

附录 B 练习答案.....	3
练习 1.1.....	3
练习 2.1.....	4
练习 2.2.....	4
练习 2.3.....	5
练习 2.4.....	5
练习 3.1.....	5
练习 3.2.....	6
练习 3.3.....	7
练习 4.1.....	7
练习 5.1.....	8
练习 6.1.....	8
练习 7.1.....	9
练习 8.1.....	10
练习 8.2.....	11
练习 10.1.....	12

附录 B 练习答案

本附录包括了本书中所有练习的答案

练习 1.1

1.

MATLAB 命令窗口是 MATLAB 启动时第一个看到的窗口，用户可以在命令窗口提示符">>"后面输入命令，这些命令会被立即执行。命令窗口也可以用来执行 M 文件。编辑/调试窗口是用来新建，修改或调试 M 文件的。图像窗口用来显示 MATLAB 的图形输出。

2.

在 MATLAB 中你可以使用下列几种方式获取帮助。

- 在命令窗口中输入 `help <command_name>`，本命令将会在命令窗口中显示关于些命令的有信息。
- 在命令窗口中输入 `lookfor <keyword>`，本命令将会在命令窗口中显示所有的在第一注释行中包含该关键字的命令和函数。
- 通过在命令窗口输入 `helpwin` 或 `helpdesk` 启动帮助浏览器，或者是在启动板中选择 "Help"。帮助浏览器包含了基于超文本的 MATLAB 所有特性的描述，HTML 或 PDF 格式的在线手册，这是 MATLAB 最全面的帮助资源。

3.

工作区是命令、M 文件或函数执行时被 MATLAB 使用的变量或数组的收集器，所有命令都在命令窗口（所有的脚本文件也是从命令窗口执行）共享公共工作区，因此它们也共享所有变量，工作区的内容可以通过 `whos` 命令来查看，或者通过工作区浏览器来图形化地查看。

4.

要清除工作区的内容，只需在命令窗口中输入 `clear` 或 `clear variables` 即可。

5.

执行此操作的命令如下：

```
>> t = 5;
>> x0 = 10;
>> v0 = 15;
>> a = -9.81;
>> x = x0 + v0 * t + 1/2 * a * t^2
x = -37.6250
```

6.

执行此操作的命令如下：

```
>> x = 3;
>> y = 4;
>> res = x^2 * y^3 / (x - y)^2
res =
    576
```

问题 7 或 8 没有单一的“正确”答案。

练习 2.1

1.

数组是在内存中被组织成行和列的数据集合，只有一个名称，数据要通过在数组名后面圆括号里加上表示数据所在行和列的数字来访问。术语"向量"通常用来描述只有一维的数组，而"矩阵"通常用来描述二维或更多维的数组。

2.

(a) 这是一个 3×4 数组；

(b) $c(2,3) = -0.6$ ；

(c) 数组中值为 0.6 的元素是 $c(1,4)$ ， $c(2,1)$ 和 $c(3,2)$ 。

3.

(a) 1×3 ；(b) 3×1 ；(c) 3×3 ；(d) 3×2 ；(e) 3×3 ；(f) 4×3 ；(g) 4×1

4.

$w(2,1) = 2$

5.

$x(2,1) = -20i$

6.

$y(2,1) = 0$

7.

$v(3) = 3$

练习 2.2

1.

(a) $c(2,:) = [0.6 \ 1.1 \ -0.6 \ 3.1]$

(b) $d(:,4) = \begin{bmatrix} 0.6 \\ 3.1 \\ 0.0 \end{bmatrix}$

(c) $c(1:2,2:4) = \begin{bmatrix} -3.2 & 3.4 & 0.6 \\ 1.1 & -0.6 & 3.1 \end{bmatrix}$

(d) $c(6) = 0.6$

(e) $c(4,end) = [-3.2 \ 1.1 \ 0.6 \ 3.4 \ -0.6 \ 5.5 \ 0.6 \ 3.1 \ 0.0]$

(f) $c(1:2,2:end) = \begin{bmatrix} -3.2 & 3.4 & 0.6 \\ 1.1 & -0.6 & 3.1 \end{bmatrix}$

(g) $c([1 \ 3],2) = \begin{bmatrix} -3.2 \\ 0.6 \end{bmatrix}$

(h) $c([2 \ 2],[3 \ 3]) = \begin{bmatrix} -0.6 & -0.6 \\ -0.6 & -0.6 \end{bmatrix}$

2.

(a) $a = \begin{bmatrix} 7 & 8 & 9 \\ 4 & 5 & 6 \\ 1 & 2 & 3 \end{bmatrix}$ (b) $a = \begin{bmatrix} 4 & 5 & 6 \\ 4 & 5 & 6 \\ 4 & 5 & 6 \end{bmatrix}$ (c) $a = \begin{bmatrix} 4 & 5 & 6 \\ 4 & 5 & 6 \end{bmatrix}$

3.

$$(a) a = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix} \quad (b) a = \begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 5 \\ 0 & 0 & 6 \end{bmatrix} \quad (b) a = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 9 & 7 & 8 \end{bmatrix}$$

练习 2.3

1.

要求的命令为 “format long e”。

2.

(a) 这些语句读取用户的圆的半径，然后计算并显示圆的面积。

(b) 这些语句用整数显示 π 值，所以显示的字符串为 “The value is 3!”。

3.

第一个语句采用指数形式输出 12345.67，第二个语句采用浮点数格式输出该值，第三个语句以一般形式输出该值，第四个语句采用 12 位字符宽，并且小数点后四位的形式输出。这些语句的结果如下：

value = 1.234567e+004

value = 12345.670000

value = 12345.7

value = 12345.6700

练习 2.4

1.

(a) 操作非法。数据相乘必须有相同形式，或者数组与标量之间相乘。

(b) 合法矩阵相乘：result = $\begin{bmatrix} 4 & 4 \\ 3 & 3 \end{bmatrix}$

(c) 合法数据相乘：result = $\begin{bmatrix} 2 & 1 \\ -2 & 4 \end{bmatrix}$

(d) 操作非法。矩阵相乘 $b * c$ 产生一个 1×2 数组，而 a 是一个 2×2 数组，故相加非法。

(e) 操作非法。在两个不同大小的数组之间相乘 $b .* c$ 是非法的。

2.

结果可通过 $x = A/B$ 操作得到： $x = \begin{bmatrix} -0.5 \\ 1.0 \\ -0.5 \end{bmatrix}$

练习 3.1

1. $a > b$ 1

2. $b > d$ 0

3. $a > b \& c > d$ 0

4. $a == b$ 0

5. $a \& b > c$ 0

6. $\sim b$	1
7. $a \& b > c$	$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$
8. $a > c \& b > c$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
9. 非法。因为两个数组大小不一样。	
10. $a*b^2 > a*c$	0
11. $d \mid b > a$	1
12. $(d \mid b) > a$	0
13. $\text{isinf}(a/b)$	0
14. $\text{isinf}(a/c)$	1
15. $a > b \& \text{ischar}(d)$	1
16. $\text{isempty}(c)$	0

练习 3.2

- if $x \geq 0$
 $\text{sqrt_x} = \text{sqrt}(x);$
else
 $\text{disp}(\text{'ERROR: } x < 0\text{'});$
end
- if $\text{abs}(\text{denominator}) < 1.0\text{E-}300$
 $\text{disp}(\text{'Divide by 0 error.'});$
else
 $\text{fun} = \text{numerator} / \text{denominator};$
 $\text{disp}(\text{fun});$
end
- if $\text{distance} \leq 100$
 $\text{cost} = 0.50 * \text{distance};$
elseif $\text{distance} \leq 300$
 $\text{cost} = 50 + 0.30 * (\text{distance} - 100);$
else
 $\text{cost} = 110 + 0.20 * (\text{distance} - 300);$
end

4.
这些语句不对。要正常工作，第二个 if 语句必须是 elseif。

5.
这些语句合法，它们将显示 “Prepare to stop.”。

6.
这些语句能够执行，但它们不会按程序员的意愿工作。如果温度为 150，这些语句将输出 “Human body temperature exceeded”，而不是 “Boiling point of water exceeded”，这是因为 if 结构执行了第一个为 true 的语句而跳过其它的。要正确工作，这些测试条件的顺序要反过来。

练习 3.3

1.

```
x = 0:pi/10:2*pi;
```

```
x1 = cos(2*x);
```

```
y1 = sin(x);
```

```
plot(x1,y1,'-ro','LineWidth',2.0,'MarkerSize',6,'MarkerEdgeColor','b','MarkerFaceColor','b')
```

3.

```
'\itfrm(\itx\rm) = sin \theta cos 2\phi'
```

4.

```
'\bfPlot of \Sigma \itx\rm\bf^{2} versus \itx'
```

5.

这个字符串创建字符: τ_m

6.

这个字符串创建字符: $x_1^2 + x_2^2(\text{units:m}^2)$

7.

显示反斜线采用双反斜线('\\')

练习 4.1

1.

4 次

2.

0 次

3.

1 次

4.

2 次

5.

2 次

6.

```
ires = 10
```

7.

```
ires = 55
```

8.

```
ires = 25;
```

9.

```
ires = 49;
```

10

使用循环与分支:

```
for ii = -6*pi:pi/10:6*pi
```

```
    if sin(ii) > 0
```

```
        res(ii) = sin(ii);  
    else  
        res(ii) = 0;  
    end  
使用向量代码:  
arr1 = sin(-6*pi:pi/10:6*pi);  
res = zeros(size(arr1));  
res(arr1>0) = arr1(arr1>0);
```

练习 5.1

1.
脚本文件是保存在文件中的 MATLAB 语句集合。脚本文件共享命令窗口工作区，所以任何之前运行的脚本文件中定义的变量都能被当前脚本文件使用，脚本文件定义的变量在文件执行之后还保留在工作区中。脚本文件没有输入参数，也没有输出参数，不过脚本文件之间可以通过工作区交换数据。相反，每一个 MATLAB 函数运行在它自己的独立工作区，函数通过输入参数列表获取输入数据，通过输出参数列表给调用者返回数据。
2.
help 命令显示某个函数的所有注释行，直到遇到空白注释行或执行语句为止。
3.
H1 注释行是文件中注释的第一行，这一行可以被 lookfor 命令搜索并显示。应该把该函数的用途摘要写在这一行中。
4.
在值传递机制中，每个输入参数的副本而不是参数本身从调用者传递给函数。这样设计可以避免输入数据在函数内被自由的修改，这可能并不是调用者实际上需要的，因而也为设计好程序提供保证。
5.
MATLAB 函数可以有任意数量的参数，并且并不是每次函数调用时，每个输出参数都必须具备。nargin 函数用来确定函数被调用时实际提供了多少个参数，而 nargout 函数用来确定函数被调用后实际上有多少个输出参数。
6.
函数调用不正确。调用 test1 必须提供二个输入参数。在这种情况下，变量 y 在 test1 中将没有定义，函数被忽略。
7.
函数调用正确。

练习 6.1

1.
(a) result = 1，因为比较的是数的实部。
(b) result = 0，因为两个数的绝对值是确定的。
(c) result = 25
- 2.

plot(array)函数绘制了数组的虚部和实部，实部在 x 轴上，而虚部在 y 轴上。

3.

该向量可以通过 double 函数转换。

4.

这些语句把两行连结在一起，变量 res 包含字符串 “This is a test! This line, too.”。

5.

这些语句非法——没有 strcat 函数。

6.

这些语句非法——这两个字符串必须有相同的列数，可这些字符串长度不同。

7.

这些语句合法，产生结果

```
res = [This is a text!  
      This line, too.]
```

注意现在每行有 17 个字符，第二行垫长了长度。

8.

这些语句合法，结果是 res = 1，这是由于这两个字符串前 5 个字符相同。

9.

这些语句合法，结果是 res = [4 7 13]，这是由于字母 “s” 在字符串中的那些位置出现。

10.

这些语句合法，结果是 res = 'This IS a test!'。

11.

这些语句非法——使用 strncmp 时你必须指定要比较的字符个数。

练习 7.1

1.

稀疏数组(sparse array)是一种只为数组中的非零元素分配内存的特殊类型数组，内存中存储了稀疏数组中非零元素的下标和值。相反地，不管元素的值是不是 0，都为每个元素分配内存称为全数组。稀疏数组可以通过用 full 函数转换成全数组，全数组也可以通过用 sparse 函数转换成稀疏数组。

2.

胞数组(cell array)一个“指针”数组，每个元素都可以指向任何类型的 MATLAB 数据，它们可以是胞数组中的普通数组，也可以是不同类型的数据，例如一个数字数组，一个字符串或者是另一个胞数组，一个结构等等。此外，胞数组使用花括号 {} 而不是圆括号 () 来选择与显示单元中的内容。

3.

内容索引(Content indexing)包括在单元下标前后放置 {} 和以普通方式标记单元内容，这种索引方式定义了胞中数据结构的内容。

胞索引(Cell indexing)包括在保存到胞中的数据前后放置 {} 和以普通下标记号标记的胞下标。这种索引方式创建了包含指定数据的数据结构，并把结构赋给胞。

4.

结构是每个独立元素都有一个名称的一种数据类型，结构里的每个元素称为域，一个结构中的域可以有不同类型，域的定位是通过在结构名后附上域名的，结构名与域名之间用逗号分隔开。结构可以是普通数组或普通数组中的胞数组。结构元素通过名称来定位。

5.

`varargin` 函数出现在输入参数的最后一项，它返回一个胞数组，数组包含了函数被调用时实际上指定的参数，每一个参数保存在胞数据中独立元素中。这个函数允许 MATLAB 函数支持任意数量的输入参数。

6.

(a) `a(1,1) = (3x2 double)`. 胞数据元素 `a(1,1)` 的内容是一个 `3x3` 双精度数组，这种数据结构已经显示出来。

(b) $a(1,1) = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ ，这个语句显示了保存在元素 `a(1,1)` 中数据结构的值。

(c) 这些语句非法，这是由于你不能把一个数据结构乘以一个值。

(d) 这些语句合法，这是由于你可以把一个数据结构的内容乘以一个值。结果是

$$\begin{bmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \\ 14 & 16 & 18 \end{bmatrix}$$

$$(e) a\{2,2\} = \begin{bmatrix} -4 & -3 & -2 \\ -1 & 0 & 1 \\ 2 & 3 & 4 \end{bmatrix}$$

(f) 这个语句合法。用 $\begin{bmatrix} -17 \\ 17 \end{bmatrix}$ 这样一个 `2x1` 双精度数组初始化胞数组元素 `a(2,3)`。

(g) `a{2,2}(2,2) = 0`。

7.

$$(a) b(1).a - b(2).a = \begin{bmatrix} -3 & 1 & -1 \\ -2 & 0 & -2 \\ -3 & 3 & 5 \end{bmatrix}。$$

(b) `strncmp(b(1).b,b(2).b,6) = 1`，这是由于两个结构元素的包含字符串元素的前 6 个字符相同。

(c) `mean(b(1).c) = 2`

(d) 这个语句非法，这是由于你不能把结构数组的个别元素当作数组本身。

(e) `b = 1x2 struct array with fields:`

a
b
c

(f) `b(1) =`

a:[3x3 double]
b:'Element 1'
c:[1 2 3]

练习 8.1

1.

`textread` 函数是设计用来读取格式化好的 ASCII 文件到数据列中，每一列的格式可以是不同类型。这个命令在导入由其它程序输出的表格数据时相当有用，这是因为它可以处理单一文件中的不同类型的数据。

2.

MAT 文件相对有效地使用磁盘空间，它保存了变量的全精度——没有精度损失，如保存为 ASCII 格式，则在转换过程中会有精度损失。此外，MAT 文件保留了工作区中每个变量的全部信息：包括类别，名称，以及是不是全局的。缺点是 MAT 文件为 MATLAB 专用，不能与其它程序共享数据。

3.

fopen 函数用来打开文件，fclose 函数用来关闭文件。在 PCs（不是 Unix 计算机）中，文本文件与二进制文件是不同的。如果要以文本方式打开文件，则必须在 fopen 函数中的允许（permission）参数在加上“t”。

4.

```
fid = fopen('myinput.dat','at')
```

5.

```
fid = fopen('input.dat','r');
if fid < 0;
disp('File input.dat does not exist.');
```

end

6.

这些语句不正确，它们以文本方式打开文件，但却以二进制方式读取数据。（应该用 fscanff 函数来读取文本数据，这我们会在后继章节看到。）

7.

这些语句正确。它们创建了一个 10 个元素数组 x，打开一个二进制文件 file1,把数组写入文件，然后关闭它。接着，文件又被打开，并以[2 Inf]的格式把数据读取到数组 array 中去，

最终数组的内容是 $\begin{bmatrix} 1 & 3 & 5 & 7 & 9 \\ 2 & 4 & 6 & 8 & 10 \end{bmatrix}$ 。

练习 8.2

1.

格式化 I/O 操作产生格式化文件，一个格式化文件以 ASCII 文本保存可读的字符、数字等等。格式化文件的优点是所保存的数据可以直到看到，并且在不同程序之间方便地交换数据。然而，格式化 I/O 操作花费更多的时间读出和写入，并且格式化文件占用了更多的磁盘空间。非格式化 I/O 操作把内存中的信息不需要经过转换直接复制到磁盘，由于没有转换的过程，所以这些操作比格式化 I/O 操作要快得多。此外，相同的数据占用的磁盘空间也少得多。然而，非格式化数据被人直接检查与翻译。

2.

当要在 MATLAB 与其它程序之间交换数据，或者人们要检查与修改文件中的数据时，就应该使用格式化 I/O 操作，否则，就应该使用非格式化 I/O 操作。

3.

```
fprintf('          Table of Cosines and Sines\n\n');
fprintf('  theta          cos(theta)   sin(theta)\n');
fprintf('  =====          =====          =====\n');
for ii = 0:0.1:1
    theta = pi * ii;
    fprintf('%7.4f      %11.5f      %11.5f\n', theta, cos(theta), sin(theta));
end
```

4.

这些语句不正确。变量 **a** 和 **b** 可以被正确打印出来，但是 **%d** 格式描述符将没办法正确显示字符串。由于 **%d** 的作用，字符串的第一个字符将显示成十进制数字，而其余的字符在 **%g** 的作用下将被正确显示出来。

5.

从技术上讲，这些语句正确，但结果却不是所预想的。这些语句确实可以在一个文件中混合保存二进制数据和格式化数据，但这之后不管出于何种目的使用这个文件将变得非常困难。通常应该把二进制数据与格式化数据分开放到独立文件中。

练习 10.1

下表列出了在本章中讨论到的图形组件类型及它们的用途。

表 B-1 第十章中讨论到的 GUI 组件

组件	创建者	描述
图形控件		
按钮 Pushbutton	uicontrol	创建一个按钮的图形组件，用鼠标单击时触发回调信号。
开关按钮 Toggle Button	uicontrol	创建一个开关按钮的图形组件，开关按钮要么是“开（on）”要么是“关（off）”，每次单击时都改变状态，每次鼠标单击时都触发回调信号。
单选按钮 Radio Button	uicontrol	单选按钮是一种开关按钮，当它被选中时圆圈内显示一个小点。单选按钮组用来实现排它性选择，每次单击鼠标时会触发回调信号。
复选按钮 Check Box	uicontrol	复选按钮是一种开关按钮，当它被选中时里面显示一个小方块。每次单击鼠标时会触发回调信号。
编辑框 Edit Box	uicontrol	编辑框用来显示文本，也允许用户修改显示的信息。用户按回车键（Enter）触发回调信号。
列表框 List Box	uicontrol	列表框用来显示一系列的文本字符串。用户通过单击或双击选择一行文本字符串。当用户选择字符串时触发回调信号。
弹出菜单 Popup Menus	uicontrol	弹出菜单在响应用户鼠标单击时显示一系列的文本字符串。当没有单击弹出菜单时，只有当前选择的字符串可见。
滑动条 Slider	uicontrol	滑动条一个通过用鼠标拖动该控件平滑、连续地调节数值的图形控件。每次滑动条改变时都触发回调信号。
静态元素		
框架 Frame	uicontrol	创建的框架是图形窗口中的一个矩形框，框架通常用来把几个控件分为一组。框架不会触发回调信号。
文本域 Text Field	uicontrol	创建一个标签，在图形窗口中的某处显示一行字符串。文本域不会触发回调信号。
菜单与轴		
菜单项 Menu Items	uimenu	创建一个菜单项。当鼠标在菜单项上释放时触发回调信号。
上下文菜单 Context Menus	uicontextmenu	创建一个上下文菜单，即当用户鼠标在某个对象上右击时在该对象上面显示的菜单。
轴 Axes	axes	创建一组新轴以便在上面显示数据。轴从来不会触发回调信号。

2.

回叫函数即是用户在某个特定的 GUI 组件上发生某个动作（如鼠标单击，键盘输入等等）时执行的函数。它们通常用来当用户在 GUI 组件上单击或打字时执行某个动作。回叫函数在 `uicontrol`，`uimenu` 或 `uicontextmenu` 等组件中用“Callback”属性指定。当创建一个新 GUI 时，`uicontrol` 的回叫函数自动被向导设置成 `xxx_Callback`。其中 `xxx` 是相应 GUI 组件的 Tag 属性的值。菜单回叫函数不会被自动创建，它们必须由程序员手动定义。

3.

创建 MATLAB GUI 的基本步骤是：

(1) 确定 GUI 需要什么样的元素，每个元素需要什么样的函数。在纸上精略画出组件的布局。

(2) 使用 MATLAB 中 `guide` 向导（GUI 开发环境）在 `figure` 图形中放置组件。图形的大小，组件的对齐及组件之间的空间都可以向导中的工具来调整。

(3) 使用 MATLAB 中的 `Property Inspector` 属性检查器（内建在向导中）给每个组件指定名称（即“tag”），并且设置它们的属性，比如颜色、要显示的文本等。

(4) 把 `figure` 图形保存到文件中。当图形被保存后，它们将在磁盘中产生文件名相同、但扩展名不同的两个文件。`.fig` 文件包含了实际创建的 GUI，`.m` 文件包含的则是图形加载的代码，也保存了每个 GUI 元素的回叫函数原型。

(5) 为每个回叫函数编写代码实现行为。

4.

`handles` 数据结构一个包含了图形中所有组件句柄的结构。每个元素都有组件的名称，也有组件句柄的值。这个结构传递给每个回叫函数，允许每个回叫函数去访问图形中的组件。

5.

应用数据通过把数据添加到 `handles` 句柄结构中来实现把数据保存到 GUI 中，当结构被修改后，要使用 `guidata` 函数来保存结构。由于 `handles` 句柄结构自动传递给每个回叫函数，添加到结构中的附加数据都可以被 GUI 中的回叫函数使用。（任何一个修改 `handles` 句柄结构的函数在退出之前都必须确保已经调用了 `guidata` 函数保存修改了的 `handles` 句柄结构版本。）

6.

图形对象可以通过设置“visible”的属性值为“off”使其不可见。图形对象可以通过设置“Enable”的属性值为“off”使其对用户的鼠标单击或键盘输入没有响应。

7.

（普通）按钮、开关按钮、单选按钮、复选按钮、列表框、弹出菜单和滑动条都对鼠标单击有响应。编辑框对键盘输入有响应。

8.

对话框是一种特殊的、用来显示信息或者获取用户输入的图形。对话框通常用来显示错误、提出警告、提问问题或者获取用户输入。对话框可以由表 10.4 中的任何函数创建，这些对话框包括 `errorDlg`，`warnDlg`，`inputDlg`，`uiGetfile` 等。

9.

模式对话框不允许在它消失之前程序中的其它窗口被访问，而普通对话框不会妨碍其它窗口被访问。

10.

标准菜单是绑定到横贯图形顶部的菜单栏的，而上下文菜单可以附着到任何 GUI 组件中。当菜单栏被鼠标单击时，标准菜单即被激活，而上下文菜单必须在其所附着的 GUI 组件中右击时才能被激活。菜单从 `uimenu` 组件中创建，而上下文菜单同时从 `uimenu` 和 `uicontextmenu` 组件中创建。

11.

加速键即是那些可以通过敲击键盘从而使菜单项被选中的键。键盘记忆键是可以执行菜单项的 CTRL + key 组合。加速键与记忆键之间的主要区别是：加速键只是在菜单打开之后使菜单项被选中，而记忆键则即使菜单没有被打开，它也可以触发动作（执行菜单项）。