编码规范说明书

题 目： 背包问题知识社区系统

团队名称： 夏日限定

学生姓名： 阮凯 杨凯 潘晴 孟姣姣

指导教师： 代祖华

**目录**

[第一部分 注释 1](#_Toc1847)

[1.1 注释规范 1](#_Toc12297)

[1.2 类属性注释 1](#_Toc30780)

[1.3 方法注释 1](#_Toc22618)

[1.4 代码间注释 2](#_Toc19404)

[第二部分 命名总体规则 3](#_Toc15427)

[2.1 变量命名 3](#_Toc14311)

[2.2 函数命名 3](#_Toc5518)

[2.3 类（Class）命名 3](#_Toc9173)

[2.4 接口（Interface）命名 4](#_Toc30844)

[2.5 命名风格以及约束 4](#_Toc23680)

[第三部分 编码规则 5](#_Toc10555)

[3.1 大括号规则 5](#_Toc15524)

[3.2 小括号规则 5](#_Toc10706)

[3.3 缩进规则 5](#_Toc28537)

[3.4 If Then Else规则 6](#_Toc11424)

[3.5 比较规则 7](#_Toc27080)

[3.6 Case规则 7](#_Toc14243)

[3.7 对齐规则 7](#_Toc21643)

[3.8 单语句规则 7](#_Toc26397)

[3.9 单一功能规则 7](#_Toc15738)

[3.10 简单功能规则 8](#_Toc29274)

[3.11 错误检查规则 8](#_Toc17730)

[第四部分 编程准则 9](#_Toc27552)

[4.1 变量使用 9](#_Toc56)

[4.2 数据库操作 9](#_Toc15497)

[4.3 对象使用 9](#_Toc9659)

[4.4 模块设计原则 9](#_Toc10849)

[4.5 结构化要求 10](#_Toc19375)

[4.6 函数返回值原则 10](#_Toc19001)

[第五部分 数据库命名规范 11](#_Toc5805)

[第六部分 其他准则 12](#_Toc19151)

[6.1 每行最多字符数 12](#_Toc19678)

[6.2 操作符前后空格 12](#_Toc27262)

[6.3 操作符前后空格 12](#_Toc26790)

# 第一部分 注释

## 1.1 注释规范

1、变量注释和变量在同一行，与变量分开至少四个“空格”键；

2、将复杂的注释放在函数头；

3、注释与所描述内容进行同样的缩排；

4、注释掉的代码要配合相关说明。

5、每行注释的最大长度为100个字符；

6、将注释与注释分隔符用一个空格分开；

7、程序段或语句的注释在程序段或语句的上一行；

8、注释和代码同时更新，不用的注释删除；

9、重要变量必须有注释；

## 1.2 类属性注释

在类的属性必须以以下格式编写属性注释：

            /// <summary>

            /// <Properties depiction>

/// </summary>

## 1.3 方法注释

在类的方法声明前必须以以下格式编写注释

             /// <summary>

             /// depiction：<对该方法的说明>

             /// </summary>

             /// <param name="<参数名称>"><参数说明></param>

                 /// <returns>

                 ///<对方法返回值的说明，该说明必须明确说明返回的值代表什么含义>

             /// </returns>

             ///Writer：作者中文名

             ///Create Date：<方法创建日期，格式：YYYY-MM-DD>

## 1.4 代码间注释

代码间注释分为单行注释和多行注释：

//<单行注释>

           /\*多行注释1

             多行注释2

             多行注释3\*/

代码中遇到语句块时必须添加注释（if,for,foreach,……）,添加的注释必须能够说明此语句块的作用和实现手段（所用算法等等）。

# 第二部分 命名总体规则

## 2.1 变量命名

1、不以下划线或美元符号开始或结束；

2、不使用拼音与英文混合的方式；

3、方法名、参数名、成员变量、局部变量都使用lowerCamelCase(第一个词的首字母小写，以及后面每个词的首字母大写)风格，遵从驼峰形式。

**注：**变量名和常量名最多可以包含255个字符，但是，超过25到30个字符的名称比较笨拙。此外，要想取一个有实际意义的名称，清楚地表达变量或常量的用途，25或30个字符应当足够了。

## 2.2 函数命名

1、函数名用大写字母开头的单词组合而成；

2、如果一个函数的参数名与保留关键字冲突，最好是为参数名添加一个后置下划线而不是使用缩写或错误的拼写（如class\_ 比 clss好。

## 2.3 类（Class）命名

1、名字应该能够标识事物的特性；

2、名字尽量不使用缩写，除非它是众所周知的；

3、名字可以有两个或三个单词组成，但通常不应多于三个；

4、在名字中，所有单词第一个字母大写。例如  IsSuperUser，包含ID的，ID全部大写，如CustomerID；

5、使用名词或名词短语命名类；

6、少用缩写；

7、不要使用下划线字符 (\_)。

例：public class FileStream

public class Button

public class String

## 2.4 接口（Interface）命名

和类命名规范相同，唯一区别是  接口在名字前加上“I”前缀

例：

interface IDBCommand;

interface IButton;

## 2.5 命名风格以及约束

以下的命名风格是众所周知的:

1、b (单个小写字母)

2、B (单个大写字母)

3、小写串，如:getname

4、带下划的小写串，如:\_getname,lower\_case\_with\_underscores

5、大写串，如:GETNAME

6、永远不要用字符"|",'O',"I'作变量名(某些字体种，字符不能与数字1,0分开，当要使用'I'时，用'L'代替它。

7、模块名：全小写名字，可以在模块种使用下划线来提高可读性。

# 第三部分 编码规则

## 3.1 大括号规则

将大括号放置在关键词下方的同列处，例如：

if ($condition)       while ($condition)

{                  {

      ...                   ...

}                  }

## 3.2 小括号规则

1、不要把小括号和关键词（if 、while等）紧贴在一起，要用空格隔开它们。

2、不要把小括号和函数名紧贴在一起。

3、除非必要，不要在Return返回语句中使用小括号。因为关键字不是函数，如果小括号紧贴着函数名和关键字，二者很容易被看成是一体的。

## 3.3 缩进规则

使用一个“Tab”为每层次缩进，缩进为4个空格。例如：

function func()

   {

      if (something bad)

      {

         if (another thing bad)

         {

            while (more input)

            {

            }

         }

      }

   }

## 3.4 If Then Else规则

如果你有用到else if 语句的话，通常最好有一个else块以用于处理未处理到的其他情况。可以的话放一个记录信息注释在else处，即使在else没有任何的动作。其格式为：

if (条件1)                 // 注释

   {

   }

   else if (条件2)            // 注释

   {

   }

   else                           // 注释

   {

   }

**注：**if 和循环的嵌套最多允许4层

## 3.5 比较规则

总是将恒量放在等号/不等号的左边。一个原因是假如你在等式中漏了一个等号，语法检查器会为你报错。第二个原因是你能立刻找到数值而不是在你的表达式的末端找到它。例如：

if ( 6 == $errorNum ) ...

## 3.6 Case规则

default case总应该存在，如果不允许到达，则应该保证：若到达了就会触发一个错误。Case的选择条件最好使用int或string类型。

## 3.7 对齐规则

变量的申明和初始化都应对齐。例如：

   int       m\_iCount;

   int       i,j;

   float     m\_fIncome,m\_fPay;

   m\_iCount  = 0;

   i         = 1;

   m\_fIncome = 0.3;

## 3.8 单语句规则

除非这些语句有很密切的联系，否则每行只写一个语句。

## 3.9 单一功能规则

原则上，一个程序单元（函数、例程、方法）只完成一项功能。

## 3.10 简单功能规则

原则上，一个程序单元的代码应该限制在一页内（25~30行）。

## 3.11 错误检查规则

1、编程中要考虑函数的各种执行情况，尽可能处理所有流程情况。

2、检查所有的系统调用的错误信息，除非要忽略错误。

3、将函数分两类：一类为与屏幕的显示无关， 另一类与屏幕的显示有关。对于与屏幕显示无关的函数，函数通过返回值来报告错误。对于与屏幕显示有关的函数，函数要负责向用户发出警告，并进行错误处理。

4、错误处理代码一般放在函数末尾。

5、对于通用的错误处理，可建立通用的错误处理函数，处理常见的通用的错误。

# 第四部分 编程准则

## 4.1 变量使用

1、不允许随意定义全局变量。

2、一个变量只能有一个用途；变量的用途必须和变量的名称保持一致。

3、所有变量都必须在类和函数最前面定义，并分类排列。

## 4.2 数据库操作

1、查找数据库表或视图时，只能取出确实需要的那些字段。

2、使用无关联子查询，而不要使用关联子查询。

3、清楚明白地使用列名，而不能使用列的序号。

4、用事务保证数据的完整性。

## 4.3 对象使用

尽可能晚地创建对象，并且尽可能早地释放它。

## 4.4 模块设计原则

1、不允许随意定义公用的函数和类。

2、函数功能单一，不允许一个函数实现两个及两个以上的功能。

3、不能在函数内部使用全局变量，如要使用全局变量，应转化为局部变量。

4、函数与函数之间只允许存在包含关系，而不允许存在交叉关系。即两者之间只存在单方向的调用与被调用，不存在双向的调用与被调用。

## 4.5 结构化要求

1、禁止出现两条等价的支路。

例如：if (a == 2)

       else if (a== 3)

             //

       else if (a == 2)

             //

       else

           //

2、避免使用GOTO语句

3、用 IF 语句来强调只执行两组语句中的一组。禁止 ELSE GOTO 和 ELSE RETURN。

4、用 CASE 实现多路分支

5、避免从循环引出多个出口。

6、函数只有一个出口。

7、不使用条件赋值语句。

8、避免不必要的分支。

9、不要轻易用条件分支去替换逻辑表达式

## 4.6 函数返回值原则

函数返回值：避免使用结构体等复杂类型，使用bool类型：该函数只需要获得成功或者失败的返回信息时候；使用int 类型：错误代码用负数表示，成功返回0

# 第五部分 数据库命名规范

使用本系统遵循以下命名规范：

1、表命名：用一个或三个以下英文单词组成，单词首字母大写，如：DepartmentUsers；

2、表主键名称为：表名+ID，如Document表的主键名为：DocumentID

3、存储过程命名：表名+方法,如：p\_my\_NewsAdd,p\_my\_NewsUpdate;

4、视图命名：View\_表名，如：ViewNews;

5、Status为表中状态的列名，默认值为0，在表中删除操作将会改变Status的值而不真实删除该记录；

6、Checkintime为记录添加时间列，默认值为系统时间；

7、表、存储过程、视图等对象的所有都为dbo，不要使用数据库用户名，这样会影响数据库用户的更改。

# 第六部分 其他准则

## 6.1 每行最多字符数

1、单行字符数限制不超过120个，超出需要换行，换行时遵循如下原则：

2、第二行相对第一行缩进 4个空格，从第三行开始，不再继续缩进；

3、运算符与下文一起换行；

4、方法调用的点符号与下文一起换行；

5、在多个参数超长，逗号后进行换行；

6、在括号前不换行。

## 6.2 操作符前后空格

1、赋值运算符、逻辑运算符、加减乘除符号、三目运行符等二元操作符的左右必须加一个空格；

2、一元操作符前后不加空格；

3、中括号、点等这类操作符前后不加空格。

## 6.3 操作符前后空格

1、注释与其上面的代码用空行隔开；

2、在每个类声明后、每个函数定义结束之后都要加空行；

3、在一个函数体内，逻辑上密切相关的语句之间不加空行，其他地方应加空行分隔。