

VB 查漏补缺知识点目录

一、format 格式化输出.....	2
二、命令按钮、单选按钮、复选按钮相同异同点，单击事件的调用方法。.....	6
三、各种数据类型的定义位置、关键字和作用域.....	11
四、变量的作用域与生存期问题.....	14
五、形参类型与窗体模块中的 public 类型分析.....	15
六、参数传递，参数传递出现冲突时的改错题，最好改形参的传递方式。.....	18
七、函数的调用方法、过程的调用的调用方法。.....	26
八、inputbox 函数、msgbox 函数与命令、shell 函数与命令的使用。.....	34
九、文件操作中#号的省略用法.....	39
十、VB 的 Tabindex、 tabstop 属性分析.....	39
十一、VB 的 default 属性和 cancel 属性分析。.....	43
十二、赋值语句的计算和转化功能.....	44
十三、len 函数遇到非字符串变量.....	47
十四、对象的 move 方法，及实现相同功能的方法。.....	48
十五、窗体的 moveble 属性.....	51
十六、数组使用的注意事项.....	52
十七、数组的 lbound()和 Ubound()函数的使用场合。.....	55
十八、VB 过程中的 shift 和 button 参数详解.....	60
十九、图片框与图像框知识.....	61
二十、VB 窗体的 Borderstyle 和 controlBox、minbutton、maxbutton 属性分析.....	63
二十一、通用转化函数的特点.....	66
二十二、picture 属性.....	67
二十三、各种属性的读取与设置.....	68
二十四、组合框和列表框的 text 属性分析、listindex 属性分析。.....	69
二十五、向列表框或组合框添加记录的方法.....	72

VB 网络查漏补缺知识点

一、 format 格式化输出

Print Format(332.678,"####.##") 这是格式化输出，其中 format 是一个函数，其返回值是一个字符串，当然可以把函数的返回值赋给一个字符串变量，但 VB 中一般习惯配合 print 语句，实现格式化输出。

该函数有两个参数，第一个参数是数（可以是数值型的，也可以是日期型的），第二个参数是格式串，用双引号引起来的格式串。

（一）第一个参数是数值型的。

如果第一个参数是数值，一般格式串中字符有：“.”（小数点）、“,”（千位分隔符）、“#”（在实际输出时，这位有就输出，没有就不输出，在小数点前后功能一样）、“0”（在实际输出时，这位没有也要补 0 输出）、%（转化为百分数输出，相当于乘 100，后面带%）、e 或 d（转化为科学记数法输出）。

举例说明：Print Format(332.678,"####.##") 输出 332.68（格式串中的整数部分是四位，因用的是#号，原数中有就输出，没有就不输出，原数只有 3 位，故整数部分只输出 3 位，小数部分用了两个#号，有就输出 2 位，没有就不输出，原数中有三位，按要求输出两位，当然需要四舍五入取整）

Print Format(32.678,"0000.##") 输出 0032.68（小数点前面有四个 0，原数的整数部分有没有四位都要补充四位，用 0 补）

Print Format(32.678,"0###.##") 输出 0022.68（格式串中的百位是#，而千位 0，输出时千位都补 0 了，百位能不补吗？当然要补）

Print Format(332.6,"####.##") 输出 332.6（格式串中，小数部分为两个#，原数中只有一位小数，故结果只需要输出一位小数了）

Print Format(332.60,"####.##") 输出 332.6，因为写这条语句时，小数点后面的 0 自动丢失，自动变成了 Print Format(332.6,"####.##")，要想输出 332.60，需要写成 Print Format(332.6,"####.00")。

Print Format(332.6,"####.##0") 输出 332.600，因为小数点后面的千分为是 0，所以需输出千分位小数（补 0），既然有千分位，那么百分位也要有（补 0）。

Print Format(332,"####.##") 输出结果为：332.（原数中没有小数，格式串中小数部分用的是#，故不输出小数，但有小数点，要注意）

Print Format(34332.678, "#,###.##") 输出结果: 34,332.68 (格式串中有千位分隔符,故结果加入了千位分隔符,在这里格式串的整数部分只有四个#,而原数的整数部分是五位的,结果必须5位,否则就不是原数了)

Print Format(34332.678, "###,###.##") 输出结果: 34,332.68 (格式串中的千位分隔符,原则上写在千位位置,写错位置也无所谓,写一个就够了,也多了也无所谓)

Print Format(332.678, "#,###.##") 输出结果: 332.68 (格式串中整数部分四个#,而原数中只有3位整数,故最终只输出3位整数即可,既然只有三位,千位分隔符也不能写了。

Print Format(332.6, "#,###.00%") 这里首先把332.6变成百分数即33260%,格式化输出时整数部分有四个#,而原数有5位整数,当然需要输出5位整数(否则与原数不相等了),需要加千位分隔符,需要两位小数部,最后加百分号,即结果是: 33,260.00%

Print Format(332.6, "0,000.###%") 这里首先把332.6变成百分数即33260%,格式化输出时整数部分有四个0,而原数有5位整数,当然需要输出5位整数(否则与原数不相等了),需要加千位分隔符,小数部分,多于两位保留两位,否则有几位留几位,最后加百分号,即结果是: 33,260.% (这里要带小数点)。

Print Format(332.6, "#,###.00%%"),后面可以有多个%,一个%相当于乘100,两个相当于乘10000,依次类推,后面加多个%,结果为3,326,000.00%%。

Print Format(23456, "##.00E+00")输出23.46E+03,这里,小数点前面的两个#号,表示转化为科学记数法后,整数部分两位,小数部分输出两位(这里用的是0,必须两位),前面说的是E前面的尾数部分,E后面有两个0,即指数部分用两位表示,这里的+号表示指数部分必须加符号,如果用-号,表示指数部分正数不加符号,只有负数才加符号。

下面多举几个例子:

Print Format(23456, "##.00E+00") 23.46E+03

Print Format(0.23456, "##.00E+00") 23.46E-02

加号表示指数部分必须加符号。

Print Format(23456, "##.00E-00") 23.46E03

Print Format(0.23456, "##.00E-00") 23.46E-02

减号表示指数部分正数不加符号,指数部分负数加符号。

正确的科学记数表示法E后面都要有符号

E+符号+后面的0,有两个0,则指数部分至少用两位数表示(如指数是1,则用01表示,当然指数是123,那么必须用三位表示)。有一个0或者后面是#,不管是几个#号,都是指数

部分不超过 10 就用 1 位，当然超过了 100，也得用 3 位。

```
如：Print Format(23456, "##.00E-0")    23.46E3
```

```
Print Format(23456, "##.00E-#")    23.46E3
```

```
Print Format(23456, "##.00E-##")    23.46E3
```

```
Print Format(23456, "##.00E-###")    23.46E3
```

还有科学记数的整数部分，想要几位，就有几位，所以小数点前面的#和 0 效果是一样的。

如：

```
Print Format(2, "00.00E-00")    20.00E-01
```

```
Print Format(2, "##.00E-00")    20.00E-01
```

最后格式化输出数值型数据格式符也就是这些。另外记住科学记数法只能 E 或 e（格式串里用大写则输出大写，格式串里用小写则输出小写），而不能用 d,除此之外的字符，一般都是原样输出，就连整个数前面的符号，不管是+而是-号，都是原样输出。没意义，考试也不会考的。

总结：格式串中除#和 0 外，还有“.”小数点和“,”千位分隔符，输出结果遵循以下原则：
0 表示此处必须有；#表示此处可以没有；输出结果要和原数相等；要遵守四舍五入原则；格式串中有“.”，输出结果一定带小数点，格式串中没有“.”，输出结果一定没有小数点；格式串有没有千位分隔符，结果一定没有，格式串中有千位分隔符，输出结果也不一定有。

+号和-号只是在科学记录法表示时用，写在 E 的后面，必须有，-号表示指数部分为正数省略，为负数才写，+表示指数部分无论为正为负都要写符号

(二) 第一个参数是日期型的。

```
Format(#1/13/2002#, "m/d/yy") 结果为：1-13-02
```

日期型格式化输出比较简单，主要指输出是按年月日输出，还是按月日年输出；是按 4 位年份，还是 2 位年份；如果月份是 mm，则月份一定要用两位数表示，如果月份是 m，则月份能用一位表示就用一位表示，不能用 1 位表示，才用两位表示。同样日期可用 d 或 dd 表示，道理和月份一样。

日期数据最后输出的分隔符一般都是“-”号，也就是说格式串中用“/”或“-”分隔，输出时都是用“-”分隔。

```
print Format(#1/13/2002#, "mm/d/yyyy") 输出结果为：01-13-2002
```

```
print Format(#1/13/2002#, "mm-d-yyyy") 输出结果为：01-13-2002
```

练习题

1、在 Visual basic 中，下列语句的输出结果是 ()

Print format\$(5689.36, " 000,000.000")

- A). 5,689.36 B). 5,689.360
C). 5,689.3 D). **005,689.360**

标准答案:d

2. 在 Visual Basic 中，命令 “ Print Format(332.678, "####.##") ” 也是一个控制数值输出格式的命令，它的输出结果是 ()。

- A). 332.678 B). 332.68 C). 332.67 D). 333

标准答案:b

3. 执行语句 “Print Format(5459.478, " ##,##0.00”)” 的正确输出结果是 ()

- A). 5459.48 B). 5,459.48 C). 5459.478 D). 5459.47

标准答案:b

4. 执行语句 “Print Format(5459.478, " 00,000.00”)” 的正确输出结果是 ()

- A). 5459.48 B). 05,459.48 C). 5459.478 D). 5459.47

标准答案:b

5. 设 a 为 Integer 型变量，s 为 single 变量，把 3.75 赋值给 s 后，再执行下列某一语句，则在文本框中显示的结果与执行语句 a=s : Text1=a 的结果相同的是 ()

- A). A= Int(s): Text1=a
B). a= Fix(s) : Text1=a
C). **a=CInt(s): Text1=a**
D). a=Format(s, "####.#") : Text1=a

标准答案:c 、 d

解析: CInt 有四舍五入功能，为 4。Format(s, "####.#") 结果为字符串 3.8，将“3.8”赋给整型变量 a，先转化为数值 3.8，再赋给 a，会自动四舍五入变成 4

6. 以下的 4 个表达式中，有 () 个能够对 Single 类型变量 x(正数) 的第 3 位小数正确地进行四舍五入计算。

- val(Format(x, "#.00")
int(100*(x+0.005))/100
CInt(100*(x+0.005))/100
Fix(100*(x+0.005))/100

- A). 1个 B). 2个 C). 3个 D). 4个

标准答案:c

解析: 1、val(Format(x, "#.00")), 很明显, 输出两位小数, 整数部分原样输出, format 函数的结果是字符串, 用 VAL 转化为数值。

2、CInt 函数有四舍五入功能, 所以不正确。

3、int 和 fix 对于正数而言, 没有区别。

7. Format(1732.46, "+##, ##0.0%") 的返回值为()

- A). "173.246.0" B). "1,732.5%"
C). "+173,246.0%" D). "+173,246%"

标准答案:c

解析: 含有百分号, 且转化百分数后只保留一位小数, 整数部分有多少位输出多少位, 带千位分隔符, 且输出的值一定要和原数值相等的, 这里只有 "+173,246.0%" 符合题意。

这里的 "+" 跟数的正负没关系, 只是原样输出。

8. 在 Visual basic 中, 以下程序段的输出结果是()。

Print Format(#2/14/1998#, "m/d/yy")

- A). 2-14-98 B). 02-14-98 C). 2/14/98 D). 02/14/98

标准答案:a

说明: 在 XP 系统结果为 A, 在 WIN10 系统为 C, 同学试一下是不是, 还有在 WIN7 系统结果为哪一个(a)。

二、 命令按钮、单选按钮、复选按钮相同与异同点、单击事件的调用方法。

(一)相同点:

- 1、都有 picture 和 style 属性, 都需要先设置 style 为图形模式 (1-Graphical), 再设置 picture 属性, 才能显示图片。
- 2、都有 Value 属性, 且 Value 属性为这三个控件的默认属性。

(二)不同点

- 1、命令按钮和单选按钮的 Value 属性为逻辑值, 而复选按钮的 Value 属性为数值 (0、1、2)
- 2、命令按钮和复选按钮都只有单击事件, 没有双击事件, 而单选按钮单击、双击事件都有。命令按钮和复选按钮将双击事件解释为两次单击事件。
- 3、default 和 cancel 为命令按钮的特有属性, 而单选按钮和复选按钮没此属性。

（三）单击事件的调用方法

1、命令按钮：

（1）任何时候单击该命令按钮，都能调用其单击事件。

（2）任何时候设置其 Value 值为真，都能调用其单击事件，然后其 Value 值自动变成假。

2、复选按钮：

（1）任何时候单击该复选按钮，都能调用其单击事件，因为只要单击其 Value 值就会变化。

（2）在代码中改变其 Value 值就会触发其单击事件。其 Value 共有三个值，只要变了，就会触发单击事件。

总结：无论采用什么方法，只要改变了其 Value 值就会触发复选按钮的单击事件。

3、单选按钮：

（1）当单选按钮没被选中时候，单击该单选按钮，会触发其单击事件。而单选按钮已被选中，再单击其不会触发其单击事件。

（2）在代码中将单选按钮的 Value 值为由假修改为真时，会触发其单击事件，而由真修改为假时不会触发，其它修改也不会触发。

总结：对于单选按钮，无论采用什么方法，只有其 value 属改由假变为真时，才触发其单击事件。而单击 option 按钮不一定能使其 value 属性由假变真，故单击也不一定能执行其单击事件。

（四）其它：

单击一个没有焦点的命令按钮，即触发单击事件，又触发其 gotfocus 事件；单击一个已有焦点的命令按钮，只触发其单击事件，不触发其 gotfocus 事件，因为 gotfocus 事件是获得焦点的那一刻触发，焦点也存在于该命令按钮，那一刻早已过去了。只有失去了，再获得的那一刻再触发。

单击了没有被选中的单选按钮后，一定会触发其单击事件，同时由于其获得了焦点，所以也触发其 gotfocus 事件。注意：如果该单选按钮已被选中，再单击该单选按钮，即不触发单击事件，也不触发 gotfocus 事件。

单击没有被选中的复选按钮，即触发单击事件，又触发其 gotfocus 事件；单击已选中的复选按钮，只触发其单击事件，不触发其 gotfocus 事件。

解析：gotfocus 事件的触发条件是获得焦点，即原来没焦点，现在有焦点触发；而单击事件的触发条件，前面已分析。仅此而已。另外这两个事件的触发顺序是先触发 gotfocus 事件，再触发单击事件。

练习题:

1. 单选按钮 (OptionButton) 用于一组互斥的选项中, 若一个应用程序包含多组互斥条件, 在不同的 () 中安排适当的单选按钮, 即可实现

- A). 框架控件 (Frame) 或图像控件 (Image)
- B). 组合框 (ComboBox) 或图像控件 (Image)
- C). 组合框 (ComboBox) 或图片框控件 (PictureBox)
- D). 框架控件 (Frame) 或图片框控件 (PictureBox)

标准答案:d

解析:

2. 复选框不支持鼠标的双击事件, 如果双击则系统会解释为两次单击事件。()

- A). 正确
- B). 错误
- C).
- D).

标准答案:a

解析: 命令按钮和复选按钮都没有双击事件, 双击则系统会解释为两次单击事件。而单选按钮即有单击事件, 又有双击事件。

3. 在 Visual basic 中, 当单击了没有被选中的单选按钮控件后, 下列正确的是 ()。 (正确率 84%)

- A). 只执行 click 事件
- B). 既执行事件 click, 也执行事件 GotFocus
- C). 只执行 GotFocus
- D). 具体执行哪个事件要在程序或属性中设定

标准答案:b

解析:

单击了没有被选中的单选按钮后, 一定会触发其单击事件, 同时由于其获得了焦点, 所以也触发其 gotfocus 事件。

注意: 如果该单选按钮已被选中, 再单击该单选按钮, 即不触发单击事件, 也不触发 gotfocus 事件。

另外: 单击一个没有焦点的命令按钮, 即触发单击事件, 又触发其 gotfocus 事件; 单击一个已有焦点的命令按钮, 只触发其单击事件, 不触发其 gotfocus 事件, 因为 gotfocus 事件是获得焦点的那一刻触发, 焦点也存在于该命令按钮, 那一刻早已过去了。只有失去了, 再获得的那一刻再触发。

4. 设已经在某个选项按钮 Option1 的 Click 事件过程中编写了一些代码, 运行时, 以下说法正确的是 []。

- (1)任何时候用鼠标单击选项按钮都会执行其 Click 事件过程
 - (2)任何时候执行语句 Option.Value=True 都会执行选项按钮的 Click 事件过程
 - (3)当选项按钮处于未选择状态时，用鼠标单击选项按钮都会执行 Click 事件过程
 - (4)当选项按钮处于未选择状态时，执行语句 Option.Value=True 将执行选项按钮的 Click 事件过程
- A). (2) (3) B). (1) (4) C). (3) (4) D). (1) (2)

标准答案:c

解析:按钮、单选按钮、复选按钮有一些相似的属性和特点

一、这三种按钮的 picture 属性的设置都需要配合 style 属性完成，也就是说要设置 picture 属性，使用按钮上显示图片，都需要将其 style 属性设置为图形模式。

二、执行这三种按钮的单击事件，除了单击按钮本身处，还可以通过设置其 Value 属性来实现。

1、标准按钮 command 任何时候单击或设置其 value 属性为真都会执行其单击事件。

2、复选按钮 Check 任何时候单击或改变其 value 属性的时候都会执行其单击事件。对于复选按钮，其 value 属性发生改变时，就执行其单击事件，且任何时候单击都会改变其 value 属性。

3、对于单选按钮，只有其 value 属性由假变为真时，才执行其单击事件，而单击 option 按钮不一定能使其 value 属性由假变真，故单击也不一定执行其单击事件。且记对于单选按钮，只有其 value 属性由假变真时，才执行其单击事件。

5. 设置一个单选按钮 (OptionButton) 所代表选项的选中状态，应当在属性窗口中改变的属性是

- A). Name B). Caption C). Text D). Value

标准答案:d

解析:

6. 下面[]控件不支持 DblClick 事件。

- A). Image B). Form
C). OptionButton D). CheckBox

标准答案:d

解析:单选按钮 (OptionButton) 可以响应双击事件

复选按钮 (CheckBox) 和标准按钮 (CommandBox) 不能响应双击事件

7. 要设置命令按钮的背景图形, 必须设置的两个属性是 ()

- A). Value 和 Icon
- B). Style 和 Icon
- C). Style 和 Picture
- D). Caption 和 Picture

标准答案:c

解析:如果要在命令按钮上显示背景图形, 只设置按钮的 Picture 是没有效果的, 还必须设置按钮的 Style 属性为图形模式。

不只是命令按钮有此特点, 按钮类 (单选按钮和复选按钮) 都是这样的。

8. 以下关于选项的说法, 正确的是 []。

- A). 在一个容器中的单选按钮可以同时有多个被选中 2 个
- B). 在一个容器中的单选按钮可以同时有多个被选中
- C). 一个窗体上 (包括其它容器中) 的所有单选按钮一次只有一个被选中
- D). 一个窗体上 (不包括其它容器中) 的所有单选按钮一次只有一个被选中

标准答案:d

解析:

9. () 对象可以响应 dblclick() 事件

- A). 复选框
- B). 计时器
- C). 窗体
- D). 命令按钮

标准答案:c

解析: 计时器只响应 timer 事件

命令按钮和复选按钮没有双击事件, 如果双击, 则解释为两次单击事件。

单选按钮有双击事件。

12. 若要设置命令按钮为图形风格的按钮, 可通过 () 属性来设置。

- A). Style
- B). BorderStyle
- C). BackStyle
- D). Appearance

标准答案:a

解析: 要想设置命令按钮为图形风格, 需要将命令按钮的 style 属性设置为 1。另外设置成图形风格后, 可以在命令按钮上显示图片。

13. 下列控件中没有 Picture 属性的是 ()

- A). 窗体
- B). 标签
- C). 命令按钮
- D). 以上都不是

标准答案:b

解析: 标签没有 Picture 属性

14. 下列叙述中正确的是()

- A). 命令按钮的默认属性为 Caption B). 标签的默认属性为 Caption
C). 复选框的默认属性为 Caption D). 文本框的默认属性为 Caption

标准答案:b

解析:命令按钮、复选框、单选框的默认属性都是 Value, 文本框的默认属性为 text.

15. 下列选项中, ()不可以实现窗体上某个命令按钮的单击事件。

- A). 命令按钮的 value 属性为 True
B). 用鼠标直接单击某个命令按钮
C). 当命令按钮获得焦点后, 按回车键
D). 当命令按钮的 Enabled 属性值为 True 时, 任何情况下

标准答案:d

解析: 要理解题目表达的意思, 即什么时候能运行按钮的单击事件。

命令按钮的 value 属性是其默认属性, 设置其值为真, 即可调用一次该按钮的单击事件, 调用完成后, 其值就变成假了。

16. 在 Visual Basic 中, 若要使命令按钮获得焦点, 可使()用方法来实现。

- A). Refrseh B). SetFocus C). GotFocus D). Value

标准答案:b (正确率 89%)

解析:这里说的方法, 是对象的三要素(属性、事件、方法)这一的方法。

选项中只有两个方法,其中 SetFocus 是使用命令按钮获得焦点的方法。

三、各种数据类型的定义位置、关键字和作用域

1、过程级变量:

只能在过程中定义, 其作用域为定义该变量的过程。定义过程的关键字只有两个, 一个是 dim,用 dim 定义的过程级变量为动态过程级变量, 出了过程就释放, 再次进入过程要重新建立; 另一个是 static,用 static 定义的过程级变量为静态过程级变量, 出了过程虽然不释放, 但看不到了, 只有在再进入该过程才能再次看到, 原来的值能保留, 继续使用, 当定义该变量的过程所在的窗体卸载时, 该静态过程级变量释放。

2、模块级变量(包括窗体级变量和标准模块级变量)

模块级变量在模块的通用声明字段定义, 用 private 或 dim 定义, 窗体级变量在窗体的通用声明字段用这两个关键字定义, 而标准模块级变量在标准模块的通用声明字段用这两个关

键字定义。

模块级变量的作用域是定义该变量的整个模块，该模块卸载该变量释放。

3、全局变量：在窗体或标准模块的通用声明字段定义，用 **public** 关键字定义，其作用域是整个程序，只有整个程序结束，变量才释放。

常数、固定长度字符串、数组、用户自定义类型及 **declare** 语句不允许作为窗体模块的 **public** 成员，也就是说在窗体的通用声明字段不能定义以上全局变量，但可以在标准模块的通用声明字段定义以上全局变量。

全局变量的引用，窗体模块定义的全局变量，只有在本窗体引用时不用写窗体名，引用其它窗体的全局变量，不管有没有重名变量，都必须加上窗体名。标准模块定义的全局变量，不管在哪里引用，只要是没有重名的变量，都可省略标准模块的名子。

4、关于变量的重名问题

(1) 过程级变量不用担心和其它过程中的变量重名，因其作用域不同。

(2) 过程级变量和模块级变量重名。在过程中，只要是定义了和模块级变量同名的变量，那么在该过程中，就只能看到自己定义的过程级变量，本模块的同名变量看不到了。

(3) 各模块中全局变量重名，使用本模块的变量，可以直接引用（当然也可以带上模块名），使用其它模块的全局变量，只要带上定义该变量的模块名字就可以。当然如果本模块没有和标准模块定义的全局变量重名，引用标准模块的全局变量，也可以省略标准模块的名字。

（比如 form1 没定义全局变量 X,而 form2 和标准模块 Module1 都定义了全局变量 X,则在 form1 中直接引用 x, 指的是标准模块 Module1 中的 X,当然如果还有一个标准模块 Module2 也定义了全局变量 X, 那就不行了，必须写上模块的名字了）

考试也只能考这么多。下面可以了解

(4) 只有全局变量的前面可以带上模块名字，模块级变量前面带上模块名字会提示语法错误。

(5) 过程级变量和本模块的全局变量重名，在过程中，直接写变量名，就是自己定义的过程级变量，带上本模块的名字，可以引用本模块的同名全局变量。

(6) 模块级变量和本模块的全局变量重名，这是不允许的。

(7) 过程级变量和其它模块的全局变量重名，在过程中，直接写变量名，就是自己定义的过程级变量，也可以引用其它模块的同名全局变量，只要带上定义该同名变量的模块名字就可以了。

(8) 模块级变量和其它模块的全局变量重名，在本模块中，直接写变量名，就是引用的

自己定义的模块级变量，也可以引用其它模块的同名全局变量，只要带上定义该同名变量的模块名字就可以了。

总结：

- (1、过程级变量重名互不影响。
- (2、过程级变量和模块级变量重名，在过程中无法看到模块级变量。
- (3、过程级变量和本模块的全局变量重名，在过程中可以带上模块名字来引全局变量。
- (4、本模块的全局变量不能和模块级变量重名。
- (5、只有全局变量的前面才能带模块名字。
- (6、本模块的变量和其它模块的全局变量重名，只要带上模块的名字，就能引用其它模块的全局变量。

(7、引用标准模块的全局变量时，只要不会引起异意，就可以省略标准模块的名字，但无论何时都不能省略窗体模块的名字。

练习题：

1. Visual Basic 中假定一个工程文件由一个窗体 Form1 和两个标准模块 Model1 及 Model2 组成。

Model1 代码如下：

```
Public x as integer
public y as integer
Sub s1()
    x=1
    s2
End Sub
Sub s2()
    y=10
    form1.show
End Sub
```

Model2 代码如下：

```
Sub Main()
    s1
End Sub
```

其中 Sub Main() 被设为启动对象，程序运行后，各模块的执行顺序是()。

- A). Model2 Model1 Form1
- B). Model1 Model2 Form1
- C). Form1 Model1 Model2
- D). Model2 Form1 Model1

标准答案:a

解析：程序从 SUB main() 过程启动，而该过程存在于 Model2 模块中，故首先执行的模块是 Model2，SUB main() 过程中调用 Model1 模块中的过程 s1，所以第二个被执行模块是标准模块 Model1，在 Model1 模块中的过程 S2 中显示 Form1 窗体，所以 Form1 是最后被执行的模块。

2. 如果在程序中要将 c 定义为静态变量，且为整型数，则应使用的语句是

- A). Redim a As Integer
- B). Dim a As Integer
- C). Public a As Integer
- D). Static a As Integer

标准答案:d

解析：

3. 使用 Public Const 语句声明一个全局的符号常量时，该语句应放在()。

- A). 窗体模块或标准模块的通用声明段
- B). 窗体模块的通用声明段
- C). 标准模块的通用声明段
- D). 过程中

标准答案:c

解析:在窗体的通用声明字段，不能使用 **Public Const** 语句声明一个全局的符号常量时。

四、变量的作用域与生存期问题

变量分为三种类型的变量

1、动态过程级变量：其作用域为定义该变量的过程，出了过程，变量即释放。生存期为该过程的生存期，过程结束，则生存期结束。

静态过程级变量：其作用域为定义该变量的过程，第一次进入过程，定义该变量，出了过程，变量不释放，但其它任何过程都看不到，也不能使用，再次进入该过程，还能看到原来的值（不用再次定义），继续使用。生存期为该过程所在的窗体的生存期，窗体卸载，该变量的生存期结束。

2、模块级变量：其作用域是定义该变量的模块（窗体或标准模块）。生存期也为所在的模块的生存期。窗体卸载，窗体内定义的窗体级变量生存期结束。

3、全局变量：其作用域是整个程序，生存期也是整个程序的生存期，程序结束，生存期结束。

五、形参类型与窗体模块中的 public 类型分析

1、常数、固定长度字符串、数组、用户自定义类型及 **declare** 语句不允许作为窗体模块的 **public** 成员，也就是说在窗体的通用声明字段不能定义以上全局变量，但可以在标准模块的通用声明字段定义以上全局变量。

2、定长字符串不能做形参，但定长字符串数组可以做形参，当然需要实参也必须是定长字符串数组，长度可以不同。

3、数组做形参，只能是动态组数，不能是常规数组，且只能传地址，即不能用 **byval** 修饰 (**byref** 常省略)，而实参用数组名，可以加空括号，也可以不可空括号。

练习题：

1. 在窗体模块的通用声明处用下面的语句声明变量、数组，正确的语句有 () 个

① `Public A(5) As Integer`

② `Public N As integer`

③ `Public St As Strng*10`

④ `Private b() As Integer`

A). 2 B). 3 C). 4 D). 1

标准答案:a

解析：在窗体的通用声明字段不能定义 **public** 类型的常量、定义字符串、数组和用户自定义类型。但可以定义 **private** 类型的。

2. 下面有关数组处理的叙述中，错误的是 ()

①在过程中用 `ReDim` 语句定义的动态数组，其下标的上下界可以是赋值的变量

②在过程中，可以使用 `Dim`、`Private` 和 `Static` 语句定义数组

③用 `ReDim` 语句重新定义动态数组时，不得改变该数组的数据类型

④可用 `Public` 语句在窗体模块的通用说明处定义一个全局数组

A). ①②③④ B). ①③④ C). ①②③ D). ②④

标准答案:d

解析：`private` 只能模块的通用声明字段使用，不能在过程中使用。所以②错误。

在窗体的通过声明字段不能定义 **public** 类型的数组、定义字符串、常量和用户自定义类型。故④错。

3. 在窗体的通用说明部分有以下说明语句，其中正确的是 ()

A). `Private Const A As Integer = 235/2`

- B). Const B As Integer= CInt(2. 53)
- C). Dim Const C As Integer=3579
- D). Public Const S As Boolean=True

标准答案:a

解析: 学会分析很重要。

- 1、“Public Const S As Boolean=True” 错（因为窗体模块不能定义全局常量）
- 2、“Dim Const C As Integer=3579” 没看见过这么写的，const 前面可以是 private 或 public，前者是模块级的常量，后者是全局常量（标准模块可以定义），但不能用 dim。（错）
- 3、给常量赋值前可以是表达式，但不能包含函数。
- 4. 可以在()中使用语句 Public PubStr As String*20 定义一个定长字符串。

- A). 窗体模块 B). 标准模块 C). 类模块 D). A、B、C 三者均可

标准答案:b

解析:在窗体的通用声明字段不可以定义：常数、固定长度字符串、数组及用户自定义类型的 public 成员，这些类型的 public 成员只能在标准模块中定义。

5. 在窗体模块的通用声明处，可以使用()语句说明数组

①Public A(10)As Integer

②Dim A(10) As Integer

③Private A(10) As Integer

④Static A(10)As Integer

- A). ①② B). ②③ C). ③④ D). ①④

标准答案:b

解析: 1、在窗体的通用声明字段不可以定义：常数、固定长度字符串、数组及用户自定义类型的 public 成员，故①是错误的。

2、Static 关键字只能在过程中使用，故④是错误的。

6. 下列有关数组的叙述中，不正确的是 ()

- ①在过程中用 REDIM 语句定义的动态数组，其下标的上下界可以是变量
- ②数组作为形式参数时，传递的是每个数组元素的值
- ③在窗体模块的通用声明处可以用 PUBLIC 说明一个全局数组
- ④数组定义语句中可以用负数或小数来指定某一维的维下界或维上界的值

- A). ③④ B). ①② C). ②③ D). ②④

标准答案:c

解析:常数、固定长度字符串、数组、用户自定义类型及 Declare 语句不允许作为对象模块的 Public 成员

①在过程中用 REDIM 语句定义的动态数组, 其下标的上下界可以是变量。绝对正确, 用 DIM 定义常规数组, 这样是不可以的。

②数组作为形式参数时, 传递的是每个数组元素的值。这句话毛病太多了。数组名做实参只能传地址。

③在窗体模块的通用声明处可以用 PUBLIC 说明一个全局数组。是不可的。“常数、固定长度的字符串、数组、用户自定义类型以及 Declare 语句不允许作为窗体对象的 public 成员”, 但是在标准模块中可以。

④数组定义语句中可以用负数或小数来指定某一维的维下界或维上界的值。都可以, 可以自动四舍五入调整为整数。

7. 下面有关 Visual Basic 数组处理的叙述中, 正确的是()

- A). 在过程中使用 ReDim 语句可以改变动态数组数据的类型
- B). 用 ReDim 语句重新定义动态数组时, 可以改变数组的大小, 但不能改变数组的维数
- C). 在过程中, 可以使用 Dim、Private 和 Static 语句来定义数组
- D). 不可以用 Public 语句在窗体模块的通用声明段说明一个全局数组

标准答案:d

解析:一、redim 可以改变数组的大小、维数, 上下限, 唯独不能改变数组的类型。

二、在过程中, 可以用 DIM、Static 定义变量, 但不能用 Private 定义变量, Private 和 Public 只能在模块的通用声明字段中定义变量

三、常数、固定长度字符串、数组、用户自定义类型及 Declare 语句不允许作为窗体对象模块的 Public 成员。只能在标准模块的通用声明字段内定义。

8. 在 Visual Basic 中, 使用 Public Const 语句声明一个全局的符号常量, 该语句应该放在 ()

- A). 过程中 B). 窗体模块或标准模块的通用声明字段
- C). 标准模块的通用声明字段 D). 窗体模块的通用声明字段

标准答案:c

解析:常数、固定长度字符串、数组、用户自定义类型及 Declare 语句不允许作为窗体对象模

块的 Public 成员。只能在标准模块的通用声明字段内定义。

9. 在 Visual basic 中，以下声明语句中错误的是()

- A). Const var1= 123
- B). Dim var2
- C). Public vat As Integer=1234
- D). Dim intSum As Integer

标准答案:c

解析:Public vat As Integer=1234

这条语句一看应该是定义常量的，但定义常量应该用 `const`，故该语句错误，正确的应该写成：`Public Const vat As Integer=1234`。且记，全局类型的常量只能在标准模块的通用声明字段内定义。

10. 在 Visual Basic 中，在窗体模块的声明字段中声明变量时，不能使用的关键字是()

- A). Public
- B). Static
- C). Dim
- D). Private

标准答案:b

解析:static 用来定义静态过程级变量，过程级变量只能在过程中定义

11. 在过程中定义的变量，若希望在离开该过程后，还能保存过程中局部变量的值，则应使用（ ）关键字在过程中定义局部变量。

- A). dim
- B). Private
- C). Public
- D). Static

标准答案:d

解析:

六、参数的传递，参数传递时出现冲突时的改错题，最好改形参的传递方式。

(一) 参数的传递方式

传地址(传址)和传数值(传值)两种传递方式。

如果形参用 `BYREF` 修饰(或省略)，实参用变量名，则为传地址，其它所有情况，都是为传数值。

(二) 参数传递时的注意事件:

1、形参和实参数量上要一样多。(特殊情况，可以省略部分参数的除外。只有系统已定义好的标准函数或 `SUB` 过程，可以省略部分参数，我们自定义的函数或 `SUB` 过程，暂时还没学这么多，所以我们自己编的过程或函数，形实结合，参数数量都要一样多)

2、形实结合时，位置必须对应好。

3、形实结合时，如果是传地址，类型必须完全一样，如果是传数值，要求类型匹配，兼容即可。比如数值型的相互兼容，再比如一边是数，一边是由数组成的串，也可以兼容。

（三）填空题如果在过程或函数的定义处有一个空，让填形参，或者在调用处有一个空，让填实参，一定要考虑到参数的传递方式，是不是必须传数值，如果必须传数值的话，如果两个空都有，一般都是在形参处用 `byval` 修饰形参（这里记住有两种填法，可以形参处用 `byval` 修饰形参，或者在实参处给变量名加括号，但**最好**是用**第一种方法**），当然如果只有一个空，那么就不用选择用哪种方法了，但一定要考虑到是否需要必须传数值（或者必须传地址）。

做改错题时，如果有参数传递情况，也要考虑是否参数的传递方式有错误，该传地址传了数值，或该传数值而传了地址。必须考虑到这个问题，当然改错的时候，需要必须传数值了，最好将形参改成用 `BYVAL` 修饰，当然如果还有其它要求，或者实参处还有别的错误，那另当别论。

练习题：

1、Private Function gcd() As Integer

Dim r As Integer

r = m Mod n

Do While r <> 0

 m = n

 n = r

 r = m Mod n

Loop

()

End Function

Private Sub Command1_Click()

Dim n As Integer, p As Integer

n = Val(Text1)

For p = n - 1 To 2 Step -1

 If() Then List1.AddItem p

Next p

End Sub

2. 下面的过程定义语句中合法的是 ()

- A). Sub Proc1(n) As Integer B). Function Proc1(Proc1)
C). Function Proc1(ByVal n) D). Sub Proc1(ByVal n())

标准答案:c

解析:过程名(函数名)都不能与参数同名(形参和实参)

过程不能有返回值

数组做过程的形参, 只能传地址

3. 要想在过程调用后返回两个结果, 下面的过程定义语句合法的是 ()

- A). Sub Proc1(n, ByVal m)
B). Sub Proc1(n , m)
C). Sub Proc1(ByVal n , m)
D). Sub Proc1(ByVal n , ByVal m)

标准答案:b

解析:

4. 在 Visual basic 中, 下列对过程的定义正确的是()

- A). Sub F(x%, 3)
B). Sub FF(x%,y%)As Integer
C). Function M(ByVal x%)As Boolean
D). Function MM(x+y)

标准答案:c

解析:一、函数或过程的形参不能是常量、也不能是表过式, 只能是变量。

二、函数有返回值, 过程不能有返回值

5. 在参数传递过程中, 使用关键字 () 来修饰参数, 可以使之按值传递。

- A). ByVal B). Value C). ByRef D). Reference

标准答案:a

解析:

6. 下列程序的输出结果为 ()

```
Private Sub Command1_Click()  
    For i = 1 To 3  
        GetValue (i)
```

```

Next I
Print GetValue(i)
End Sub
Private Function GetValue(ByVal a As Integer)
    Static s As Integer
    s = s + a
    GetValue = s
End Function

```

A). 4 B). 5 C). 10 D). 11

标准答案:c

解析: function 函数可以当一般 sub 过程调用, 而一般的 sub 过程, 不能当 function 函数来调用。

function 函数用 call(或省略) 时, 和一般 sub 过程的使用方法是完全一样的。

7. 下面关于过程参数的说法, 错误的是()

- A). 过程的形参不可以是定长字符串类型的变量
- B). 形参是定长字符串型的数组, 则对应的实参必须是定长字符串型数组, 且长度相同**
- C). 若形参是按地址传递的参数, 形参和实参也可以按值传递方式进行形实结合
- D). 按值传递参数, 形参和实参的类型可以不同, 只要相容即可

标准答案:b

解析: 对于我们不太熟悉的问题, 要使用排除法, 学会分析。

1、过程的形参可以是定长字符串类型的数组, 但不能是定长字符串类型的变量, 所以 A 正确。

2、形参按地址传递的参数, 形实结合也可按数值传递。所以 C 正确。

3、形实结合按值传递, 无须类型完全相同, 相容即可, 所以 D 也正确。

再看选项 B, 我们不太清楚, 其它三项绝对正确, 那么 B 选项看着别扭, 那么就是 B 错误了。

说明: 形参是定长字符串类型的数组, 那么实参也必须是定长字符串数组, 但是长度可以不同。

8. 程序中的不同过程之间, 不能通过() 进行数据传递

- ①全局变量 ②窗体或模块级变量 ③形参与实参结合 ④静态变量

- A). ①②④ B). ①②③ C). ②④ D). ④

标准答案:d

解析: 关键要理解本题表达的是什么意思。

一个程序的不同过程之间可以通过全局变量进行数据传递, 既一个过程中的数据可以通过全局变量传递到另一个过程中去。当然也可能通过窗体或模块级变量传递; 形实结合也可以传递。

静态变量属于过程级变量, 只在定义的过程有效, 出了定义它的过程看不到, 所以不能在不同的过程之间传递数据。

9. 以下有关过程的说法中, 错误的是 ()

- A). 在 SUB 或 function 过程中不能再定义其它 SUB 或 function 过程
B). 在调用过程时, 与使用 Byref 说明的形参对应的实参只能按地址传递方式结合
C). 递归过程既可以是递归 function 过程, 也可以是递归 SUB 过程
D). 在调用过程时, 形参为数组的参数对应的实参只能是数组。

标准答案:b

解析: 1、在调用过程时, 形参为数组的参数对应的实参只能是数组。该选项绝对正确, 要记住。

2、在调用过程时, 与使用 Byref 说明的形参对应的实参若为变量名, 则传地址, 若为表达式, 则传数值。

10. Sub 过程与 Function 过程最根本的区别是 ()

- A). Sub 过程的过程名不能返回值, 而 Function 过程能通过过程名返回值
B). Sub 过程可以使用 Call 语句或直接使用过程名调用, 而 Function 过程不可以
C). Function 过程可以有参数, Sub 过程不可以
D). 两种过程参数的传递方式不同

标准答案:a

解析: Sub 过程不能用调用函数的方式调用, 但 Function 过程可以用调用一般过程的方式调用。

一般过程的方式有两种方法: 用 call 时, 参数加括号, 省略 call 时, 参数不能加括号。

11. 以下关于子过程或函数的定义中, 正确的是 ()

- A). Sub f1(n As String*1)
B). Sub f1(n As Integer)As Integer

C). Function f1(f1 As Integer)As Integer

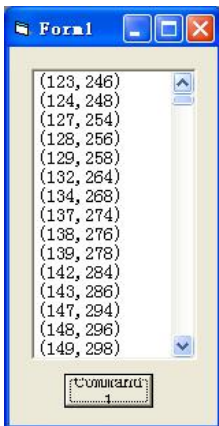
D). Function f1(ByVal n As Integer)

标准答案:d

解析:

- 1、定长字符串不能做形参。
- 2、过程没有返回值。
- 3、过程或函数名字不能和形参名字相同
- 4、function 函数一般有返回值，但也可以没有返回值。

12、本程序功能是查找给定范围内满足条件的整数对。每个整数的各位数字各不相同，且不得为数字 0，第二个数等于第一个数的两倍。例如：123 和 246 就是符合条件的数对。请修改程序的错误之处。(第十套)



```
Private Sub Command1_Click()
```

```
    Dim i As Integer
```

```
    Dim n As Integer
```

```
    For i = 123 To 5678
```

```
        n = i * 2
```

```
        If fun(i) And fun(n) Then
```

```
            List1.AddItem "(" & i & "," & n & ")"
```

```
        End If
```

```
    Next i
```

```
End Sub
```

```
Private Function fun( n As Integer) As Boolean 'Private Function fun(ByVal n As Integer) As Boolean '
```

```
Dim a() As Integer, i As Integer, j As Integer
```

```
Do
```

```
    i = i + 1
```

```
    ReDim Preserve a(i)    '改错可以去掉它
```

```
    a(i) = n Mod 10
```

```
    If a(i) = 0 Then Exit Function
```

```
    n = n \ 10
```

```
Loop Until n = 0
```

```
For i = 1 To UBound(a) - 1
```

```
    For j = i + 1 To UBound(a)
```

```
        If a(i) = a(j) Then Exit For    'If a(i) = a(j) Then Exit Function '
```

```
    Next j
```

```
Next i
```

```
fun = True
```

```
End Function
```

13. 在 Visual Basic 中, 名为 sort 的 Sub 子过程的形式参数为一数组, 以下的定义语句中正确的是()。

- A). Private Sub sort(a() As Integer)
- B). Private Sub sort(a(10)As Integer)
- C). Private Sub sort(ByVal a() As Integer
- D). Private Sub sort(a(,)As Integer)

标准答案:a

解析:

数组做形参, 只能是动态数组, 故 B 排除。

数组名形实结合, 只能传地址, 故 C 排除。

D 选项格式都不正确, 因为动态数组定义时只有数组名和空括号两部分组成。动态数组定义时根本不确定维数。

14. 在过程定义中用()表示形参的传址。

- A). ByRef
- B). ByVal
- C). Value
- D). Var

标准答案:a

解析:

15. 下列有关数组的叙述中, 不正确的是 ()

- ①在过程中用 REDIM 语句定义的动态数组, 其下标的上下界可以是变量
 - ②数组作为形式参数时, 传递的是每个数组元素的值
 - ③在窗体模块的通用声明处可以用 PUBLIC 说明一个全局数组
 - ④数组定义语句中可以用负数或小数来指定某一维的维下界或维上界的值
- A). ③④ B). ①② C). ②③ D). ②④

标准答案:c

解析:常数、固定长度字符串、数组、用户自定义类型及 Declare 语句不允许作为对象模块的 Public 成员

①在过程中用 REDIM 语句定义的动态数组, 其下标的上下界可以是变量。绝对正确, 用 DIM 定义常规数组, 这样是不可以的。

②数组作为形式参数时, 传递的是每个数组元素的值。这句话毛病太多了。数组名做实参只能传地址。

③在窗体模块的通用声明处可以用 PUBLIC 说明一个全局数组。是不可的。“常数、固定长度的字符串、数组、用户自定义类型以及 Declare 语句不允许作为窗体对象的 public 成员”, 但是在标准模块中可以。

④数组定义语句中可以用负数或小数来指定某一维的维下界或维上界的值。都可以, 可以自动四舍五入调整为整数。

16. 以下有关数组参数的说明中, 正确的是 ()

- A). 在过程中也可用 Dim 语句对形参数组进行说明
- B). 动态数组作为实参时, 可用 REDIM 语句在过程中改变对应形参数组的维界
- C). 调用过程时, 数组名不能作为实参
- D). 数组可以按地址传递也可以按值传递。

标准答案:b

解析:

17. 在过程调用中, 参数的传递可以分为 () 和 () 两种方式

- A). 按位置传递 按参数传递 B). 按地址传递 按参数传递
- C). 按参数传递 按值传递 D). 按值传递 按地址传递

标准答案:d

解析:

18. 在过程定义中用()表示形参的传址。

A). ByRef B). ByVal C). Value D). Var

标准答案:a

解析:

七、函数的调用方法: 可以做函数调用, 也可做过程调用; 过程的调用不能做函数调用。

(一) SUB 过程的调用方法: (SUB 过程一律没有返回值)

不管是通用 SUB 过程, 还是事件过程, 它们的调用方法是一样的。

可以用 call 调用, 也可以省略 call

如果用 call 调用, 则参数必须放在括号中, 如果省略 call, 则参数不能放在括号中。

当然省略 call 时, 如果只有一个参数, 也可括起来, 这里的括起来相当于把这个参数变成了表达式。

(二) FUNCTION 函数的调用方法:

FUNCTION 函数都有返回值, 即使在编写的时候, 没有给函数名赋值, 那么也返回一个默认值 (逻辑型为 false, 数值型为 0, 字符串型为空串), 函数在定义的时候, 函数名没定义类型, 那么返回值就是默认的变体型。

函数即可以作为函数来调用, 也可以按一般的 SUB 过程来调用, 因为 FUNCTION 函数是 SUB 过程的特殊形式。

做函数调用时, 参数必须放在括号里, 调用后一定会得到返回值, 一般需要将返回值赋给一个变量, 或输出, 否则会报错。

按一般 SUB 过程调用时, 不会得到返回值, 调用方法和一般 SUB 过程一样, 即可以加 call, 也可以省略 call, 加 call, 则参数必须放在括号中, 如果省略 call, 则参数不能放在括号中。

任何一个函数都可以当成一般 SUB 过程来调用, 只是由于得不到返回值, 可能实现不了原来的功能, 所以平时我们很少这么用。但有一些特殊的函数, 当成一般 SUB 过程调用, 也能实现部分功能, 比如我们常用的 msgbox 函数和 shell 函数。其实我们平时说这 msgbox 和 shell 既可以做函数使用, 也可以做命令 (语句) 使用, 其实它们实质就是函数, 当成一般 SUB 过程调用, 再省略用 call, 也就是我们平时说的按命令或语句来使用。这时当然没有返回值了, 参数当然不能放在括号中了。

切记：任何 FUNCTION 函数，当成一般的 SUB 过程调用，都没有语法上的错误。

练习题：

1. 在应用程序中，用 " Private Function Fun(x As Integer,y As Single)"定义了函数 Fun

函数 Fun 的过程中的变量 I、J 均定义为 Integer 型，能正确引用函数 Fun 的是()

- ①Fun(I, j) ② Call Fun(1, 3.65) ③Fun(3.14, 234) ④Fun("245", "2315") A). ①
②③ B). ①②③④ C). ①③ D). ②③④

标准答案:d

解析：

形实结合时，如果传地址，参数类型需要完全一样，如果传数值，参数类型需要兼容

①传的是地址，参数类型不完全一致，故不可以。

②传的是数值，参数类型兼容即可，这里加了 Call，当 sub 过程来调用，是可以的。

③传的是数值，参数类型兼容即可。

我们看选项，只有 D 选项，没有①，而①是错误的，所以只能选 D 选项。这里数串和数也是兼容的，记住。

2. 下列程序的输出结果为 ()

```
Private Sub Command1_Click()  
    For i = 1 To 3  
        GetValue (i)  
    Next I  
    Print GetValue(i)  
End Sub  
Private Function GetValue(ByVal a As Integer)  
    Static s As Integer  
    s = s + a  
    GetValue = s  
End Function
```

- A). 4 B). 5 C). 10 D). 11

标准答案:c

解析:function 函数可以当一般 sub 过程调用，而一般的 sub 过程，不能当 function 函数来

调用。

function 函数用 call(或省略)时, 和一般 sub 过程的使用方法是完全一样的。

3. 递归过程既可以是递归 Function 过程, 也可以是递归 Sub 过程

A). 正确 B). 错误 C). D).

标准答案:a

解析:都能是递归。

4. 以下有关过程的说法中, 错误的是 ()

A). 在 SUB 或 function 过程中不能再定义其它 SUB 或 function 过程

B). 在调用过程时, 与使用 Byref 说明的形参对应的实参只能按地址传递方式结合

C). 递归过程既可以是递归 function 过程, 也可以是递归 SUB 过程

D). 在调用过程时, 形参为数组的参数对应的实参只能是数组。

标准答案:b

解析: 1、在调用过程时, 形参为数组的参数对应的实参只能是数组。该选项绝对正确, 要记住。

2、在调用过程时, 与使用 Byref 说明的形参对应的实参若为变量名, 则传地址, 若为表达式, 则传数值。

5. 以下关于子过程或函数的定义中, 正确的是 ()

A). Sub f1(n As String*1)

B). Sub f1(n As Integer)As Integer

C). Function f1(f1 As Integer)As Integer

D). Function f1(ByVal n As Integer)

标准答案:d

解析:

1、定长字符串不能做形参。

2、过程没有返回值。

3、过程或函数名字不能和形参名字相同

4、function 函数一般有返回值, 但也可以没有返回值。

6. 在 Visual basic 中, SUB 过程的过程名可以有返回值, 而 FUNCTION 过程不能

A). 正确 B). 错误

标准答案:b

解析:FUNCTION 有返回值, SUB 过程没有返回值。

7. Sub 过程与 Function 过程最根本的区别是 ()

- A). Sub 过程的过程名不能返回值, 而 Function 过程能通过过程名返回值
- B). Sub 过程可以使用 Call 语句或直接使用过程名调用, 而 Function 过程不可以
- C). Function 过程可以有参数, Sub 过程不可以
- D). 两种过程参数的传递方式不同

标准答案:a

解析:Sub 过程不能用调用函数的方式调用, 但 Function 过程可以用调用一般过程的方式调用。

一般过程的方式有两种方法: 用 call 时, 参数加括号, 省略 call 时, 参数不能加括号。

8. 下面的过程定义语句中合法的是 ()

- A). Sub Proc1(n) As Integer
- B). Function Proc1(Proc1)
- C). Function Proc1(ByVal n)
- D). Sub Proc1(ByVal n())

标准答案:c

解析:过程名(函数名)都不能与参数同名(形参和实参)

过程不能有返回值

数组做过程的形参, 只能传地址

9. 在 Visual basic 中, 下列对过程的定义正确的是()

- A). Sub F(x%, 3)
- B). Sub FF(x%, y%)As Integer
- C). Function M(ByVal x%)As Boolean
- D). Function MM(x+y)

标准答案:c

解析:一、函数或过程的形参不能是常量、也不能是表过式, 只能是变量。

二、函数有返回值, 过程不能有返回值。

10. 若在应用程序中用“Private Sub Sub1(X As Intege, Y As Single)”定义了子程序 Sub1, 调用程序中的变量 I、J 均为 integer 型, 则正确调用子程序 Sub1 的 Call 语句是()

- ① Call Sub1(I, J)
- ② Call Sub1(3.1415, I)

③ Call Sub1(314, 234)

④ Call Sub1("245", "231.5")

A). ①③ B). ③④ C). ①②③ D). 无

标准答案:b

解析: 形实结合时, 如果传地址, 参数类型需要完全一样, 如果传数值, 参数类型需要兼容

①传的是地址, 参数类型不完全一致, 不可以。

②第二个参数传的地址, 形参和实参的类型不完全一致, 不可以。

③④都传的是数值, 兼容即可。

11. 在应用程序中, 用 " Private Function Fun(x As Integer,y As Single)"定义了函数 Fun

函数 Fun 的过程中的变量 I、J 均定义为 Integer 型, 能正确引用函数 Fun 的是()

①Fun(I, j) ② Call Fun(1, 3.65) ③Fun(3.14, 234) ④Fun("245", "2315")

A). ①②③ B). ①②③④ C). ①③ D). ②③④

标准答案:d

解析:

形实结合时, 如果传地址, 参数类型需要完全一样, 如果传数值, 参数类型需要兼容

①传的是地址, 参数类型不完全一致, 故不可以。

②传的是数值, 参数类型兼容即可, 这里加了 Call, 当 sub 过程来调用, 是可以的。

③传的是数值, 参数类型兼容即可。

我们看选项, 只有 D 选项, 没有①, 而①是错误的, 所以只能选 D 选项。这里数串和数也是兼容的, 记住。

12. 以下两个过程在程序运行后, 单击窗体, 以下说法正确的是()

```
Private Sub ppp(a As Single, b As Single)
```

```
    a=a+b
```

```
    Print a;
```

```
End Sub
```

```
Private Sub Form_Click
```

```
    X=18
```

```
    Call ppp(x+1, (x))
```

```
    Print x
```

End Sub

- A). 实参(x)表示传址调用
- B). 实参 x+1 表示传值调用**
- C). 窗体上显示 37 19
- D). 语句 Call ppp(x+1, (x))可以修改为 Call ppp x+1, (x)

标准答案:b

解析: 实参 x+1 和(x) 都是表达式, 都属于传数值。

13. 下列程序的输出结果为 ()

```
Private Sub Command1_Click()  
    For i = 1 To 3  
        GetValue (i)  
    Next I  
    Print GetValue(i)  
End Sub
```

```
Private Function GetValue(ByVal a As Integer)  
    Static s As Integer  
    s = s + a  
    GetValue = s  
End Function
```

- A). 4
- B). 5
- C). 10
- D). 11

标准答案:c

解析:function 函数可以当一般 sub 过程调用, 而一般的 sub 过程, 不能当 function 函数来调用。

function 函数用 call(或省略) 时, 和一般 sub 过程的使用方法是完全一样的。

14. 在窗体 Form1 中用” Public Sub Fun(x As Integer, Y As Single)” 定义过程 Fun;

在窗体 form2 中定义了变量 i 为 Integer, j 为 Single.

若要在 Form2 的某事件过程中调用 Form1 中的 fun 过程,

则下列语句中正确的语句有 () 个。

- ① Call Fun(I, j)
- ② Call Form1.Fun(I, j)

③ Form1.Fun(i), j

④ Form1.Fun i+1, j

A). 1 B). 2 C). 3 D). 4

标准答案:c

解析: 首先调用其它窗体中的过程, 必须加上窗体名称 (和调用其它窗体中的变量须加上窗体名称一样)。所以①不正确

其次形实结合时, 如果传地址, 参数类型需要完全一样, 如果传数值, 参数类型需要兼容

还有用 call 调用, 参数放在括号里, 省略 call , 参数不能放在括号里。

②③④都满足条件。

15. 假设已通过下列 Sub 语句声明了 proc1 过程, 若要调用该过程, 可以采用 () 语句

Sub Proc1(x As Integer)

A). s=Proc1(3) B). Proc1 32765
C). Print Proc1 (100) D). Call Proc1 (40000)

标准答案:b

解析:

过程没有返回值, 所以 A 选项和 C 选项不正确。

过程可用 call 调用, 也可以省略 call。如果用 call 调用, 则参数必须放在括号中, 如果省略 call, 则参数不能放在括号中。

该过程的形参是整型, 实参 40000 超过了整型数的范围, 而 32765 在整型数范围内。

16. 假定已定义了一个过程 Sub Add(a As Single, b As Single), 则正确的调用语句是 ()。

A). Call Add (12, 12, x) B). Add 12, 12
C). Call Add 2*x, Sin(1.57) D). Call Add x, y

标准答案:b

解析:调用过程的方式有两种:

第一种,用 call 调用, 参数必须写在括号中

第二种, 省略 call, 参数不能放在括号中

注意,如果只有一个参数, 省略了 call, 而参数也加了括号, 这时候不能算错, 因为这里的括号是把参数变成了表达式。请同学们把这句话考虑明白。

17. 设一个工程由两个窗体组成, 其名称分别为 Form1 和 Form2, 在 Form1 上有一个名称为

Command1 的命令按钮。窗体 Form1 的程序代码如下：

```
Private Sub Command1_Click()  
    Dim a As Integer  
    a=10  
    Call g(Form2, a)  
End Sub  
  
Private Sub g(f As Form, x As Integer)  
    y=IIf(x>10, 000, -100)  
    f.Show  
    f.Caption=y  
End Sub
```

- A). Form1 的 Caption 属性值为 100 B). Form2 的 Caption 属性值为-100
C). Form1 的 Caption 属性值为-100 D). Form2 的 Caption 属性值为 100

标准答案:b

解析:

这里过程 g 的参数 f 是一个指向窗体对象的变量，这种用法不是很多，但我们要会用。也就是说 f 代表一个窗体。

这里调用过程 Call g(Form2, a), 用窗体 2 (Form2) 做实参, 通过执行过程中的语句 f.Show, 将窗体 2 显示出来, 通过 f.Caption=y 语句, 将 y 的值 (-100) 赋给窗体 2 的标题属性。

18. 在运行时, 以下[]方法不可以执行命令按钮(name 属性为 command1)的 Click 事件过程。

- A). 使用命令按钮的访问键
B). 在代码中执行语句: call command1_click()
C). 在代码中执行语句: command1.value=true
D). 使用命令按钮的快捷键

标准答案:d

解析:

19. 若在应用程序中用“Private Sub Sub1(X As Intege, Y As Single)”定义了子程序 Sub1, 调用程序中的变量 I、J 均为 integer 型, 则正确调用子程序 Sub1 的 Call 语句是()

- ① CaLL Sub1 (I, J)
 - ② CaLL Sub1 (3. 1415, I)
 - ③ CaLL Sub1 (314, 234)
 - ④ CaLL Sub1 ("245" , "231. 5")
- A). ①③ B). ③④ C). ①②③ D). 无

标准答案:b

解析：形实结合时，如果传地址，参数类型需要完全一样，如果传数值，参数类型需要兼容

- ①传的是地址，参数类型不完全一致，不可以。
- ②第二个参数传的地址，形参和实参的类型不完全一致，不可以。
- ③④都传的是数值，兼容即可。

八、inputbox 函数、msgbox 函数与命令、shell 函数与命令的使用。

（一）inputbox 函数：（输入框函数）

其返回值为字符串型。该函数可有多个参数，平时用的最多的是前三个参数，只有第一个参数是不能省略的。第一个参数是输入框的提示信息，第二个参数是输入框的标题，第三个参数是输入框的默认值。如 X=Inputbox(a,b ,c)。使用时，这三个参数也可以用变量代替。如 a\$="请输入 X 的值",b\$="输入",c\$="abc" ，最后的效果一样。

（二）msgbox 函数与命令：（消息框函数或命令）

msgbox 即可以做函数使用，也可做命令使用。做函数使用有返回值，需要将该返回值赋给一个变量（当然也可以直接 print 输出，很少这么用），做命令使用，没有返回值。

Msgbox 做函数使用，其返回值是一个整形数。不管是做函数还是做命令使用，其可有多个参数，最常用的也是三个，只有第一个参数不能省略，第一个参数是消息框的提示信息，第二个参数是确定有哪几个按钮、焦点落在哪个按钮上、消息框的图标（叹号、错号等），第三个参数是消息框的标题。

做函数用必须有返回值，且参数必须有括号括起来。做语句用没有返回值，且参数不能用括号括，当然做语句用时如果只有一个参数，也可括起来，这里的括起来相当于把这个参数变成了表达式。

即可做函数又可做语句使用的还有一个 shell,和 msgbox 的用法是一样的。

msgbox 的返回值是整型一共有 7 个，

点“确定”按钮返回值是 1、点“取消”按钮返回值是 2、点“终止”按钮返回值是 3，

点“重试”按钮返回值是4、点“忽略”按钮返回值是5、点“是”按钮返回值是6，点“否定”按钮返回值是7

(三) shell 函数与命令:

shell 即可以做函数使用，也可以做语句使用，其作用是调用外部的可执行文件。

shell 格式为:

p=shell("c:\123.exe",2),只有两个参数,第一个参数不能省略,第二个参数可以省略。

第一个参数是可执行的外部文件,包括盘符、路径、文件名,第二个参数是运行时的窗口的状态,有三个值,0(或省略)表示标准窗口,1表示最小化窗口,2表示最大化窗口。

做函数用必须有返回值,且参数必须有括号括起来。做语句用没有返回值,且参数不能用括号括,当然做语句用时如果只有一个参数,也可括起来,这里的括起来相当于把这个参数变成了表达式。

Shell 返回值是一个数值,表示所打开文件的句柄。

练习题:

1. 用 inputbox 函数可以帮助生成 ()

- A). 消息框 B). “保存”对话框 C). **输入对话框** D). “颜色”对话框

标准答案:c

解析:

2. 下面表达式返回值的类型不同的一项是 ()。

- A). Chr(66) B). String(4,"X")
C). **Val("34.5")** D). inputBox("请输入:")

标准答案:c

解析:只有 Val 函数返回的是数值,而其它几个返回的都是字符串。

3. 执行下列代码,分别输入 12 和 89,输出的结果是 ()。

```
Private Sub Form_click()  
    a = InputBox("D")  
    b = InputBox("E")  
    Print a + b  
End Sub
```

- A). 101 B). DE1289 C). DE D). **1289**

标准答案:d

解析:InputBox 函数的返回值是字符串。

4. 在 Visual Basic 中, 设有语句:

```
x= InputBox("输入数值", "20", "教育")
```

程序运行后, 如果从键盘上输入数值 10, 并按 Enter 键, 则下列叙述中正确的是()

- A). 变量 x 的值是数值 10 B). 变量 x 的值是字符串"10"
C). 20 是默认值 D). 在 InputBox 对话框标题栏中显示的是"教育"

标准答案:b

解析:inputbox() 函数有三个参数, 第一个参数不能省略, 为提示信息; 后面的两个参数都可以省略, 第二个参数为输入框的标题, 第三个参数为默认值。该函数的返回值为字符串型。

5. 下面有关语句 a=msgbox("是否继续", vbokcancel, "退出") 的说法中不正确的是()

- A). 单击"确定"按钮返回值是 1 B). 单击"取消"按钮返回值是 0
C). 消息框的标题为"退出" D). 变量 a 的值是数值型。

标准答案:b

解析:

1、msgbox 的返回值是整型, 第三个参数是消息对话框的标题。

2、msgbox 的返回值是整型一共有 7 个,

点"确定"按钮返回值是 1、点"取消"按钮返回值是 2、点"终止"按钮返回值是 3
点"重试"按钮返回值是 4、点"忽略"按钮返回值是 5、点"是"按钮返回值是 6
点"否定"按钮返回值是 7

6. MsgBox 函数建立的消息框能显示信息也能输入信息。()

- A). 正确 B). 错误 C). D).

标准答案:b

解析:MsgBox 函数建立的消息框不能输入信息。

7. MsgBox 函数建立的消息框能显示信息也能输入信息。()

8. 下列叙述不正确的是()

- A). InputBox 函数返回值为字符型 B). MsgBox 函数返回值为数值型
C). MsgBox 语句返回值为字符型 D). len 函数返回值为数值型

标准答案:c

解析:MsgBox 语句没有返回值。

9. 已知 P= MsgBox("Howareyou?", VbYesNo+ VbQuestion, "Hello!"), 变量 P 是 Integer

数据类型。

A). 正确 B). 错误

标准答案:a

解析:MsgBox 函数的返回值为整型变量。

10. inputBox 函数返回值的类型为 。

A). 数值 B). 变体 C). 字符串 D). 不确定

标准答案:c

解析:inputBox 函数返回值的类型为 字符串

msgbox 函数返回值的类型为 整型

11. MsgBox 函数的返回值的类型是 ()

A). 逻辑型 B). 日期型 C). 字符串 D). 整型

标准答案:d

解析:inputBox 函数返回值的类型为 字符串

msgbox 函数返回值的类型为 整型

12. 下列窗体的()方法或函数可以调用外部的可执行文件。

A). Open B). Input C). Show D). Shell

标准答案:d

解析:shell 即可以做函数使用,也可以做语句使用,其作用是调用外部的可执行文件。

shell 格式为:

p=shell("c:\123.exe",2),第一个参数是可执行的外部文件,包括盘符、路径、文件名,第二个参数是运行时的窗口的状态,有三个值,0(或省略)表示标准窗口,1表示最小化窗口,2表示最大化窗口。

做函数用必须有返回值,且参数必须有括号括起来。做语句用没有返回值,且参数不能用括号括,当然做语句用时如果只有一个参数,也可括起来,这里的括起来相当于把这个参数变成了表达式。

即可做函数又可做语句使用的还有一个 msgbox,和 shell 的用法是一样的。

13. 下列叙述不正确的是 ()

A). MsgBox 语句返回值为所单击按钮的键值 B). MsgBox 函数返回值为所单击按钮的键值
C). Inputbox 函数返回值为字符型 D). SHELL 函数的返回值为所执行文件的工作代码

标准答案:a

解析:

14. 以下关于 MSGBOX 的叙述中, 错误的是 ()

- A). MSGBOX 函数返回一个整数
- B). MSGBOX 函数的第二个参数是一个整数, 该参数只能确定对话框中显示的按钮的数量。
- C). MSGBOX 语句没有返回值
- D). 通过 MSGBOX 函数可以设置信息框中图标和按钮的类型

标准答案:b

解析:msgbox 即是一个语句, 也是一个函数

msgbox 做函数使用时必须有返回值 (必须将函数赋给一个变量), 参数必须放在括号里。

msgbox 做语句使用时不能有返回值, 参数不能放在括号里, 当然如果只有第一个参数, 其它都省略的话, 那么这第一个参数放在括号里也不算错 (相当于一个表达式)。

msgbox 可以有三个参数, 第一个参数是提示信息, 是必须的, 不能省略; 第三个参数是消息框的标题, 可以省略; 第二个参数确定消息框的按钮类别、消息框的图标 (叹号、错号等) 及焦点落在哪个按钮上, msgbox 函数返回一个整形数值, 返回 1 表示按的是确定, 按取消返回 2, 终止 (3) 重试 (4) 忽略 (5) 是 (6) 否 (7)

15. 在 Visual Basic 的窗体上添加一个命令按钮, 名称为 Command1 单击该命令按钮时, 执行如下事件过程:

```
Private Sub Command1_Click()
```

```
A="Software and hardware"
```

```
B= Right(a, 8)
```

```
C= Mid(a, 1,8)
```

```
MsgBox a, 1, b
```

```
End Sub
```

在弹出信息框的标题栏中显示的信息是 ()

- A). software and hardware B). software
- C). 1 D). hardware

标准答案:d

解析:消息框有三个参数, 第一个不能省略, 为提示信息

第二个参数可以省略, 确定消息框的按钮个数、消息框图标、及焦点落在哪个按钮上。

第三个参数可以省略, 为消息框的标题

消息框和输入框的参数顺序是不一样的。

输入框（inputbox 函数）的第一个参数也是提示信息，不能省略，第二个参数是输入框的标题，第三个参数是默认值，后两个参数可以省略。

msgbox 既是一个语句，也是一个函数

msgbox 做函数使用时必须有返回值（必须将函数赋给一个变量），参数必须放在括号里。

msgbox 做语句使用时不能有返回值，参数不能放在括号里，当然如果只有第一个参数，其它都省略的话，那么这第一个参数放在括号里也不算错（相当于一个表达式）。

msgbox 可以有三个参数，第一个参数是提示信息，是必须的，不能省略；第三个参数是消息框的标题，可以省略；第二个参数确定消息框的按钮类别、消息框的图标（叹号、错号等）及焦点落在哪个按钮上，msgbox 函数返回一个整形数值，返回 1 表示按的是确定，按取消返回 2，终止（3）重试（4）忽略（5）是（6）否（7）

九、文件操作中#号的省略用法

切记：(1)文件号前面的#号，不能省略的有：

向文件写数据 `print #`、`write #` 和读数据 `input #`、`line input #`

(2) 文件号前面的不能带#号的，都是常用的函数，如 `eof(1)`、`lof(1)`、`seek(1)`

(3)平时习惯写可以省略的：打开文件时的文件号，关闭文件时的文件号，向随机及二进制文件读写数据的 `get` 文件号和 `put` 文件号

(4)平时不习惯写可以写上的：`input(长度,文件号)`，这里的文件号平时不写，写上也不错。

(5)再一次强调：

如果是改错题，如果你看了我今天的分析，并且记性好，记住了，那就没问题了。如果还是记不住，那么先不考虑文件号的问题，先找其它错误，如果找不到其它错误了，再找文件号的错误。

如果是填空题：我们平时习惯带的，填空时一定要带上，平时习惯不带的，填空时一定不要带上，即便是写不写都可以的，也要按平常的习惯，千万不要别出心裁。

十、VB 的 Tabindex、 tabstop 属性分析

(一) Tabindex 属性：

窗体上的所有控件，只要其能获得焦点，那么这个控件就有 TabIndex 属性，TabIndex 的值从 0 开始，哪个对象建立的早，哪个对象的 TabIndex 属性值更小，但可以修改，且这些控件的 TabIndex 值都不相同。

该属性的意义在于，按 TAB 键时，焦点按 TabIndex 值的顺序在各个控件之间切换。

(二) tabstop 属性：

其本意是按 TAB 键时，能停在这里，其默认值为真。若设置某对象的 TabStop 属性为 false，则按 TAB 键不会停在这个对象上（即跳过这个对象）。但只是按 TAB 键不接收焦点（不停下来），用鼠标点可以获得焦点。当然也可以从键盘上输入数据。

要想让某对象按 TAB 键或用鼠标点都不能获得焦点，可以将该对象的 Enabled 设置为假，不可用了，当然就不能获得焦点了，这里与对象的 tabstop 属性的真与假没关系。

(三) 程序刚启动时，哪个对象先获得焦点：

虽然窗体有 setfocus 方法，但窗体没有 Tabindex 和 tabstop 属性。窗体能获得焦点，窗体获得焦点实质上是窗体成为活动窗口，窗体获得焦点成为活动窗体后，焦点马上转移到窗体的对象上。所以窗体启动后，窗体上哪个对象的 Tabindex 属性为 0，哪个对象先获得焦点。但如果该对象的 tabstop 属性为假（默认为真），则会转跳到下一个对象上（TabIndex 属性为 1 的那个对象），依次类推。

练习题：

1. 单击滚动条的中间区域时，滑块的改变量是由（ ）设定。

- A). largechange B). Smallchange
- C). Scroll D). Tabstop

标准答案:a

解析:单击滚动条的中间区域时，滑块的改变量是由 largechange 设定。

单击滚动条箭头，滑块的改变量

2. 若在一个应用程序窗体上，依次创建了 CommandButton、 TextBox、 Label 等控件，运行该程序显示窗体时，（ ）会首先获得焦点

- A). 窗体 B). CommandButton
- C). Label D). TextBox

标准答案:b

解析：首先题目问的是窗体上的三个对象哪一个会先获得焦点，所以与 A 选项无关

具有 TabIndex 属性的所有对象，哪个建立的早，哪个对象的 TabIndex 属性值更小，

所以 CommandButton 的 TabIndex 属性值最小，为 0。

标签不能获得焦点。

3. 一个窗体上有三个文本框，按放置顺序分别是 Text1、Text 2、Text3, 若使程序运行时 Text3 首先获得“焦点”，那么，应设置其()属性值为 0

- A). Index B). Tabindex C). Tag D). Top

标准答案:b

解析:哪一个控件的 Tabindex 属性值为 0，启动后哪一个控件先获得焦点。

4. 在 Visual Basic 中，以下关于焦点的叙述中，错误的是 ()

- A. 如果文本框的 TabStop 属性为 false，则不能接收从键盘上输入的数据
B. 当文本框失去焦点时，触发 lostfocus 事件
C. 当文本框的 enabled 属性为 false 时，其 tab 顺序不起作用
D. 可以用 TabIndex 属性改变 TAB 顺序。

标准答案:a

解析:

窗体上的所有控件，只要其能获得焦点，那么这个控件就有 TabIndex 属性，且这些控件的 TabIndex 值都不相同，TabIndex 的值从 0 开始，该属性的意义在于，按 TAB 键时，焦点按 TabIndex 值的顺序在各个控件之间切换。

若设置某控件的 TabStop 属性为 false (默认为 true)，按 TAB 键时该控件不会接收焦点 (在这个控件上不停)。但只是按 TAB 键不接收焦点，用鼠标点可以获得焦点。当然也可以从键盘上输入数据。

若设置某控件的 Enabled 属性为 false (默认为 true)，因为该控件不可用，所以按 TAB 键不会接收焦点，用鼠标点也不会接收焦点。

5. 窗体上已建立了多个控件，如 Text1、Label1、Command1, 若要使程序一运行焦点就定位在 Command1 控件上，应将 Command1 的【 】属性设置为【 】。

- A). Index 1 B). TabStop True
C). TabIndex 0 D). Enabled false

标准答案:c

解析:

6. 窗体上有若干命令按钮和一个文本框，程序运行时焦点置于文本框中，为了在按下回键时执行某个按钮的 Click 事件过程，需要将该按钮的()属性设置为 True

A). Cance B). Default C). Enabled D). Visible

标准答案:b

解析:

7. 如果将文本框的【 】属性设置为 True, 则运行时不能对文本框中的内容进行编辑。

A). Visible B). TabStop C). Locked D). MultiLine

标准答案:c

解析:

8. 为了防止用户随意将光标置于控件上, 应【 】。

A). 将控件的 TabStop 属性设置为 False B). 将控件的 Enabled 属性设置为 False

C). 将控件的 TabStop 属性设置为 True D). 将控件的 TabIndex 属性设置为 0

标准答案:b

解析:

9. 要将焦点设置在某个控件上, 以下四个选项中正确的是 ()

A). 只能使用鼠标直接点击该控件 B). 本题中三选项中涉及的方法均可用

C). 只能在程序中调用该控件的 setfocus 方法 D). 只能使用 TAB 键将焦点移到该控件上

标准答案:b

解析:

10. 在窗体上有若干控件, 其中有一个名称为 Text1 的文本框。影响 Text1 的 Tab 顺序的属性是()

A). Enabled B). TabStop C). Visible D). TabIndex

标准答案:d

解析:

11、程序启动后, 希望将焦点获在 text1 上, 可以用以下哪种方法实现。

A.在窗体的 load 事件中, 加上语句 text1.setfocus

B.在设计时, 将 text1 的 TabIndex 属性设置为 1

C. 在设计时, 将 text1 的 TabIndex 属性设置为 0, TabStop 属性设置为假

D. 在设计时, 将 text1 的 TabIndex 属性设置为 0, TabStop 属性设置为真

答案: D

解析:

在窗体的 LOAD 事件中, 不用使用窗体上某对象的.setfocus 方法, 这样做报错。

对象的 TabIndex 属性最小值从 0 开始。当某对象的 TabStop 属性设置为假的时候，按 TAB 键不能获得焦点。

十一、VB 的 default 属性和 cancel 属性分析。

default 为命令按钮的特有属性(单选按钮和复选按钮没此属性),一个窗体上只能有一个命令按钮的 default 为真，即便放在多个图片框里也只能有一个为真。设置此属性为真的意义在于，如果焦点不在按钮上（比如在文本框中），敲回车，则执行 default 为真的那个命令按钮的单击事件。（注意：如果焦点在某个命令按钮上，则敲回车，执行的则是具有焦点的那个按钮的单击事件，故此题说的有点毛病）。

与此属性类似的还有一个 cancel 属性，当按 ESC 键时，会执行 cancel 属性为真的那个命令按钮的单击事件。

众多命令按钮，它们的 default 属性可以都为假，但为真的只能有一个。（cancel 理同），一般而言某按钮的 default 为真，再设置其 cancel 为真没实际意义，但可以设置此二属性同时为真。

练习题：

1. 下列控件属性中，属性值的类型不相同的一项是（）。

- A). Label 控件的 Enabled 属性与 TextBox 控件的 Enabled 属性
- B). Command 控件的 Default 属性与 Command 控件的 Cancel 属性
- C). **OptionButton 控件的 value 属性与 CheckBox 控件的 value 属性**
- D). Command 控件的 Visible 属性与 Form 控件的 Visible 属性

标准答案:c

解析：所在对象的 Enabled 属性和 Visible 属性都是逻辑值。

OptionButton 控件的 value 属性为逻辑值，而 CheckBox 控件的 value 属性为 0、1、2

2. 在 Visual Basic 中，若已把一个命令按钮的 default 属性设置为 true，则下面可导致按钮的 click 事件过程被调用的操作是（）

- A). 用鼠标右键单击此按钮
- B). 按键盘上的 ESC 键
- C). 按键盘上的回车键
- D). 用鼠标右键双击此按钮

标准答案:c

解析：希望同学们看一看，这个属性很重要，好长时间没做这种题了。

default 为命令按钮的特有属性(单选按钮和复选按钮没此属性),一个窗体上只能有一个命令按钮的 default 为真,即便放在多个图片框里也只能有一个为真。设置此属性为真的意义在于,如果焦点不在按钮上(比如在文本框中),敲回车,则执行 default 为真的那个命令按钮的单击事件。(注意:如果焦点在某个命令按钮上,则敲回车,执行的则是具有焦点的那个按钮的单击事件,故此题说的有点毛病)。

与此属性类似的还有一个 cancel 属性,当按 ESC 键时,会执行 cancel 属性为真的那个命令按钮的单击事件。

众多命令按钮,它们的 default 属性可以都为假,但为真的只能有一个。(cancel 理同),一般而言某按钮的 default 为真,再设置其 cancel 为真没实际意义,但可以设置此二属性同时为真。

3. 窗体上有若干命令按钮和一个文本框,程序运行时焦点置于文本框中,为了在按下回键时执行某个按钮的 Click 事件过程,需要将该按钮的()属性设置为 True

A). Cance B). Default C). Enabled D). Visible

标准答案:b

解析:

4. 在 Visual basic 中,为了在按下 Enter 键时执行某个命令按钮的事件过程,需要把该命令按钮的一个属性设置为 True,这个属性是 ()

A). Cancel B). Default C). Value D). Enabled

标准答案:b

解析:一、把某按钮的 default 属性设置为 True,若焦点在文本框中,此时按回车键,会执行该按钮的单击事件。(如果焦点不在文本框中,而是在其它的按钮上,敲回车,则会执行具有焦点的那个按钮的单击事件。)

二、按钮还有一个和 default 属性相似的属性,叫 cancel 属性,当某按钮的 cancel 属性设置为 True,则按 ESC 键,会触发该按钮的单击事件。

三、如果有多个按钮,只能有一个按钮的 default 属性为真,但可以都为假;cancel 属性亦然;也可以将同一个按钮的 default 和 cancel 都设置为真,那没什么意义。

十二、赋值语句的计算和转化功能

1、具有方向性,表示把右边的值赋给左边的变量

2、先计算后赋值。可以进行相应转换后为相容类型的变量或控件属性赋值。切记。

3、左边必须是变量，右边可以是表达式。

4、两个等号连用，第一个是赋值符，第二是关系运算符

5、在整个语句中前面没有任何语句定义符，这时的第一个等号为赋值符号，而如果前面有其它的话句定义符，这时的等号都是关系运算符。

练习题：

1. 在某过程中已说明变量 a 为 integer 类型，变量 s 为 String 类型，过程中的以下四组语句中，不能正常执行的是()

A). $s=2*a+1$ B). $s="237" \& ".11"$: $a=s$

C). $s= 2*a>3$ D). $a=2 : s=16400*a$

标准答案:d

解析：

赋值语句，可以先计算，再转化，再赋值，所以这几个选项，这么赋值都可以。只是 $a=2$, $16400*2=32800$, 整形数除一个常数最后还是一个整形数，结果溢出。

2. 下列说法不正确的是()

A). VB 允许将一个数字字符串赋值给一个数值型的变量

B). VB 允许使用未经说明的变量，其类型都是 Variant 类型

C). Cls 方法只清除运行时在窗体或图片框中显示的文本或图形

D). 事件过程既可建在窗体模块中，也可建在标准模块中

标准答案:d

解析:事件过程只能建立在窗体模块中，标准模块中的过程只能是通用 sub 过程，或通过 function 函数。

3. 下列不是合法的赋值语句的是 ()

A). $Inumber_1=45$ B). $(4+5)*3.5=num$

C). $Labell.caption=time$ D). $Num=(4+5)*3.5$

标准答案:b

解析:赋值语句的左边只能是变量名。

4. 在程序中分别将变量 Inta、B1、St 和 D 定义为整型、布尔型、字符串型和日期型，下列赋值语句在执行时会出错的是()

A). $Inta=4.6$ B). $B1=#True\#$

C). $St=5 \& 123.5$ D). $D=#10/05/01\#$

标准答案:b

解析:赋值语句可以先计算, 并且进行适当的类型转换以后再赋值。

#是日期型常量的定界符。

5. 赋值语句的功能是计算表达式并转换为相同类型数据后为变量或控件属性赋值。()

A). 正确 B). 错误 C). D).

标准答案:a

解析:赋值语句, 支持先计算, 后赋值, 可以进行相应转换后为相同类型的变量或控件属性赋值。切记。

6. 关于语句 If x=1 Then y=1, 下列说法正确的是[]。

A). x=1 为赋值语句, y=1 为关系表达式 B). x=1 为关系表达式, y=1 为赋值语句
C). x=1 和 y=1 均为关系表达式 D). x=1 和 y=1 均为赋值语句

标准答案:b

解析:

7. 假设变量 boolVar 是一个布尔型变量, 则下面正确的赋值语句是【 】。

A). boolVar=3<4 B). boolVar=#True#
C). boolVar=' True' D). boolVar=. True.

标准答案:a

解析:

8. 下列正确的赋值语句是 ()。

A). -m=n+1 B). m=m-1 C). m+1=m D). 3m=n+x

标准答案:b

解析:

9. 在 Visual Basic 中, 设有如下变量声明语句:

Dim TestDate As Date, 可以为变量 TestDate 正确赋值的表达式是 ()

A). TestDate=#1/1/2002#
B). TestDate=Format("m/d/yy", #1/1/2002#)
C). TestDate=date("1/1/2002")
D). TestDate="1/1/2002"

标准答案:a

解析:一、date() 该函数的功能是返回当前的日期, 该函数的参数为空, 加上参数是错误的。

二、Format("m/d/yy",#1/1/2002#),在这里的format是格式化输出函数,但使用有误,应该写做Format(#1/1/2002#,"m/d/yy"),意思是输出时按月一位(如果月只有一位的话)、日一位(如果日只有一位的话)、年两位,中间用/分隔。format函数一般配合print语句使用。

如 print Format(#1/13/2002#,"m/d/yy") 输出结果为: 1/13/02

format函数的返回值类型为字符型,也可以将其返回值赋给一个字符串变量:如 ab\$=Format(#1/13/2002#,"mm-dm-yyyy"),则 ab="01-13-2002"

如果强制将其返回值赋给一个日期型变量,会将format函数的结果做强制转换,没有语法错误,但失去了格式化的意义。

三、format格式化输出函数的格式为(原始数据,"格式串"),即原始数据在前,格式串在后,格式串中用双引号引起来

十三、len函数遇到非字符串变量

len函数的参数可以是字符串常量或字符串变量,用以返回字符串的长度,也可以是非字符串类型的变量(这时不能是常量),此时返回值为该种类型的变量的长度。整型变量长度2个字节,单精度变量长度4个字节等。

如: **dim a as integer**

Dim b as double

Dim c as string *5

则 len(a)=2 , len(b)=8 , len(c)=5(定长字符串,长度永远不会变)

练习题:

1. 下列叙述不正确的是()

- A). InputBox 函数返回值为字符型 B). MsgBox 函数返回值为数值型
C). MsgBox 语句返回值为字符型 D). len 函数返回值为数值型

标准答案:c

解析:MsgBox 语句没有返回值。

2. Dim x!=2.531 : Print Len (X)的运行结果是()

- A). 2 B). 4 C). 5 D). 出错

标准答案:b

解析: len函数的参数可以是字符串,也可以是非字符串类型的变量,此时返回值为该种类

型的变量的长度

3. 执行下面语句后，Len 函数值最大的是()

```
Dim IA As Integer, B As Single, S As String*5, Ch As String  
IA=32767  
B=23.5  
S="A"  
Ch="abcd"
```

- A). Len(IA) B). Len(B) C). Len(S) D). Len(Ch)

标准答案:c

解析:

1、len 函数的参数可以是字符串，也可以是非字符串类型的变量，此时返回值为该种类型的变量的长度，整型变量长度 2 个字节，单精度变量长度 4 个字节。

2、定长字符串其长度永远不变。

```
Len(IA)=2    Len(B)=4    Len(S)=5    Len(Ch)=4
```

4. 阅读下面 Visual Basic 程序，最后一行输出结果是()。

```
a=-262:b=5  
c=Len(Str(a)+ Str(b))  
print c
```

- A). 5 B). 4 C). 6 D). 7

标准答案:c

解析:str 函数功能是将数值转化为字符串，但前面有符号位，如果是正数，前面不显示正号，有一个前导空格，如果是负数，前面显示负号，就没了空格了。

十四、对象的 move 方法，及实现相同功能的方法。

move 为方法，几乎所有的对象都有 move 方法，此方法有四个参数，分别为对象移动后的左边界、右边界、宽度和高度，除第一个参数不能省略，其它三个参数可以省略。即便左边界不动，也必须写上左边界。

move 为方法既可实现绝对定位，又可以实现相对定位

绝对定位如：

窗体中有一个标签，执行标签的方法：label1.move 200,300 表示标签移动到距窗体左边

界 200, 上边界 300 的位置。(当然如果标签在图片框中, 执行该语句则移动到距图片框左边界 200, 上边界 300 的位置)

相对定位如: `label1.move label.left, label1.top+300`, 表示标签向上移动 300 个单位。(第一个参数不能省略)

另外, 还可以采用改变对象的 `left` 属性和 `top` 属性值的方法, 实现对象的移动。二者的区别: 前者一条语句可以实现左边界、上边界和移动, 还能改变对象的宽度和高度, 后者不能改变对象的高度和宽度, 同时实现左边界、上边界和移动需要两条语句。

`(label1.left=200 : label1.top=300) ——> label1.move 200,300`

练习题:

1. 使窗体自动向下移动的语句是 `move()`
A). `Left , top+100` B). `Top+100`
C). `Top-100` D). `Top=top+100`

标准答案:a

解析: 要仔细读题, 这道题是让把下面的选项放到 `move` 后面的括号里, 当然只能选择 A 选项, 即

`move left , top+100` , 这里方法和属性前面都省略了窗体对象名。

`move` 为方法, 几乎所有的对象都有 `move` 方法, 此方法有四个参数, 分别该为对象移动后的左边界、右边界、宽度和高度, 除第一个参数不能省略, 其它三个参数可以省略。

2. 在名为 `form1` 的窗体上有一个名称为 `Text1` 的文本框, 程序运行后, 单击窗体, 使文本框距离窗体左边界和上边界均变为 `300twip`, 下列哪些语句能够实现? ()

- A). `Private Sub Form1_Click()`
`Text1.Move 300, 300`
`End Sub`
- B). `Private Sub Form_Click()`
`Text1.Move 300, 300`
`End Sub`
- C). `Private Sub Form_Click()`
`Text1.left=text1.left+300`
`Text1.top=text1.top+300`

End Sub

D). Private Sub Form_Click()

Text1.Height = 300

Text1.Width = 300

End Sub

标准答案:b

解析:

A 选项, 事件名称错误, 窗体的单击事件, 只能是 Form_Click()。

B 选项, 文本框向上向左各移动 300twip, 不符合题意。

D 选项, 只是改变了文本框的高度和宽度。不符合题意。

3.程序的界面如题图所示, 编程实现如下功能: 程序运行时太阳静止不动, 单击“太阳升起”按钮时, 太阳升起(以 0.5s 为时间间隔, 太阳向右移动 80,向上移动 160);单击“停止升起”按钮时, 太阳静止不动



```
Private Sub Command1_Click()
```

```
    Timer1.Enabled = True
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
    Timer1.Enabled = False
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    ( Timer1.Interval = 500 )
```

```
    Timer1.Enabled = False
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
    If Image1.Top > 0 Then
```

```
( Image1.Move Image1.Left + 80, Image1.Top - 160 )
```

```
Else
```

```
Timer1.Interval = 0
```

```
End If
```

```
End Sub
```

4. 设计动画时通常用定时器控件 () 属性来控制动画速度。

A). Move B). Enabled C). Timer D). Interval

标准答案:d

解析:

5. 在 Visual Basic 中, 使窗体 Form1 自动向下移动的语句是 ()

A). Form1.move left, top+100 B). Form1.move top = top+100

C). Form1.move left, top-100 D). Form1.move top+100

标准答案:a

解析:控件的 Move 方法, 第一个参数 (左边界) 不能省略, 后面的三个参数都都可以省略, 所以即便左边界不动, 也必须写上左边界。

向下移动, Top 值增加。

十五、窗体的 moveable 属性

窗体有 moveable 属性, 而其它的对象大部分没有 moveable 属性, 因为 moveable 属性是指在程序运行时, 是否能拖动对象, 程序运行时, 可以拖动窗体, 但不可能拖动窗体上的对象。

窗体的 moveable 属性默认为真, 即可以拖动窗体, 所以平时窗口是可以移动的, 如果我们将该属性设置为假, 则窗口不能移动。

练习题:

1. 若要禁止窗体被移动, 则可以设置窗体的 Move 属性为 False 来实现。

A). 正确 B). 错误 C). D).

标准答案:b

解析:

move 为方法, 几乎所有的对象都有 move 方法, 此方法有四个参数, 分别为对象移动后的左边界、右边界、宽度和高度, 除第一个参数不能省略, 其它三个参数可以省略。

窗体还有一个 moveable 属性, 窗体有该属性, 而大部分对象无该属性。此属性默认值为

true, 窗体可以被移动, 如将该属性设置为 false, 则窗体不能被移动。

本题目中, 若要禁止窗体被移动, 则可以设置窗体的 Moveable 属性为 False 来实现。

2. 如果在运行时不允许移动窗体, 可以将窗体的[]属性设置为 False。

- A). ControlBox B). Moveable
C). WindowState D). BordorStyle

标准答案:b

解析:

十六、数组使用的注意事项

1、VB 中的数组必须先定义, 后使用。

2、dim 数组名 (下界, 上界) as 类型, 下界和上界可以是小数, 也可以为负数, 若为小数自动四舍五入取整, 下界必须<上界。

如 dim (-2.67 to 1.34) as integer 即相当于 dim (-3 to 1) as integer 有 a(-3)、a(-2)、a(-1)、a(0)、a(1)五个元素。

再如 dim (8 to 5) as integer 是不合法的, 因为这里下界大于上界。dim (5 to 8) as integer 可以。

3、定义数组时, 我们通常省略下界, 省略了下界, 默认下界为 0, 如 dim b(6) as integer, 相当于 dim b(0 to 6) as integer, 有 7 个元素, 但是如果前面有语句 option base 1, 则下界为 1。

如 option base 1 : dim b(6) as integer , 则这里相当于 dim b(1 to 6) as integer。

4、option base 语句后面只能写 1 或 0, option base 1 表示后面定义数组时, 如果省略下界, 下界从 1 开始; option base 0 表示后面定义数组时, 如果省略下界, 下界从 0 开始; 如果没写 option base 语句, 即表示省略了 option base 0 这条语句, 下界也从 0 开始。

5、常规数组, 只需要用一条 dim 语句定义, 用 dim 定义常规数组时, 上界和下界都不能是变量。如 dim a(n), 这里如果 n 为变量是错误的。

6、动态数组, 首先用 dim a() as 类型, 定义类型, 再用 redim 来改变数组的维数和元素个数, 且可以多次改变。切记类型不能改变, redim 语句里一般不写 as 类型, 但如果写的和上面一开始定义时的类型一样, 也不算错误。

7、特殊情况时, 没有最开始的 dim a() as 类型语句, 直接用 redim a() as 类型, 也可以。这相当于可以省略最开始的 dim 定义。改错题有这样的题目。

8、动态数组, 随时可以用 redim 语句改变原数组的维数和元素个数, 但是一旦用 redim 重新定义, 则所有元素的值都会清空成默认值 (如果为数值型, 则清空为 0), 如果想保留原数组

的值,那么需要使用 `preserve` 参数,并且只能改变最后一维的上界。如:4、`redim preserve a(2,5)` 这样数组 `a` 原来的元素的值可以保留。

9、数组常用的函数有:

`Lbound(a,2)` 返回数组 `A`,第 2 维的下界

`Ubound(a,2)` 返回数组 `A`,第 2 维的上界

`Lbound(a)` 返回数组 `A` 的下界 (没有维数默认是一维)

这两个函数主要是用于测试动态数组的上下界,当然也可以用于测试常规数组上下界,没有语法错误。

10、`erase` 数组名

对于动态数组而言,收回该数组分配的存储空间,即使用完该语句后,相当于刚用 `dim b() as integer` 定义完成一样,现在只定义了数组 `b` 是一个整型数组,没有任何元素。

该语句也可用于常规数组,用完以后,常规数组的元素不变,但内容都变成了默认值(数值型的数组为 0,字符型的为空串)。

11、`array` 函数的使用。

变量名=`array(参数, 参数……)`

如 `a = Array(12, 34, 56, "asc", False, #5/31/2020#)`

`array` 函数使用前,变量(如上例中的 `a`)只能定义成变体型变量或变体型数组(如 `dim a`,或 `dim a as Variant`,或 `dim a() as Variant`),通常定义成变体型变量。当然不定义,直接使用,即默认为变体型变量。参数可以是同一种类型,也可以是多种类型。也就是说这样生成的数组,各个元素的类型可以不一样。切记,只有这样生成的数组,数组各元素可以为不同类型,其余情况下的数组各元素必须为同一数据类型。

用 `array` 函数生成的数组,下标从 0 还是 1 开始,取决于 `option base` 语句。

12、数组做形参,只能是动态数组,不能是常规数组,且只能传地址,即不能用 `byval` 修饰(`byref` 常省略),而实参必须用数组名,可以加空括号,也可以不可空括号。

13、数组不允许作为窗体模块的 `public` 成员,也就是说在窗体的通用声明字段不能定义全局类型的数组,动态数组和常规数组都不行,但可以在标准模块的通用声明字段定义全局型的数组变量。

14、定长字符串不能做形参,但定长字符串数组可以做形参,当然需要实参也必须是定长字符串数组,长度可以不同。

举例:

1、以下有关数组的说法中，错误的是（）

- A.用 `redim` 语句重新定义动态数组时，其下标的上下界可以使用赋了值的变量。
- B. 用 `redim` 语句重新定义动态数组时，不能改变已经说明过的数组的类型
- C.定义数组时，数组的上下界必须是整数
- D.使用 `erase` 语句可以回收动态数组的存储空间。

答案：c

2. 下面关于过程参数的说法，错误的是（）

- A). 过程的形参不可以是定长字符串类型的变量
- B). 形参是定长字符串型的数组，则对应的实参必须是定长字符串型数组，且长度相同
- C). 若形参是按地址传递的参数，形参和实参也可以按值传递方式进行形实结合
- D). 按值传递参数，形参和实参的类型可以不同，只要相容即可

标准答案:b

解析：对于我们不太熟悉的问题，要使用排除法，学会分析。

1、过程的形参可以是定长字符串类型的数组，但不能是定长字符串类型的变量，所以 A 正确。

2、形参按地址传递的参数，形实结合也可按数值传递。所以 C 正确。

3、形实结合按值传递，无须类型完全相同，相容即可，所以 D 也正确。

再看选项 B，我们不太清楚，其它三项绝对正确，那么 B 选项看着别扭，那么就是 B 错误了。

说明：形参是定长字符串类型的数组，那么实参也必须是定长字符串数组，但是长度可以不同。

3. 以下说法不正确的是（）。

- A). 使用不带关键字 `Preserve` 的 `ReDim` 语句可以重新定义数组的维数
- B). 使用不带关键字 `Preserve` 的 `ReDim` 语句可以改变数组各维的上、下界
- C). 使用不带关键字 `Preserve` 的 `ReDim` 语句可以改变数组的数据类型
- D). 使用不带关键字 `Preserve` 的 `ReDim` 语句可以对数组中的所有元素进行初始化

标准答案:c

解析：带不带关键字 `Preserve` 的 `ReDim` 语句都不能改变数组的数据类型

4. 下面有关数组处理的叙述中，错误的是（）

①在过程中用 `ReDim` 语句定义的动态数组，其下标的上下界可以是赋值的变量

- ②在过程中, 可以使用 Dim、 Private 和 Static 语句定义数组
- ③用 ReDim 语句重新定义动态数组时, 不得改变该数组的数据类型
- ④可用 Public 语句在窗体模块的通用说明处定义一个全局数组

A). ①②③④ B). ①③④ C). ①②③ D). ②④

标准答案:d

解析: private 只能模块的通用声明字段使用, 不能在过程中使用。所以②错误。

在窗体的通过声明字段不能定义 public 类型的数组、定义字符串、常量和用户自定义类型。故④错。

5. 下列有关数组的说法中, 不正确的是()。

- A). 在窗体模块中不可以定义全局数组
- B). 可以使用 Erase 语句释放所有数组所占据的存储空间
- C). 语句 Dim X(3.6+2) As Integer 定义了一个上界是 6 的一维整型数组
- D). 形参数组的类型可以是定长字符串类型

标准答案:b

解析: 此类问题找最准确的。

1、窗体的通用声明字段不能定义全局的数组、定长字符串、常量和用户自定义类型。故 A 说有对。

2、形参一般变量不允许是定义字符串, 但形参数组可以是定长字符串。故 D 选项是正确的。

3、X(3.6+2)可以变形为 X(5.6), 继而变形为 X(6), 故 C 选项也正确。

这样也只能是 B 选项错误了, 那么错在哪里? Erase 对于动态数组而言, 收回并释放数组所占用的存储空间; 而对于常规数组而言, 只是初始化其所有元素的值为 0, 并没有释放数组所占存储空间, 所以该选项说的不准确, 错。

十七、数组的 lbound()和 Ubound()函数的使用场合。

主要是程序填空题或改错题, 用到数组的上下界时, 什么时候用函数测, 什么时候直接写具体的值。

(一) 在通用 SUB 过程或 FUNCTION 函数中, 需要使用形参数组的上下界时, 一般必须用函数得到。因为既然数组是形参, 那么编写过程或函数时也就不知道此数组到底有多少个元素, 所以不能写具体的值, 必须用函数测得到。

(二) 对于常规数组，绝对不用函数测上下界，因为常规数的上下界是确定的，测没有意义，所以填空题或改错题没有测的。

(三) 如果不是 SUB 或 FUNCTION 过程，也不是常规数组，是在一般事件过程中的动态数组，应具体问题具体分析，如果上下界已经很明确是多少了，那么一般就直接用，不用测了（当然测也没错）；如果不明确了，不知是多少了，那么就必须用函数测才可以。

举例：

程序填空题：（每空 5 分，共 60 分）

1、下面程序的功能是：通过调用排序过程，将随机产生的 10 个两位数排序，然后输出。

Option Base 1

Private Sub Command1_Click()

Randomize

Dim a(10) As Integer, i As Integer, n As Integer

For i = 1 To 10

 a(i) = Int(Rnd * 90) + 10

 Print a(i);

Next i

Print

Sort _____

For i = 1 To 10

 Print a(i);

Next i

Print

End Sub

Private Sub sort(b() As Integer)

Dim i As Integer, j As Integer, falg As Integer

For i = _____ To 1 Step -1

 falg = 0

 For j = 1 To i

 If b(j) > b(j + 1) Then

 t = b(j): b(j) = b(j + 1): b(j + 1) = t: falg = 1


```

    End If
Next j
If _____ Then Exit For
Next i

```

End Sub

2、下面程序的功能是从键盘输入长度小于 20 个字符的字符串，以“#”结束，然后对字符串从左端进行处理。处理方法：把字符串中数字内容放入数组 a 的元素中，存放时要求每个元素放一个两位数字，最后如果剩一个一位数字则单独存放。如输入的字符串为“12abc3d456e7f#”，则 12、34、56、7 分别存放到数组 a 的前四个元素中。请填空。

```

Private Sub Command1_Click()
Dim a(10) As Integer
Dim c As String
Dim w1 As String      ‘从字符串中取的一个字符
Dim w2 As Integer     ‘用来存放存入数组中的数据
Dim flag As Integer   ‘表示每次已经获取了几位数字
Dim i As Integer
i = 0: w2 = 0
c = InputBox("请输入字符串，以#结束：")
w1 = Left(c, 1)
Do While w1 <> "#"
    If w1 >= "0" And w1 <= "9" Then
        flag = flag + 1
        w2 = _____ (1) _____ + Val(w1)
    End If
    c = Mid(c, 2)
    w1 = Left(c, 1)
    If flag = 2 Then a(i) = w2: flag = 0: i = i + 1: w2 = 0
Loop
If flag > 0 Then
    A(i)=w2

```

_____(2)_____

End if

For flag = 0 To _____(3)_____

Print a(flag);

Next flag

End Sub

3、在窗体上有一个文本框和一个命令按钮，其名称分别是 `text1`、`command1`，然后编写程序。在 `text1` 文本框中输入以空格（一个空格）分隔的字符串（字符均分数字），最后无空格。请将其中的数值分解出来存入数组中，并在窗体上输出。如图，程序不完整，请填空。



Option Base 1

Private Sub Command1_Click()

Dim c() As Integer

Dim i As Integer, j As Integer, p As Integer

Dim str1 As String, str2 As String

str1 = Text1.Text

p = InStr(str1, " ")

Do

str2 = Left(str1, p - 1)

i = i + 1

ReDim Preserve c(i)

c(i) = Val(str2)

str1 = ()

p = InStr(str1, " ")

Loop ()

```

i = i + 1
ReDim Preserve c(i)
c(i) = Val(str1)
For j = 1 To ( )
    Print c(j);
Next j
End Sub

```

以上三题解析:

第一空: a 或 a(), 参数不能加括号, 不能填 (a) 或 (a()), 这里加括号是错误的, 因为调过程是没加 call, 并且给数组名加括号也不算表达式。

第二空: UBound(b) - 1

SORT 过程在编写时, 其数组 B (形参) 中有多少个元素, 根本也不知道, 所以必须用 Ubound(B)函数返回数组的上界。记住, 我们在编写过程时, 形参数组中有多少个元素谁也不知道, 只有在编写完成以后, 形实结合时, 实参确定了, 才知道, 但这时过程早已编写完毕。

还有数组做形参必须是动态数组, 但实参可以是动态数组, 也可以是常规数组。

第三空: flag = 0

第四空: w2*10

第五空: i=i+1

这里必须有一个 i=i+1, 因为这里的数组是常规数组, 无论赋了几次值, 始终是 11 个元素 (下界从 0 开始), 下面的空不能用 Ubound 函数得到数组中有多少个已赋值的元素, 所以已赋值的元素的个数, 只能由变量 i 来提供。Do 循环完成后, 如果 flag 的值不为 0, 说明 w2 中还有一个一位的数, 需要赋给一个元素, a(i)=w2, 赋完后, a(i)正好是最后一个赋了值的元素; 但可能 Do 循环完成后, flag 的值正好为 0, 没有再赋值, 这样 a(i-1)才是最后一个赋了值的元素, 为了保证一致性, 方便完成后面第三个空, 所以这里必须有一条语句, 即: i=i+1。

第六空: i-1

这里是常规数组, 数组中有多少个已赋值的元素, 只能由 i 来提供, 不能用 Ubound 函数得到, Ubound(a)函数得到始终是 10, 因为常规数组的上界一直不变。考试时不管是填空题, 还是改错题, 绝不能在这里用 Ubound(a)函数测, 没意义。

第七空: Mid(str1, p + 1, Len(str1) - p)

找到空格的位置后，将原串变短，只取空格后面的部分，要仔细想一想截，举个例子截一下，不要多截或少截字符。

第八空: While p > 0 或 Until p=0

这个空很简单，但要注意别忘了写上 While 或 Until

第九空: I 或 UBound(c)

这里是动态数组，随着截数，随着扩充元素，所以其上界，就是最后产生的数的个数，故用函数 UBound(c)完全可以；另外变量 i 也一直记录着元素的个数，且很清晰，所以 i 是几，就有几个元素。故最后直接用 for j=1 to I,也可以，用 for j=1 to ubound (c) 也可以，这里一般用第一种就行，因为变量 i 一直记录着元素的个数，很清晰。

十八、VB 过程中的 shift 和 button 参数详解

许多对象都有鼠标按下、鼠标抬起、鼠标移动事件，该类事件，都有四个参数，第一个是 button 参数，第二个是 shift 参数，第三个是 x，第四个参数是 y,前两个参数都是整形，后两个参数都是单精度类型。

这里的 x、y 表示当前鼠标所在位置的坐标。

参数 button 和 shift 虽然是个整型变量，但都是只有最后三个二进制位有效，button 的最后一位为 1，表示按下了鼠标左键，倒数第二位为 1 表示按下了鼠标右键，倒数第三位为 1 表示按下了鼠标的中间键。Shift 参数道理一样，Shift 的最后一位为 1，表示按下 shift 键，倒数第二位为 1 表示按下了 Ctrl 键，倒数第三位为 1 表示按下了 Alt 键。

所以，button=2 表示用户只按下了鼠标右键，button=3，表示用户同时按下了鼠标左右键，则 button=7 表示用户同时按下了鼠标的三个键。

同样，Shift=1，表示用户按下了 shift 键，shift=4 表示按下了 alt 键，而 shift=7，表示同户同样按下了 shift+ctrl+alt 键。

如果其它事件中还有 button 或 shift 参数，则意义都是一样的。

举例：

1. 有一个鼠标事件过程如下，程序运行后，为了能在文本框内得到” Hello” ,应在窗体上执行()操作

```
Private Sub Form_MouseDown(button As Integer, shift As Integer, X As Single, Y As Single)
```

```
    If shift=1 And button=2 Then
```

```
Text1.Text="Hello"
```

```
End If
```

```
End Sub
```

- A). 同时按下 Shift 键与鼠标左键 B). 同时按下 Shift 键与鼠标右键
C). 同时按下 Ctrl+Alt 组合键与鼠标左键 D). 同时按下 Ctrl+Alt 组合键与鼠标右键

标准答案:b

解析: 1、Shift 参数确定是否按下了 shift、ctrl 或 alt 键, 当 shift 为 1 表示按下了 shift 键, 而 shift 为 2 表示按下了 ctrl 键, shift 为 4 表示按下了 alt 键, 若同时按下了 shift+ctrl, 则参数 shift=3

2、button 参数确定按下了鼠标的哪一个键, 按鼠标左键, button=1, 按鼠标右键 button=2, 按鼠标中间键 button=4, 同时按下左右键 button=3, 三个鼠标键都按下, 则 button=7

十九、图片框与图像框知识

(一) 相同点:

都可以通过设置对象的 picture 属性, 来显示图片。在图片框或图像框中装入图片的方法有三种, 一是在设计时, 通过属性窗口设置 picture 属性; 二是在设计时, 直接右键单击图片框或图像框对象, 点粘贴, 将剪切板中的图片粘到图片框或图像框对象上; 三是在程序运行时, 用 loadpicture 函数给图片框或图像框的 picture 属性赋值, 以装入图片。

卸载图片框或图像框上图片的方法也一样, 可以在程序运行时, 用 loadpicture 函数给图片框或图像框的 picture 属性赋值, 参数为空即可。

(二) 不同点:

1、图片框可以做为容器使用, 可以在图片框中输出文字或图形, 而图像框只能用来显示图片, 不能做容器使用, 也就不能在图像框中输出文字或图形。

2、图片框最重要的属性是 autosize(标签也有这么一个属性, 功能差不多), 而图像框最重要的属性是 Stretch。

图片框的 autosize 属性默认值为假, 为假时, 图片框不会自动改变大小, 当图片很大时, 图片框中只显示部分图片, 当图片很小时, 图片只占图片框的一个角; 当 autosize 为真时, 图片框能随图片改变大小, 即控件自动适应图片。(标签的 autosize 为真时, 也是控件适应文字, 文字多, 控件变大, 文字少, 控件变小)

注意：当图片框的 `autosize=False` 时，只有特殊的一类图形文件（图元文件(*. wmf)）能自动调整大小以适应控件大小（图片适应控件），其他类型的不可以。（已在电脑上试过）

图像框的 `Stretch` 默认值也为假，`Stretch` 为假时控件适应图片（和图片框的 `autosize` 为真时效果一样），`Stretch` 为真时，图片适应控件（正好反过来）。

如果我们做一个像集，镜框固定，希望大小不一的图片能自动缩放，正好占满整个镜框，那只能选用图像框（`image`），且将其 `Stretch` 属性设置为真。

（三）最后注意，填空题或改错，如果

举例：

1、比较图片框（`PictureBox`）和图像框（`Image`）的使用，正确的描述是（ ）。

- A. 两类控件都可以设置 `AutoSize` 属性，以保证装入的图形可以自动改变大小。
- B. 两类控件都可以设置 `Stretch` 属性，使得图形根据物件的实际大小进行拉伸调整，保证显示图形的所有部分。
- C. 当图片框(`PictureBox`)的 `AutoSize` 属性为 `False` 时，只在装入图元文件(*. wmf)时，图形才能自动调整大小以适应图片框的尺寸。
- D. 当图像框(`Image`)的 `Stretch` 属性为 `True` 时，图像框会自动改变大小以适应图形的大小。

解析：图片框只有 `AutoSize` 属性，没有 `Stretch` 属性，而图像框只有 `Stretch` 属性，没有 `AutoSize` 属性。所以 A 和 B 都不对。

图像框(`Image`)的 `Stretch` 属性为 `True` 时，图片适应控件，所以 D 说反了。

当图片框的 `autosize=False` 时，只有特殊的一类图形文件（图元文件(*. wmf)）能自动调整大小以适应控件大小。所以 C 选项正确。

2. 当 `Stretch` 属性值为 `False` 时，_____。

- A). 图片大小随图片框的大小进行调整
- B). 图片大小随影像框的大小进行调整
- C). 影像框的大小随图片大小进行调整
- D). 图片框的大小随图片大小进行调整

标准答案:c

解析：首先明确 `picture` 是图片框，而 `image` 是影像框，`Stretch` 是影像框的属性。

影像框的属性是 `Stretch`，`Stretch` 为真时，图片适应控件，而为假时，控件适应图片。

图片框的属性是 `AutoSize`，`AutoSize` 为真时，控件适应图片，而为假时，只有一种特殊的图片能够适应控件，而大部分类型的图片无变化，即图片小只占控件的一部分，图片大时，

只有一部分在控件中。

标签也有 AutoSize 属性，为真时控件适应文字，即文字多，控件大，文字少，控件小，和图片框一样。

3. 为了在运行时把 d:\pic 文件夹下的图形文件 a.jpg 装入图片框 Picture1 中，所使用的语句为（ ）。

- A). Picture1. Picture =Load("d:\pic\a.jpg")
- B). Picture1. Picture =LoadPicture("d:\pic\a.jpg")
- C). Picture1. Picture =LoadPicture(d:\pic\a.jpg)
- D). Picture1.Text=LoadPicture("d:\pic\a.jpg")

标准答案:b

解析:即可以用 LoadPicture 函数装入图片，也可能用 LoadPicture 函数卸载图片，其方法是 picture1.picture=LoadPicture()

4. 运行时，要清除图片框 P1 中的图像，应使用语句[] 。

- A). P1.Picture=LoadPicture()
- B). Picture1.Picture =LoadPicture
- C). Picture =LoadPicture(" C:\WINDOWS\Clouds.bmp ")
- D). Picture1.Picture = ""

标准答案:a

解析:LoadPicture()函数可以为图片框装入图片，也可以清除其中的图片，只要该函数的参数为空，即可清除图片。

二十、VB 窗体的 Borderstyle 和 controlBox、minbutton、maxbutton 属性分析

(一) Borderstyle 属性：许多控件都有 Borderstyle 属性，Borderstyle 本意是边框样式，控件不一样，该属性的具体含义也不太一样。

1、文本框、图片框、框架、标签、图象框这五种控件都有 Borderstyle 属性，且含义一样，这里的边框的样式，是指有立体感，还是没有立体感，属性值只有 0 和 1 两个值。Borderstyle 属性为 1，表示有立体感，如文本框、图片框、框架它们的默认 Borderstyle 属性值都是 1，所以平时感觉这几个控件有立体感。Borderstyle 属性为 0 表示没立体感，如标签、图象框它们的默认 Borderstyle 属性值都是 0，所以平时感觉这两个控件没有立体感。当然用户可以改变 Borderstyle 的值。

2、图形控件（shape）也有 Borderstyle 属性，这里的边框样式，是指图形的边框是实线、虚线、点线、还是点虚线等等，其实也就边框的线型。其默认值为 1，表示实线，如果为 0 表示没有线，其它就是虚线、点线、点虚线等等，共有 0-6 七个值。

3、窗体也有 Borderstyle 属性，含义与其它控件不太一样，这里的边框样式，是指窗体是一般的窗口，还是一个对话框，一共有 0-5 六个值。我们主要记住当 Borderstyle=2 时（默认值），此时的窗体就是我们平时见的最多的窗口，有标题栏、控制菜单（及控制菜单图标）、最大化、最小化和关闭按钮；当 Borderstyle=3 时，此时的窗口就是我们平时所说的对话框，有标题栏、控制菜单（及控制菜单图标）和关闭按钮，这里没有最大化和最小化按钮。

Borderstyle=0，以上什么都没有，连标题栏都没有。

（二）窗体的 controlBox 属性，其默认值为真，即有控制菜单及控件菜单图标(只要有控制菜单就有控制菜单图标)；如果设置 controlBox=false，则窗体没有控制菜单及控制菜单图标，同时最大化、最小化和关闭按钮也没有了（不管最大化、最小化按钮的属性为何值）。也就是说 controlBox=false，标题栏上什么也不显示了。

（三）minbutton 和 maxbutton 属性，该两个属性都是逻辑值，它们的默认值都是真。前者为真最小化按钮有效，为假不最小化按钮无效。后者为真最大化按钮有效，为假不最大化按钮无效。当两个都为真，都显示，都有效。当一个为真一个为假，都显示，为真的有效，为假的无效。当两个都为假，则都不显示。该属性与关闭按钮无关。

（四）窗体的 Borderstyle、controlBox、minbutton、maxbutton 四属性解析：

1、Borderstyle=2，controlBox=true，minbutton=true，maxbutton=true 时，就是一般的窗体口，什么都有。

2、Borderstyle=2，controlBox=true,最大化和最小化按钮一个为真，一个为假，则有控制菜单，最大化和最小化按钮都有，一个可用，一个不可用，有关闭按钮。

3、Borderstyle=2，controlBox=true,最大化和最小化都为假，则有控制菜单，没有最大化和最小化按钮，有关闭按钮。

4、Borderstyle=2，controlBox=false,标题栏上什么都没有。（minbutton、maxbutton 没作用了）

5、Borderstyle=3，controlBox=true，窗体有控制菜单，没有最大化和最小化按钮，有关闭按钮（minbutton、maxbutton 没作用）

6、Borderstyle=3，controlBox=false，标题栏上什么都没有。

总结：minbutton、maxbutton 都为假，则最大化最小化都不显示；controlBox 为 false，

不管最大化最小化属性为真还是假，标题栏上什么按钮都没有；BorderStyle=3，不管最大化最小化属性为真还是假，最大化最小化按钮都不显示。

举例：

1. BorderStyle 属性是用来表示线条的 ()

A). 颜色 B). 宽度 C). 线型 D). 长度

标准答案:c

解析：BorderStyle 其中文意思是边框的类型。对线条对象来说指的是线型。

对于窗体对象来说，BorderStyle（边框模式）属性，指的是窗口的类型，可取多个值，值为2（也是默认值）表示窗体为标准窗口；如果取值为3，则为我们平时说的对话框，只有关闭按钮，没有最大化和最小化按钮；如果取值为0，那么整个窗体连标题栏也没有。

2. 当将窗体的[]属性值设置为False时，将不显示窗体的控制菜单、最大化/最小化及关闭按钮。

A). WindowState B). MinButton

C). ControlBox D). MaxButton

标准答案:c

解析：

ControlBox 控制菜单属性，当其值为真时，窗体上有控制菜单，有关闭按钮，有最大化按钮和最小化按钮，（当然要想有最大化按钮和最小化按钮，还需要maxbutton和minbutton的属性为真，如果maxbutton和minbutton都为假，则没有最大化和最小化按钮，如果只是一个为假，那么为假的那个按钮有灰色图标，但不可用）

ControlBox 属性为假时，窗体的标题栏上，什么图标都没有。

3. 要使窗体在运行时不可改变窗体的大小并没有最大化和最小化按钮，通过设置可达到此要求的属性是 ()

A). BorderStyle B). MinButton

C). Width D). MaxButton

标准答案:a

解析：

BorderStyle（边框模式）属性值为2（也是默认值）表示窗体为标准窗口，如果取值为3，则为我们平时说的对话框，只有关闭按钮，没有最大化和最小化按钮。

注意：还有一个属性ControlBox，如果该属性为假，则关闭按钮、最大化和最小化按钮、

控制菜单图标都没有。

另外如果我们设置窗体的 MinButton 和 MaxButton 同时为假，也可以实现没有最大和最小化按钮，但这里不符合题意。（单选题）

4. 在 Visual Basic 中，ControlBox 属性只适用于窗体，当窗体的 () 属性设置为 0-None 时，则 Controlbox 属性不起作用。

- A). Borderstyle B). Enable
- C). Windowstate D). Autoredraw

标准答案:a

解析:

ControlBox 控制菜单属性，当其值为真时，窗体上有控制菜单图标，有关闭按钮，有最大化按钮和最小化按钮，（当然要想有最大化按钮和最小化按钮，还需要 maxbutton 和 minbutton 的属性为真，如果 maxbutton 和 minbutton 都为假，则没有最大化和最小化按钮，如果只是一个为假，那么为假的那个按钮有灰色图标，但不可用）

ControlBox 属性为假时，窗体的标题栏上，什么图标都没有。

BorderStyle（边框模式）属性，可取多个值，值为 2（也是默认值）表示窗体为标准窗口；如果取值为 3，则为我们平时说的对话框，只有关闭按钮，没有最大化和最小化按钮；如果取值为 0，那么整个窗体连标题栏也没有。

所以窗体的 **BorderStyle** 为 0 时，窗体连标题栏都没有了，即使是 **ControlBox** 为真，也起不了作用了。

二十一、通用转化函数的特点

通用转化函数是一组函数，函数名的组成是有规律的，第一个字符是 C，后面各种数据类型关键字的缩写，如 CInt、Cstr、CInG、Csng、Sdbl、Cbool、Cdate 等函数。其功能是将其它类型强制转换成指定类型。其中必须熟记的有两个。

1、Cstr，转化为字符串函数，和 Str 不同，Cstr 没有前导空格，而 Str 有前导空格。

2、Cint，转化为整型数，有四舍五入功能，和 int 不同。

举例:

1. 函数 CInt、int、Fix 都返回数值，以下能正确地描述他们返回值大小关系的是()

- A). CInt(-4.51)=Int(-4.51)<Fix(-4.51)
- B). Int(-4.51)<Cint(-4.51)<Fix(-4.51)

- C). CInt(-4.51) < Fix(-4.51) < Int(-451)
 D). Int(-4.51) < Fix(-4.51) = CInt(-4.51)

标准答案:a

解析:cint 和四舍五入的功能。

CInt(-4.51)=-5 Int(-4.51)=-5 Fix(-4.51)=-4

1. 下列所有表达式中, 能够将两位整数 X 的个位数与十位数对调(例如, 将 78 转换为 87)的是()

- ① Val(Right(X, 1) & Left(X, 1))
 ② Val(Right(Str(X), 1) & Left(Str(X), 1))
 ③ Val(Right(CStr(X), 1) & Left(CStr(X), 1))
 ④ Val(Mid(X, 2, 1) + Mid(X, 1, 1))

- A). ①② B). ②③ C). ②④ D). ①③④

标准答案:d

解析:举例说明: a=1234(数值)

left(a,2),在 VB 中这样用是允许的, 在使用 left 函数时, 系统会自动把 a 这个数值型变量转化为字符串"1234"(暂时的, 只是在这个表达式里变成了"1234",a 这个变量本身还是数值型的), 且自动转化成的这个串没有前导空格, 故 **left(a,2)="12"** , **mid(a,2,2)="23"** 用 **str** 转化有前导空格 用 **Cstr** 转化没有前导空格

二十二、picture 属性

具 picture 属性的控件有: 命令按钮、单选按钮、复选按钮、窗体、图片框、图像框。

1、命令按钮、单选按钮、复选按钮这三个控件, 都需要先设置 style 为图形模式(1-Graphical), 再设置 picture 属性, 才能显示图片。而窗体、图片框、图像框这三个控件直接设置即可。

2、在图片框、图像框、窗体中装入图片的方法有三种, 一是在设计时, 通过属性窗口设置 picture 属性; 二是在设计时, 直接右键单击图片框、图像框、窗体对象, 点粘贴, 将剪切板中的图片粘到图片框、图像框、窗体对象上; 三是在程序运行时, 用 loadpicture 函数给图片框、图像框、窗体的 picture 属性赋值, 以装入图片。

在命令按钮、单选按钮、复选按钮这三个控件装入图片的方法只有上面的第一种和第三种, 也就是说不能右键点按钮粘贴剪功板中的图片。

3、卸载控件上图片的方法都一样，可以在程序运行时，用 loadpicture 函数重新为 picture 属性赋值，参数为空即可。

举例：

1. 如果要在命令按钮上显示图形文件，应设置命令按钮的[]]
- A). Graphics 属性 B). Picture 属性
C). Style 属性和 Graphics 属性 D). Style 属性和 Picture 属性

标准答案:d

解析:如果要在命令按钮上显示图形文件，只设置按钮的 Picture 是没有效果的，还必须设置按钮的 Style 属性为图形模式。不仅命令按钮如此，单选和复选按钮也是如此。

二十三、各种属性的读取与设置

(一) 对象的所有属性都是可读的，都可以在程序运行时读取。

(二) 有的属性在任何时候都不能设置。如列表框和组合框的 listcount 属性。

(三) 有的属性只能在设计时设置，而不能在运行时设置，如所有对象的 name 属性。

(四) 有的属性只能地运行时设置，而不能在设计时设置(属性窗口没有此属性)。如文本框的 selstart、SelLength、SelText 属性；列表框和组合框的 listindex 属性；驱动器列表框的 drive 属性；目录列表框的 Path 属性；文件列表框的 Path 和 filename 属性等等。

(五) 大部分属性是既可以在设计时设置，也可在运行时设置。

(六) 设计时不能设置的属性，在属性窗口中不会出现。

举例：

1. 控件的属性值不可以在程序运行时动态的修改。

- A). 正确 B). 错误 C). D).

标准答案:b

解析：“控件的属性值不可以在程序运行时动态的修改”，不准确，有的属性可以改，有的不可以改。

2. 下面所列的控件中，其值既能在程序设计时设定，又能在程序运行时通过赋值改变的是()

- A). List1.ListCount B). Text1.Index
C). Dir1.Path D). Labell.Visible

标准答案:d

解析：

象这样的题，找我们最熟悉的。Label1.Visible 肯定是程序设计时能设定，程序运行时也能通过赋值改变的值。

同时也该顺便记住该题目中涉及的其它知识点。List1.ListCount 设计和运行时都不能改变其值。Text1.Index 文本框数组的索引号，设计时可以改，但程序运行时就不能再改了。Dir1.Path 程序运行时一定是能改的，但设计时属性窗口根本没有该属性。

象这样的题不能说做对了，就过去了，应该理解一下这几个属性的特点。

二十四、组合框和列表框的 text 属性分析、listindex 属性分析。

(一) 有 text 属性的控件只有三个：文本框、组合框、列表框。

(二) 组合框的 text 属性既可读又可写，即可以设计时修改，也可以在运行时修改（通过代码或交互都可以）。列表框的 text 属性也可以写，但只能在代码通过特殊的方法赋值（我们用的很少）。

(四)、列表框的 listindex 属性为-1，表示没有选中任何列表项；组合框的 listindex 属性为-1，表示其文本框中的内容不是从列表框中选择的，而是直接写进去的。列表框和组合框的 listindex 属性，可读，也可以写，代码中通过设置 listindex 属性，可以实现对列表框或组合框项目的选择。

(五) 组合框 text 属性的赋值方法：

1、在程序运行时用户直接从其文本框中输入，其 text 得以赋值，但写入的内容并没有进入其下拉列表框中，要想进入到其下拉列表框中，需要用 Combo1.additem combo1.text 方法添加。

2、在程序运行时用代码 combo1.text="文本内容"，其效果是其文本框中显示了刚才写的内容，但同样没有进入到其下拉列表框中，要想进入到其下拉列表框中，仍然需要用 Combo1.additem combo1.text 方法添加。

3、在程序运行时用代码 combo1.text=combo1.list(1)形式赋值，其中 list 后面括号里的参数必须在 0-combo1.listcount-1 范围内。

4、在程序运行时用代码 list1.text=list1.list(1)形式给列表框的 text 属性赋值也是可以的（当然参数必须在 0-list1.listcount-1 范围内），但其它形式是不允许的（list1.text="abc"错误）

(六) text 属性与 listindex 属性的联系

1、能通过设置列表框或组合框的 listindex 属性值，或以达到改变其 text 属性值的目的。

2、list1.text 和 list1.list(list1.listindex)这两个值是一样的。同样 comno1.text 和

combo1.list(combo1.listindex)的值也是一样的。

举例：

1. 将字符串“hello”添加到列表框的最后，可使用哪几种方式。

(1)List1.AddItem "hello", List1.ListCount - 1

(2)List1.AddItem "hello", List1.ListCount

(3)List1.AddItem "hello", List1.ListIndex

(4)List1.AddItem "hello"

(5)list1.list(list1.listcount)="hello"

A). (1)、(2)、(3)、(4)、(5) B). (3)、(2)、(4)、(5)

C). (2)、(3)、(4) D). (2)、(4)、(5)

标准答案:d

解析:将字符串“hello”添加到列表框的最后，有多种方式：

List1.AddItem "hello"

List1.AddItem "hello", List1.ListCount

list1.list(list1.listcount)="hello"

2、下述语句中，与 list1.list(list1.listindex)功能相同的是 ()

List1.text

List1.listcount

Text1.text

List1.listindex

标准答案:a

3.如题图所示，窗体中有一个文本框(名称为 Text1)，一个组合框(名称为 Combo1,已添加 5 个条目)，一个滚动条(名称为 Hs1,min 和 max 分别为 1 和 5)，一定时器控件(名称为 Timer1)，定时器的 Enabled 属性初始为 False、Interval 属性值为 1000(1 秒)，一个命令按钮(标题为“开始”，名称为 Command)。程序功能：

(1)“开始”命令按钮功能：启动定时器工作

(2)定时器 Timer 事件功能：一是清空文本框，二是在文本框中显示组合框中选中的数字，选中数字几，就在文本框中显示几个该数字，如选中 3,则在文本框中显示 3 个 3;三是把滚动条的 Value 值改为组合框中选定的值;四是把定时器的 Timer 事件触发时间改为与所选中的组合框的值一致的秒数。

(3)程序刚刚启动时，组合框中选中的项目为其第 1 项（内容为 1）。

请在空白位置写出正确的内容。

```
Private Sub Command1_Click()
```

```
Timer1.Enabled = True
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Text1.Text = ""
```

```
Combo1.AddItem 1
```

```
Combo1.AddItem 2
```

```
Combo1.AddItem 3
```

```
Combo1.AddItem 4
```

```
Combo1.AddItem 5
```

```
(            )
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
Dim i As Integer
```

```
Dim k As Integer
```

```
Text1.Text = ""
```

```
k=val(            )
```

```
For i = 1 To k
```

```
    Text1.Text = Text1.Text + Combo1.Text
```

```
Next i
```

```
HS1.Value = Combo1.Text
```

```
(            )=combo1.Text*1000
```

```
End Sub
```

解析：

第一空： ‘Combo1.ListIndex = 0’ 或 combo1.text=1 或 combo1.text=” 1”

解析：通过设置 combo1.listindex=0 和通过设置 combo1.text=1,实现的功能看起来是一样的，都是组合框的文本框中显示 1，但其实他们有不同之处，前者的 combo1.listindex 属性为 0，后者 combol1.listindex 的值不等 0，而等于-1。再者通过 combo1.listindex 属性赋值，

此题值的范围只能是 0~4，而能过 `combo1.text` 赋值可以是任意值，如 `combo1.text=" abc"`，也是可以的。

第二空：`k = Val(Combo1.Text)` 或 `k=val(combo1.list(combo1.listindex))`

第三空：`Timer1.Interval = Combo1.Text * 1000`

二十五、向列表框或组合框添加记录的方法

(一) 从列表框(组合框)中移除项目用 `removeitem`，后面跟的是所移除的项目的序号，只有一个参数

(二) 向列表框(组合框)中添加项目用 `additem`，后面紧跟的是添加的内容。可以有两个参数，第一个参数是添加的内容，第二个参数是添加后的位置，位置参数(第二个参数)常常省略，省略了位置参数，添加到列表框(组合框)的最后。

(三) 向列表框(组合框)添加最后一条项目的方法

1、`list1.additem "项目"` 这里是用列表框的 `additem` 方法添加

2、`list1.list(list1.listcount)="项目"` 这里实质上是给列表框的属性赋值。

当然如果我们知道列表框中共有 0-5 六条记录，也可用 `list1.list(6)="项目"`，从而实现在最后添加了一条记录，但 `list1.list(7)="项目"` 这是错误的，系统会报错，还有 `list1.list(3)="项目"`，这样不是添加记录，而是修改序号为 3 的记录内容。

3、`list1.Additem "项目", list1.listcount` 这样也可以，相当于给添加的项目指定了位置，正好是最后。当然 `list1.Additem "项目", list1.listcount+1` 就不对了，因为没这个位置。

以上例子是列表框的例子，组合框一样。

(四) 不管用什么方法，只要产生了新记录，都可以用列表框或组合框的 `NewIndex` 属性，来返回最后产生的记录序号。

举例：

1. 设组合框 `Combo1` 中有 3 个项目，则下列能删除最后一项的语句是()

- A). `Combo1.RemoveItem Text`
- B). `Combo1.RemoveItem 2`
- C). `Combo1. RemoveItem 3`
- D). `Combo1.RemoceItem Combo1.Listcount`

标准答案:b

解析：

列表框和组合框的删除方法后面跟的是删除项的序号，序号从 0 开始，共有 3 项，删除最后一项，其序号是 2。

2. 将字符串“hello”添加到列表框的最后，可使用哪几种方式。

(1)List1.AddItem "hello", List1.ListCount - 1

(2)List1.AddItem "hello", List1.ListCount

(3)List1.AddItem "hello", List1.ListIndex

(4)List1.AddItem "hello"

(5)list1.list(list1.listcount)="hello"

A. (1)、(2)、(3)、(4)、(5) B. (3)、(2)、(4)、(5)

C. (2)、(3)、(4) D. (2)、(4)、(5)

标准答案:d

解析:将字符串“hello”添加到列表框的最后，有多种方式:

List1.AddItem "hello"

List1.AddItem "hello", List1.ListCount

list1.list(list1.listcount)="hello"

3. 列表框(list1)，共有 4 个数据项，若想修改最后一项的内容，将最后一项的内容由原来的“生物”，修改为“政治”，以下选项中正确的是()

A. List1.AddItem "政治"

B. List1.AddItem "政治",3

C. List1.list(3)= "政治"

D. List1.list(list1.ListCount)= "政治"

标准答案:c

解析:Additem 方法添加的是内容，Removeitem 方法删除的是项目序号，给列表框的某列表框改内容不能用 additem 方法，只能用给 list(项目序号)赋值的形式。