

# 测量程序设计

---

*C#-WPF*

作者：朱学军

xazhuxj@qq.com

西安科技大学

测绘科学与技术学院

二零一八年三月

本文档仅适用于西安科技大学测绘专业本科三年级学生教学。

也可用于其他专业人员参考。

内部文档，版权所有 © 2018，朱学军。

未经作者本人授权，严谨出版发行！

# 教学计划及课程目标

## 教学计划

### 教学课时

该课程标准学时为 64 学时。本学期为 60 学时，上课 15 次 30 学时，上机 15 次 30 学时。必修课，考试。笔试（60%）+ 上机实验与作业（40%）

### 学习内容

学习面向对象的程序设计与编写；  
学习基本的测量算法程序编写，学习软件界面的编写。

## 编程预言的选择

注意：该课程不是再次学习某种编程语言，而是运用某种编程语言进行测量数据处理及测量算法的编写。

可能大家只学习了 C 语言，C 语言是面向过程的功能强大的语言，执行效率高，但界面程序编写较复杂。现代编程语言更多的是面向对象的编程语言，高效而且兼容 C 语言的是 C++，但界面的编写仍较复杂。因此我们选择更加现代化的编程语言 C#，它语法优美，界面编写方式有 WinForm 与 WPF 方式，虽然执行效率没有 C++ 高，但在 Windows7、Windows8/8.1、Windows10 中都几乎预装了运行库 .Net Frame，并且得到了 Microsoft 的大力推广，Microsoft 的许多软件都在使用 C# 开发，学习与开发成本相比与 C++ 显著减少，效率显著提升。

C# 的语法与 JAVA 的许多语法是极其相似的，现代软件工程的许多方法也可以用 C# 去实现。因此，在本课程中我们将学习 C# 编程语言，并且会学习如何运用 C# 语言进行测量程序设计及测量数据处理。

现代软件基本上都具有简洁易用的界面，我们还将学习 WPF 的界面编写技术，为我们的程序设计处美观简洁的界面，这些都包含在 C# 之中。

## 编程环境与工具

本课程的基本编程工具为 Visual Studio，版本理论上为 2010 及更新版，推荐大家用 Visual Studio 2015 或 2017 版（目前的最新版）。

## 参考教材

C# 语言及 WPF 界面编写参考马骏主编，人民邮电出版社出版，《C# 程序设计及应用教程》第 3 版。

测量算法参考本教程（一直更新中.....）。

# 目录

|                                    |     |                                |   |
|------------------------------------|-----|--------------------------------|---|
| 教学计划及课程目标                          | i   | 2.1.1 值类型 . . . . .            | 7 |
| 目录                                 | iii | 2.1.2 引用类型 . . . . .           | 7 |
|                                    |     | 2.1.3 装箱与拆箱 . . . . .          | 7 |
| 第一章 从 C 走向 C#                      | 1   | 2.1.4 struct 与 class . . . . . | 7 |
| 1.1 Client/Server 程序设计模式 . . . .   | 1   | 2.2 C# 中的数组 . . . . .          | 7 |
| 1.2 从面向过程走向面向对象的程<br>序设计 . . . . . | 1   | 2.3 C# 中的条件语句 . . . . .        | 7 |
| 1.2.1 代码示例 . . . . .               | 1   | 2.3.1 if 语句 . . . . .          | 7 |
| 1.2.2 代码分析 . . . . .               | 5   | 2.3.2 switch 语句 . . . . .      | 7 |
| 1.3 C# 程序的组织结构 . . . . .           | 5   | 2.4 C# 中的循环语句 . . . . .        | 7 |
| 第二章 C# 语言基础                        | 7   | 2.4.1 for 语句 . . . . .         | 7 |
| 2.1 C# 中的数据类型 . . . . .            | 7   | 2.4.2 foreach 语句 . . . . .     | 7 |
|                                    |     | 2.4.3 while 与 do while 语句 . .  | 7 |



# 第一章 从 C 走向 C#

## 1.1 Client/Server 程序设计模式

改变 C 语言中将代码写入 main 函数中的习惯；

改变 C# 语言中将代码写入 Main 函数中的习惯；

改变在 WinForm 中直接写入代码的习惯；

以上这些习惯将带来一系列的问题,在团队开发与多人协作尤其如此,如:不能进行 unittest(单元测试)、git (著名的源代码管理工具) 代码合并时会引发大量的冲突。

## 1.2 从面向过程走向面向对象的程序设计

良好的面向过程设计程序设计程序是可以很好的转向面向对象的程序设计的,我们将从一个简单的 C 程序开始设计结构较为良好的 C 代码,再将其用面向对象的 C# 进行实现。从中体会面向过程与面向对象的程序设计方法的不同。

### 1.2.1 代码示例

在测绘专业中我们经常与测量点打交道,因此我们定义一个点 Point (测点除了它的坐标 x,y 和高程 z 之外,还需有点名),另外我们定义一个简单的为大家所熟知的圆 Circle,来使用点 Point,并实现判断两圆是否相交,计算圆的面积和周长等功能。代码如下所示:

```
1 // Ch01Ex01.cpp : Defines the entry point for the console application.
2 //
3
4 #include "stdafx.h"
5
6 #define _USE_MATH_DEFINES
7 #include <math.h>
8 #include <string.h>
9 #define PI M_PI
10
11 typedef struct _point {
12     char name[11];
13     double x, y, z;
14 }Point;
15
16 typedef struct _circle {
17     Point center;
18     double r;
19     double area;
20     double length;
21 }Circle;
22
```

```

23 // 计算圆的面积
24 double Area(Circle * c) {
25     return PI * c->r * c->r;
26 }
27
28 // 计算两点的距离
29 double Distance(Point * p1, Point * p2) {
30     double dx = p2->x - p1->x;
31     double dy = p2->y - p1->y;
32     return sqrt(dx*dx + dy * dy);
33 }
34
35 // 判断两圆是否相交
36 bool IsIntersectWithCircle(Circle * c1, Circle * c2) {
37     double d = Distance(&c1->center, &c2->center);
38     return d <= (c1->r + c2->r);
39 }

```

相应的 main 函数测试代码如下：

```

1 int main()
2 {
3     Point pt1;
4     strcpy_s(pt1.name, 11, "pt1");
5     pt1.x = 100; pt1.y = 100; pt1.z = 425.324;
6
7     Point pt2;
8     strcpy_s(pt2.name, 11, "pt2");
9     pt2.x = 200; pt2.y = 200; pt2.z = 417.626;
10
11     Circle c1;
12     c1.center = pt1; c1.r = 80;
13
14     Circle c2;
15     c2.center = pt2; c2.r = 110;
16
17     printf("Circle1 的面积 = %lf \n", Area( &c1 ));
18
19     bool yes = IsIntersectWithCircle(&c1, &c2);
20     printf("Circle1 与 Circle2 是否相交 : %s\n", (yes ? "是" : "否"));
21
22     return 0;
23 }

```

程序的运行结果如下：

Circle1 的面积 = 20106.192983

Circle1 与 Circle2 是否相交 : 是

与以上代码相对应的 C# 代码为：

```

1 using System;
2
3 namespace Ch01Ex02
4 {
5     class Point
6     {

```



```
7     private string name;
8     public string Name
9     {
10         get { return name; }
11         set { name = value; }
12     }
13
14     private double x;
15     public double X
16     {
17         get { return x; }
18         set { x = value; }
19     }
20     private double y;
21     public double Y
22     {
23         get { return y; }
24         set { y = value; }
25     }
26     private double z;
27     public double Z
28     {
29         get { return z; }
30         set { z = value; }
31     }
32
33     public Point()
34     {
35         Name = "";
36         x = y = z = 0;
37     }
38     //函数重载：函数名称一样，但参数（个数或类型）不一样
39     public Point(string name, double x, double y, double z)
40     {
41         this.name = name;
42         this.x = x;
43         this.y = y;
44         this.z = z;
45     }
46
47
48     /// <summary>
49     /// 计算当前点至p2点的距离
50     /// </summary>
51     /// <param name="p2">目标点</param>
52     /// <returns>两点的距离</returns>
53     public double Distance(Point p2)
54     {
55         double dx = X - p2.X;
56         double dy = Y - p2.Y;
57         return Math.Sqrt(dx * dx + dy * dy);
58     }
59 }
60
61 class Circle
62 {
```

```
63     private Point center;
64     private double r;
65
66     private double area;
67     public double Area
68     {
69         get { return area; }
70     }
71
72     private double length;
73
74     public Circle()
75     {
76
77     }
78
79     public Circle(string centerName, double x, double y, double z, double r)
80     {
81         center = new Point(centerName, x, y, z);
82         this.r = r;
83
84         CalArea();
85     }
86
87
88     /// <summary>
89     /// 计算圆的面积
90     /// </summary>
91     private void CalArea()
92     {
93         area = Math.PI * r * r;
94     }
95
96     //判断两圆是否相交
97     public bool IsIntersectWithCircle(Circle c2)
98     {
99         double d = this.center.Distance(c2.center);
100         return d <= (r + c2.r);
101     }
102 }
103 }
```

相应的 C# 的 Main 函数测试代码如下:

```
1 using System;
2
3 namespace Ch01Ex02
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             Circle c1 = new Circle("pt1", 100, 100, 425.324, 80);
10            Circle c2 = new Circle("pt2", 200, 200, 417.626, 110);
11
12            Console.WriteLine("Circle1 的面积={0}", c1.Area );
13            Console.WriteLine("Circle2 的面积={0}", c2.Area);
14        }
15    }
16 }
```

```
15         bool yes = c1.IsIntersectWithCircle(c2);
16         Console.WriteLine("Circle1 与 Circle2 是否相交:{0}",
17             yes ? "是" : "否" );
18     }
19 }
20 }
```

程序的运行结果如下：

Circle1的面积=20106.1929829747

Circle2的面积=38013.2711084365

Circle1与Circle2是否相交:是

### 1.2.2 代码分析

C 语言中有 struct，在 C++ 中对 struct 进行了扩展，其实质就是一 class。C 与 C++ 中又特别是 C 中，struct 与指针是其根本。

类 (class) 是绝大多数的面向对象程序设计语言的关键词。

程序或软件的基本概念是：程序 = 数据结构 + 算法

从程序设计语言的语法角度分析，class 是数据与函数的复合体，是符合上述程序设计原则的。

## 1.3 C# 程序的组织结构

从以上 C# 示例代码可以看出，C# 是一纯面向对象语言，其组织程序的基本单位是类 (class)。在 class 之外是 namespace，在 class 内可以定义类成员 (class member) 和类方法 (class method)。在 method 中可以定义语句。

Visual Studio 用 Solution(.sln)、Project(.csproj) 与 File(.cs) 及文件夹进行组织。其形式如下：

Solution - 1..m - Project - 1..m - File

一个 Project 对应一个 Exe 文件或 DLL 文件或其它模块，是 class 等的功能组合。



## 第二章 C# 语言基础

由于我们都学习过 C 语言，在此我们主要讲解 C# 中不同于 C 的一些基本语法。

### 2.1 C# 中的数据类型

C# 中的数据类型可分为两类，一类为值类型，一类为引用类型。

#### 2.1.1 值类型

#### 2.1.2 引用类型

#### 2.1.3 装箱与拆箱

#### 2.1.4 struct 与 class

### 2.2 C# 中的数组

### 2.3 C# 中的条件语句

#### 2.3.1 if 语句

#### 2.3.2 switch 语句

### 2.4 C# 中的循环语句

#### 2.4.1 for 语句

#### 2.4.2 foreach 语句

#### 2.4.3 while 与 do while 语句