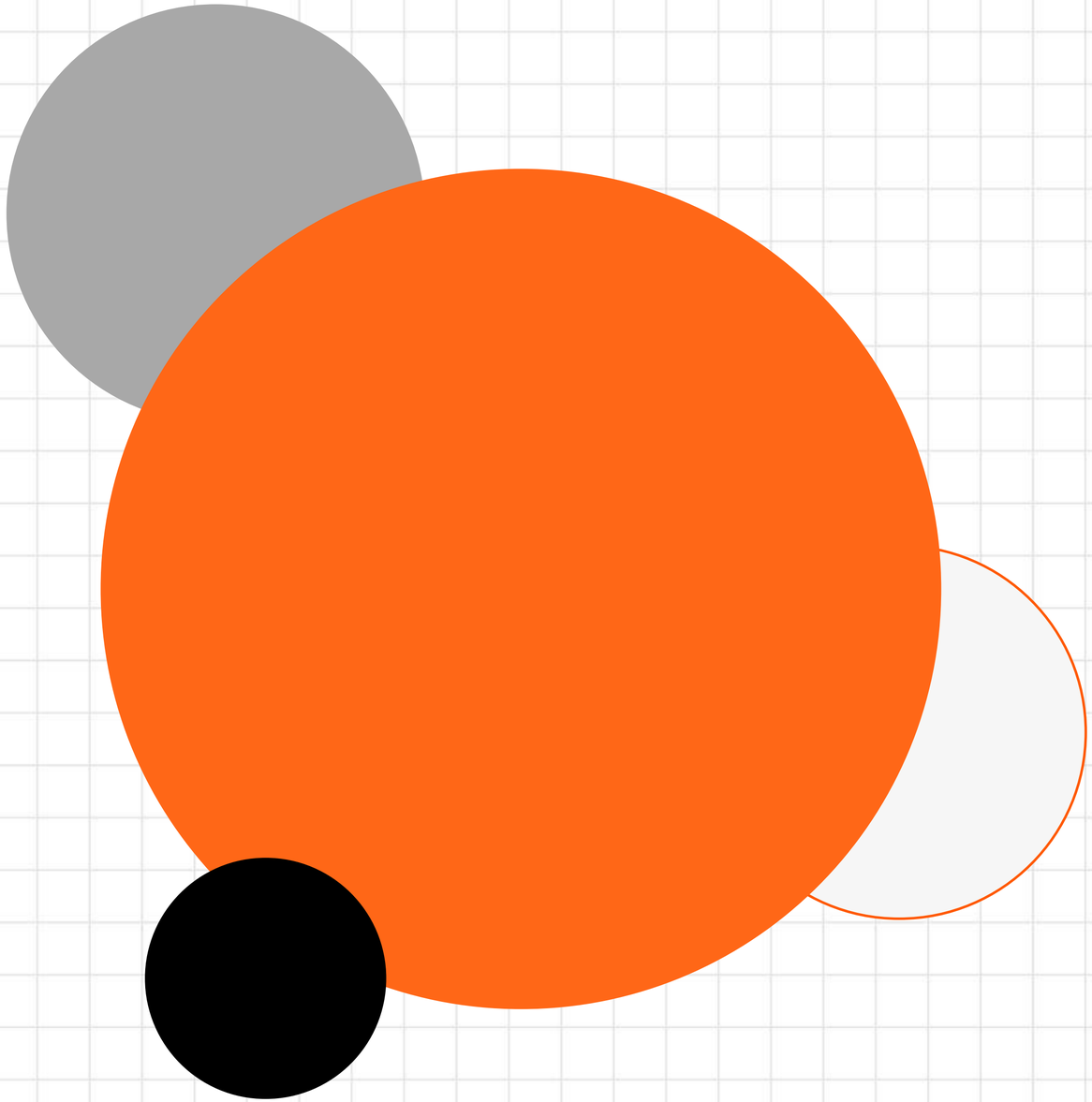
C#
代码规范

C# Code Rules

张文杰
ECADI
2023-09-01

目录 CONTENTS



PART 01

规范原因

WHY

01

代码规范的好处
BENIFIT

01
提高代码可读性

-一致的命名约定、缩进风格和组织方式使代码易于阅读，减少学习成本。

- 减少代码审查时间，促进团队合作。

02
提高代码可维护性

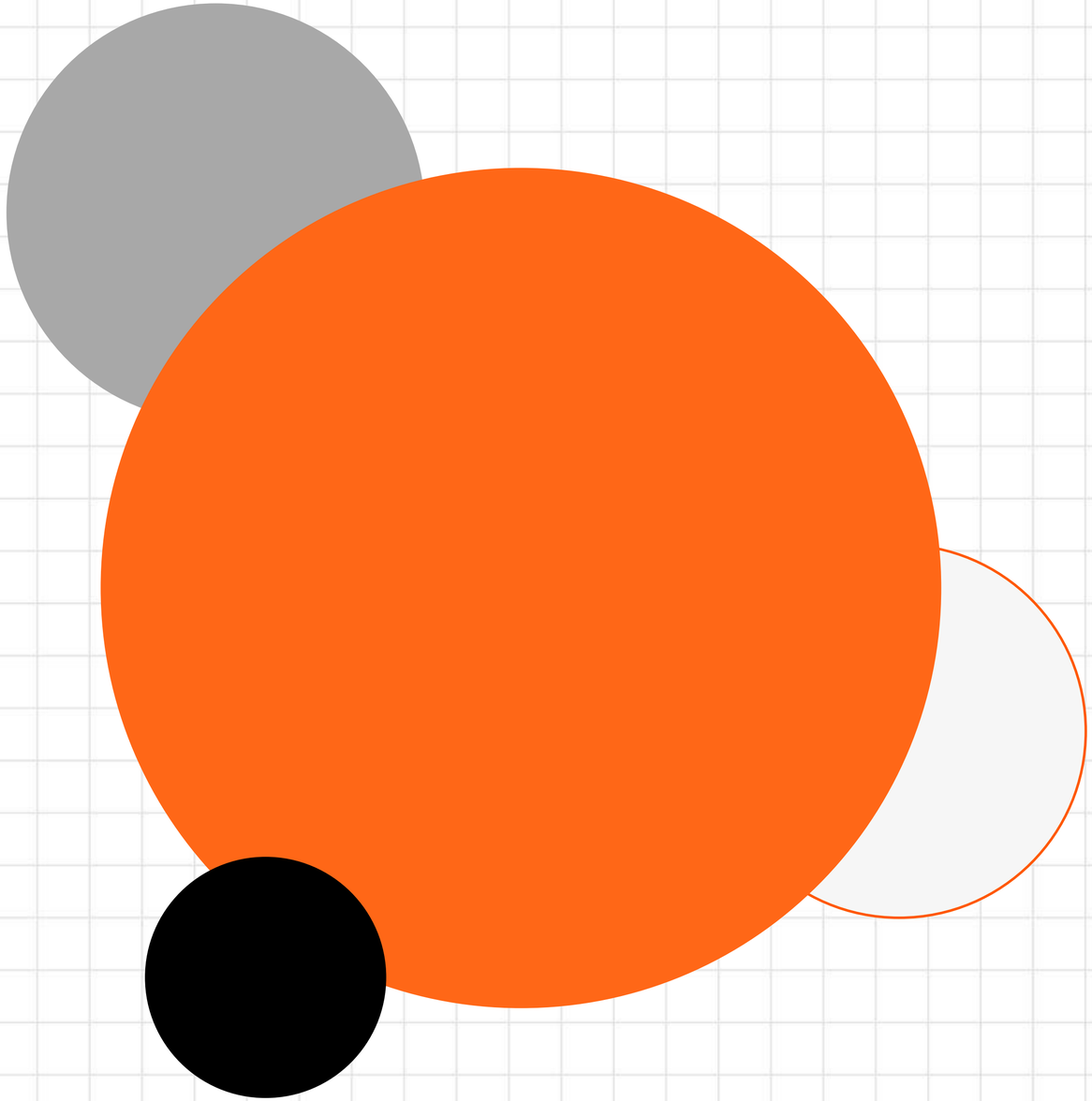
-良好的代码规范鼓励编写模块化、可扩展和可重用的代码。

- 适当的注释、合理的函数和类划分以及一致的命名约定使代码易于理解和修改。

03
提高代码质量和可靠性

-避免常见的编程错误和潜在问题，编写更可靠的代码。

- 错误处理、安全性和性能最佳实践确保代码在各种情况下都能正常运行。



PART 02

变量命名

NAMING

变量命名规范有助于提高代码的可读性、可维护性和可理解性，促进团队合作，减少学习成本，避免命名冲突，并支持代码重用和扩展。

02

GAODING.COM

常见命名方法

大驼峰

SuperMan

各单词首字母大写

小驼峰

superMan

首字母小写
各单词首字母大写

常量

SUPER_MAN

全大写
单词间使用下划线连接

全小写

superman

全小写
单词不分隔

蛇形

super_man

全小写
单词间使用下划线连接

C#标准中的命名方式

大驼峰

SuperMan

各单词首字母大写

小驼峰

superMan

首字母小写
各单词首字母大写

常量

SUPER_MAN

全大写
单词间使用下划线连接

命名方式对应场景

大驼峰命名场景	小驼峰命名场景	常量命名
类、结构体、接口	字段	常量
方法名	变量	临时常量
	参数名	

一些特殊前后缀

接口	异常类	测试类	设计模式
名称前：I	名称后：Exception	名称后：Test	名称后：设计模式
ICoder	ValueException	CatTest	OrderFactory

什么时候加 s、用复数

变量

students ==> studentList

students ==> studentCount

枚举类 Enum

默认枚举值不加 s

指定枚举值加 s

可读性标准

变量拼写

正确使用英语拼写

禁止使用拼音、英语混写

特殊场景可用中文代替拼音

语义化（讲人话）

禁止使用随意命名

看不懂的词作变量（如：你的宠物名）

与上下文无关的简写（如 int α = 1）

命名结构

方法

修饰词 + 名词

正确：getStudentCount

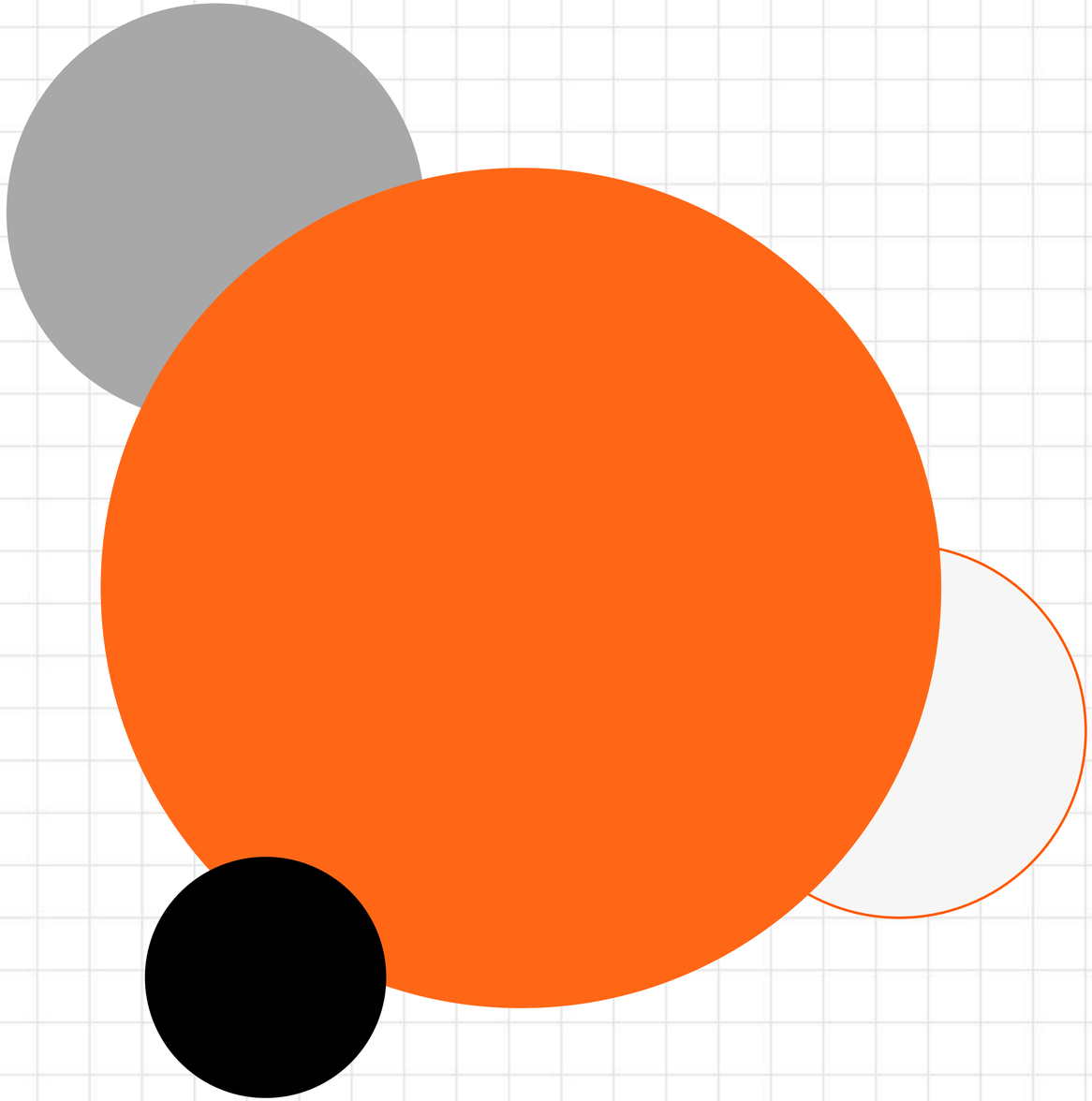
错误：studentCountGet

变量

名词 + 修饰词

正确：studentCount

错误：countOfStudent



PART 03

Lint

CODE LINT

03

GAODING.COM

代码块格式

大括号，两个均垂直左对齐

正确

```
// lint
0个引用
class Mouse
{

}
```

错误

```
0个引用
class MouseRed{

}

0个引用
class MouseBlue{}
```

换行格式规范

正确

1. 第二行开始悬挂缩进
2. 运算符、点符号优先换行
3. 多参数换行，在逗号后换行

```
0 个引用
public void SomeMethod()
{
    string message = "This is a long message that exceeds "
                     + "the maximum character limit.";

    if (result > 0
        && message.Length > 100)
    {
        Console.WriteLine("Both conditions are met.");
    }

    SomeObject obj = new SomeObject();
    obj.Method1()
       .Method2()
       .Method3();

    AnotherMethod(
        "parameter1",
        "parameter2",
        "parameter3"
    );
}
```

其他lint规范

1. if/for等保留字与括号之间都必须加空格
2. 使用空格缩进代码（编辑器功能）
3. 适当增加空行，增加代码分块可读性
4. 不需要为了美观强行给代码加空格对齐

业务规范

控制语句的逻辑规范

1. 每个判断执行的内容，如果存在
   必须显性提供return、continue、break表名逻辑
2. switch语句必须添加default语句，即使是空代码
3. if/else/for/while/do 语句中必须使用大括号。

业务规范

控制语句的嵌套规范

1. 条件、循环嵌套不允许超过三次
   （改进方式：条件判断抽离、辅助函数、数据结构）
2. 条件判断中，异常分支单独拆分，不是用else嵌套
3. 禁止回调地狱，异步编程使用async/await 处理
4. 条件判断语句复杂时，使用可读性高的变量代替

业务规范

异常处理

1. 不要把try/catch 当 if/else 使用
2. 如果调试时，明确知道Exception类型
   在catch 中表名（catch ValueException）
3. 编写try/catch 调试时需尽量缩小范围

业务规范

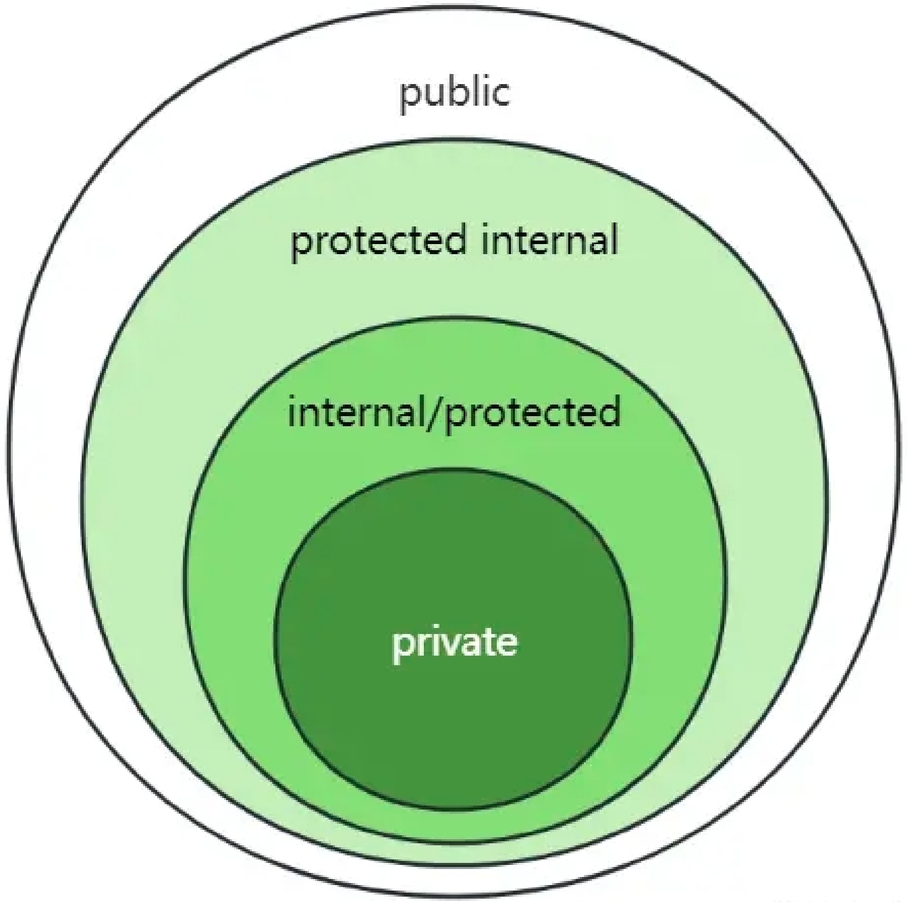
异常处理

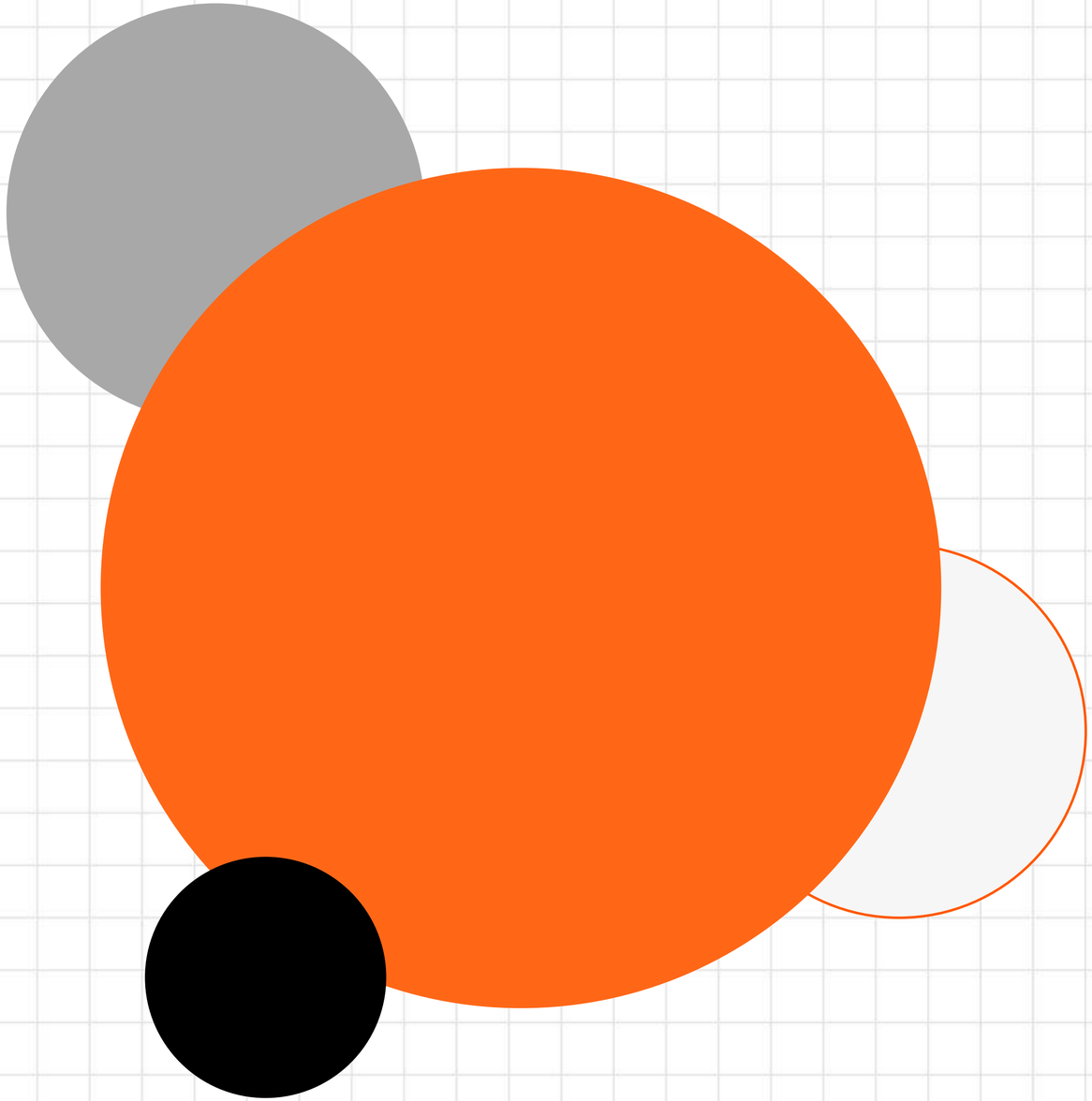
程序集：家
classA：丈夫（在家）
classB：室友（在家）
classC：儿子（在家）
classD：朋友（不在家）

丈夫有个方法：变身蜘蛛侠

这个方法各个情况声明时
谁都知道？

Public	所有人
Private	A
protected	A、C、D
internal	A、B、C





PART 04

注释规范

COMMENT

04

什么情况需要注释

注释什么内容

方法

返回值
参数
功能说明
异常说明

代码

todo
fixme
特殊操作
长代码逻辑
行为动机

变量

枚举类每个值
特殊变量

注释规范

语言

团队指定

注释复杂度

精简准确

只讲相关内容

不滥用注释

修改

代码改则注释改

注释规范

语言

团队指定

注释复杂度

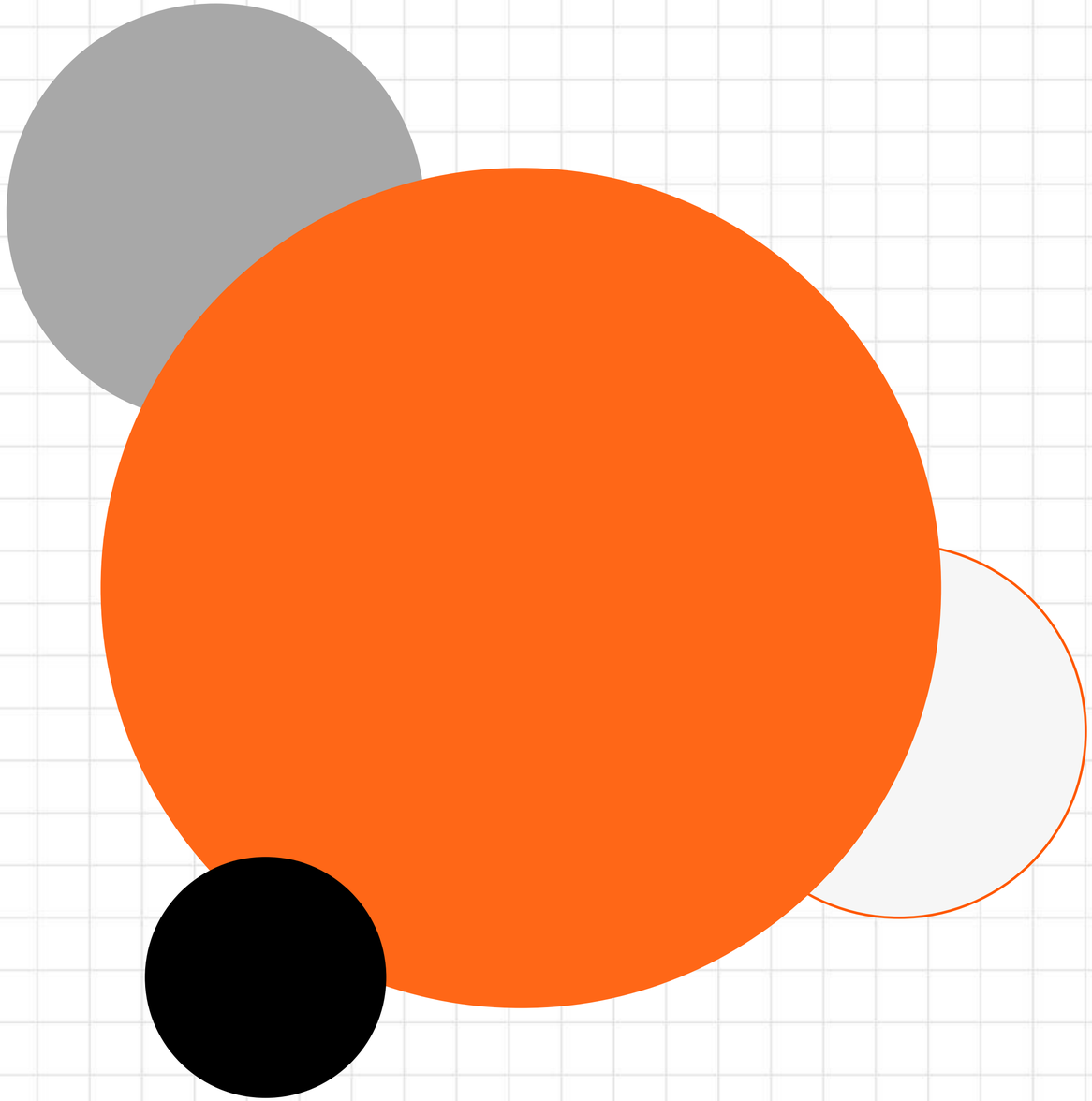
精简准确

只讲相关内容

不滥用注释

修改

代码改则注释改

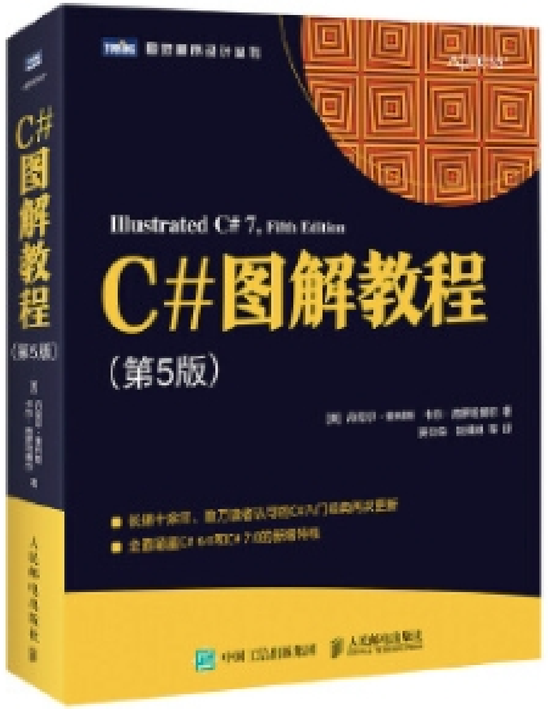


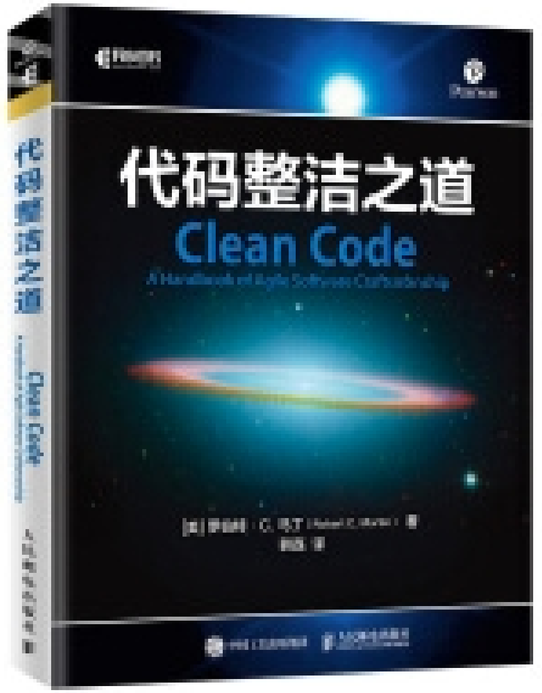
PART 04

推荐资料

04

书籍推荐


C#图解教程
（第5版）
Illustrated C# 7, Fifth Edition
C#图解教程
（第5版）


代码整洁之道
Clean Code

谢谢

THANKS FOR
WATCHING

GAODING.COM