if (strf[i] == ' ') {  s1[m1] = i;  m1++;  }  }  for (int i = 0; i < strg.size(); i++) {  if (strg[i] == ' ') {  s2[m2] = i;  m2++;  }  }  //确定是哪两个目标点  string mesf1 = strf.substr(1, s1[0] - 1);  string mesf2 = strg.substr(1, s2[0] - 1);  //用if-else语句来判断是哪个点  //并创建对应目标点对象  if (mesf1 == "C") {  pf = C;  }  else if (mesf1 == "P1") {  pf = P1;  }  else if (mesf1 == "P2") {  pf = P2;  }  else if (mesf1 == "P3") {  pf = P3;  }  else if (mesf1 == "P4") {  pf = P4;  }  else if (mesf1 == "P5") {  pf = P5;  }  else if (mesf1 == "P6") {  pf = P6;  }  else if (mesf1 == "P7") {  pf = P7;  }  else if (mesf1 == "P8") {  pf = P8;  }  else if (mesf1 == "P9") {  pf = P9;  }  else if (mesf1 == "P10") {  pf = P10;  }  else if (mesf1 == "G1") {  pf = G1;  }  else if (mesf1 == "G2") {  pf = G2;  }  if (mesf2 == "C") {  pg = C;  }  else if (mesf2 == "P1") {  pg = P1;  }  else if (mesf2 == "P2") {  pg = P2;  }  else if (mesf2 == "P3") {  pg = P3;  }  else if (mesf2 == "P4") {  pg = P4;  }  else if (mesf2 == "P5") {  pg = P5;  }  else if (mesf2 == "P6") {  pg = P6;  }  else if (mesf2 == "P7") {  pg = P7;  }  else if (mesf2 == "P8") {  pg = P8;  }  else if (mesf2 == "P9") {  pg = P9;  }  else if (mesf2 == "P10") {  pg = P10;  }  else if (mesf2 == "G1") {  pg = G1;  }  else if (mesf2 == "G2") {  pg = G2;  }  //分割出r与theta  string srf = strf.substr(s1[0] + 1, s1[1] - s1[0] - 1);  string srg = strg.substr(s2[0] + 1, s2[1] - s2[0] - 1);  string stf = strf.substr(s1[1] + 1, strf.size() - s1[1] - 2);  string stg = strg.substr(s2[1] + 1, strg.size() - s2[1] - 2);  //将字符串转化为double类型数据  stringstream stream1, stream2, stream3, stream4;  stream1 << srf;  stream1 >> rf;  stream2 << srg;  stream2 >> rg;  stream3 << stf;  stream3 >> thetaf;  stream4 << stg;  stream4 >> thetag;  //创建两个极坐标对象  prf = Polar(rf, thetaf);  prg = Polar(rg, thetag);  }  void GetCoordinate::calculate() {  //书上的计算公式存在点问题  /\*double d = sqrt((pf.x - pg.x) \* (pf.x - pg.x) + (pf.y - pg.y) \* (pf.y - pg.y));  double a = (prf.r \* prf.r - prg.r \* prg.r + d \* d) / (2 \* d);  double dx, dy;  dx = abs(pg.x - pf.x);  dy = abs(pf.y - pg.y);  double xp1 = pf.x + a \* dx / d;  double yp1 = pf.y + a \* dy / d;  double h = sqrt(prf.r \* prf.r - a \* a);  int sign;  if (prg.theta - prf.theta > 0) {  sign = 1;  }  else {  sign = -1;  }  xp = xp1 - h \* sign \* dy / d;  yp = yp1 + h \* sign \* dx / d;  \*/  //计算目标坐标  double a1, a2, a3, b1, b2, b3, c3, tv;  a1 = (prf.r \* prf.r - prg.r \* prg.r) + (pg.x \* pg.x - pf.x \* pf.x) + (pg.y \* pg.y - pf.y\* pf.y);  a2 = a1 / (2 \* (pg.y - pf.y));  b2 = (pg.x - pf.x) / (pg.y - pf.y);  a3 = 1 + b2 \* b2;  b3 = (-1) \* (2 \* pf.x + 2 \* (a2 - pf.y) \* b2);  c3 = pf.x \* pf.x + pow((a2 - pf.y), 2) - prf.r \* prf.r;  double xp1 = (-b3 + sqrt(b3 \* b3 - 4 \* a3 \* c3)) / (2 \* a3);  double yp1 = a2 - b2 \* xp1;  xp = (-b3 - sqrt(b3 \* b3 - 4 \* a3 \* c3)) / (2 \* a3);  yp = a2 - b2 \* xp;  tv = prg.theta - prf.theta;  if (tv < 0) {  if (yp == -0) {  yp == 0;  }  if (xp == -0) {  xp == 0;  }  cout << "机器人自身坐标点为：(" << xp << "," << yp << ')' << endl;  }  else {  if (yp1 == -0) {  yp1 == 0;  }  if (xp1 == -0) {  xp1 == 0;  }  cout << "机器人自身坐标点为：(" << xp1 << "," << yp1 << ')' << endl;  }  }  //示例  int main() {  string str1 = "(P8 22 0)";  string str2 = "(P7 27.7 30)";  cout << "已知信息为：" << endl;  cout << "str1" << str1 << endl;  cout << "str2" << str2 << endl;  GetCoordinate res = GetCoordinate(str1, str2);  res.init();  res.calculate();  return 0;  } |

程序运行结果：

