**C#.Net 程序设计**

**一、基本概念要点（理解基本含义，及能看懂例子程序）**

1. 变量标识符的命名规则：

* 变量名应以字母、下划线或@开头，后续可包含字母、数字、下划线或@。
* 变量名不能与C#关键字冲突。
* 建议使用有意义的名称，避免使用单个字符的变量名，除非在循环中作为计数器。
* 遵循驼峰命名法（lowerCamelCase）或者帕斯卡命名法（UpperCamelCase），其中驼峰命名法用于局部变量和方法参数，帕斯卡命名法用于类名和属性名。

2. Convert类的使用：

* Convert类提供了各种类型之间的转换方法，如ToInt32、ToString、ToDouble等。
* 例如，将字符串转换为整数：int value = Convert.ToInt32("123");
* 或者将整数转换为字符串：string strValue = Convert.ToString(123);

3. switch, for, while, foreach语句的使用：

- switch语句用于根据一个表达式的值执行不同的代码块。例如：

```csharp

int day = 5;

switch (day)

{

case 1:

Console.WriteLine("Monday");

break;

case 2:

Console.WriteLine("Tuesday");

break;

// ...

}

```

- for语句用于执行特定次数的循环。例如：

```csharp

for (int i = 0; i < 10; i++)

{

Console.WriteLine(i);

}

```

- while语句用于在满足某个条件时重复执行一段代码。例如：

```csharp

int i = 0;

while (i < 10)

{

Console.WriteLine(i);

i++;

}

```

- foreach语句用于遍历集合中的元素。例如：

```csharp

string[] days = { "Monday", "Tuesday", "Wednesday" };

foreach (string day in days)

{

Console.WriteLine(day);

}

```

4. break, continue语句的使用，i++ 和++i的区别；常用运算符的优先级：

- break语句用于退出当前循环。

- continue语句用于跳过当前循环的剩余部分并进入下一轮循环。

- i++ 和++i都是自增运算符，它们都会使i的值增加1。但是，i++ 是先使用i的当前值，然后将其加1；而++i是先将i加1，然后使用新的值。

- 常用运算符的优先级从高到低大致如下：()`[]` -> . -> ++/-- -> \* / % -> + - -> << >> -> < > <= >= -> == != -> & -> ^ -> | -> && -> || -> ? : -> = += -= \*= /= %= &= ^= |= <<= >>= -> ,

5. 异常的概念以及异常处理：try，catch，finally语句的意义及用法：

- 异常是在程序运行过程中发生的错误或意外情况。

- try/catch语句用于捕获和处理异常。try块包含可能抛出异常的代码，catch块则包含处理特定类型异常的代码。

- finally块包含必须在任何情况下都要执行的代码，无论是否发生异常。

- 例如：

```csharp

try

{

// 可能抛出异常的代码

}

catch (Exception ex)

{

// 处理异常的代码

Console.WriteLine($"An error occurred: {ex.Message}");

}

finally

{

// 必须执行的清理代码

Console.WriteLine("Cleaning up...");

}

```

6. 引用类型和值类型，它们的区别：

- 值类型包括简单类型（如int、double、bool等）和结构体。值类型的变量直接存储其值，复制时会创建一个新的副本。

- 引用类型包括类、接口、数组和委托。引用类型的变量存储的是对对象的引用（地址），复制时只是复制了引用，因此两个变量实际上指向同一个对象。

- 主要区别在于：值类型变量直接包含其数据，而引用类型变量存储的是对数据的引用。值类型变量在赋值或传递给函数时会复制其值，而引用类型变量只会复制其引用。值类型通常更快，因为它们存储在栈上，而引用类型存储在堆上。

**二、读程序能力要点**

1. **string 类**的使用；

2. **引用类型，值类型的区别；**

3. **一维数组；二维数组**的用法；

4. **List< >类**的用法；

5. **类的继承**；

6. **String 类与 StringBuilder类**的区别；

7. **类中virtual 和 override 方法的多态**；

8. **抽象类、接口实现多态；**

**三、程序编写能力要点**

（1）程序补充完善

1. **string 类的用法**;

2. **List< >类的用法**；

3. **类的属性的实现；**

**4. 类的方法参数中ref的使用；**

5. **一维数组；二维数组**的用法；

**6. 多态的实现与使用；**

（2）程序编写能力考查

1. 计算1-5000之间的质数；

2. 计算1-5000之间整数的和；

3. 计算1-5000之间整数能被3整除的数之和；

4. 计算；

5. 输入5000个数，计算并输出最大的数；

6. 输入5000个数进行排序；

7. 输入两个集合，计算两个集合之间各种运算；

8. 输入字符串，输出其中字母，数字，空格等字符的数量；

9. 计算出4位数的水仙花数；

10. 输入5000个整数，计算并输出平均值和方差；

11. 输入两个整数，计算并输出它们的最大公约数；

12. 实现一个类表示圆，实现**属性**读取改写圆心坐标、半径等属性；只读属性圆面积；按圆面积用比较大小；