

应用程序与用户的接口一般是以界面形式展现的, 而用户界面一般是由窗体及其内部控件组成。因此, [界面测试](#)主要从窗体及窗体中的控件两方面来考虑。下面将详细给出窗体以及控件的测试思路, 希望能给读者在界面测试方面指引一个方向。

一个窗体一般由标题栏、菜单栏、工具栏、状态栏及内部控件组成, 因此窗体的测试主要包括窗体本身的测试以及窗体中标题栏、菜单栏、工具栏、状态栏和内部控件的测试。

## 1、窗体的测试

1) 窗体的大小。窗体的大小要合适, 使内部控件布局合理, 不过于密集, 也不过于空旷。

2) 窗体的位置。对于主窗体, 其正中应该与显示屏正中一致; 对于子窗体, 一般应在父窗体显示区的中间。

3) 移动窗体。快速或慢速移动窗体, 背景及窗体本身刷新必须正确。

4) 缩放窗体。

### ① 鼠标拖动

对于固定大小的窗体, 鼠标拖动不能缩放其大小。

对于能用鼠标拖动缩放大小的窗体, 放大或缩小窗体后其内容也应做相应调整。

② 单击‘最大化’按钮。窗体被最大化, 内部控件大小或位置也应做相应调整。

③ 单击‘还原’按钮。应还原到窗体最初默认的大小。

④ 单击‘最小化’按钮。对于主窗体, 应最小化到系统状态栏的左下角, 并依次排列; 对于窗体中的子窗体, 应最小化到父窗体容器的左下角, 并依次排列。

5) 显示分辨率。通常情况下, 计算机的显示分辨率包括  $800 \times 600$ 、 $1024 \times 768$ 、 $1280 \times 1024$  等等。

**【注意】**由于程序员在编程时, 可能使用了固定的控件大小和位置, 不能随分辨率的改变而变化, 因此, 在分辨率为  $1024 \times 768$  下开发的程序在分辨率为  $800 \times 600$  时, 会出现显示内容被裁切的情况。

6) 宽屏和普屏。宽屏和普屏的显示器, 界面显示效果可能不一样。

## 2、标题栏的测试

### 1) 标题图标, 不同窗体的图标要易于分辨;

- ① 父窗体的标题图标;
- ② 子窗体的标题图标;
- ③ 提示信息窗体的标题图标;
- ④ 警告信息窗体的标题图标;
- ⑤ 错误信息窗体的标题图标;

### 2) 标题内容, 标题的内容要简明扼要, 且不能有错别字。

- ① 父窗体的标题内容;
- ② 子窗体的标题内容;
- ③ 提示信息窗体的标题内容;
- ④ 警告信息窗体的标题内容;
- ⑤ 错误信息窗体的标题内容;

## 3、菜单栏的测试

### 1) 菜单深度最好不超过 3 层;

### 2) 菜单通常使用 5 号字体。

### 3) 菜单前的图标不宜太大, 与字高保持一致最好。

### 4) 各项菜单是否能完成相应功能?

### 5) 各菜单与其完成的功能是否一致?

### 6) 有无错别字?

### 7) 有无中英文混合?

### 8) 快捷键或热键

- ① 是否有效?
- ② 是否重复?

9) 鼠标右键菜单;

10) 不可用菜单是否真的不可用? (这在不同权限下会出现。)

#### 4、工具栏的测试

1) 工具栏中通常使用 5 号字体, 工具栏一般比菜单栏略宽。

2) 相近功能的工具栏放在一起。

3) 工具栏的按钮要有即时提示信息, 图标要能直观的表达要完成的操作。

4) 一条工具栏的长度最长不能超过屏幕宽度。

5) 系统常用的工具栏设置默认放置位置。

6) 工具栏太多时可以考虑使用工具箱, 由用户根据自己的需求定制。

#### 5、状态栏的测试

1) 显示用户切实需要的信息, 如:

① 目前的操作

② 系统的状态

③ 当前位置

④ 时间

⑤ 用户信息

⑥ 提示信息

⑦ 错误信息

⑧ 如果某一操作需要的时间比较长, 还应该显示进度条和进程提示。

2) 状态条的高度以放置 5 号字为宜。

#### 6、控件的[测试](#)

##### 6.1 控件自身的测试

1 控件本身的大小

2 控件本身的位置

### 3 控件字体

3.1 字体的大小（与界面[其它](#)控件保持一致性）

3.2 字体半角/全角

3.3 中英文混合（一般情况下不允许中英文混合）

3.4 错别字

### 6.2 控件的[功能测试](#)

#### 6.2.1 文本框

1 作用：接受用户输入的数据或显示数据。

2 状态：可编辑（正在编辑、未编辑）、不可编辑。

3 测试点：

3.1 根据文本框作用：

3.1.1 接受输入数据

● 输入数据的内容（如输入空格或与已存在内容相冲突的数据等）

● 输入数据的长度（如只能输入8位，分别输入7、8、9位数据进行测试）

● 输入数据的类型（如只能输入数字，分别输入汉字、字母、特殊符号等）

● 输入数据的格式（如‘yyyy/mm/dd’）

3.1.2 显示数据

● 显示内容是否正确？

● 内容太长，文本框不能完全显示时，是否有未完全显示的提示？如加‘...’

● 显示内容格式是否正确？

3.2 根据文本框状态：

3.2.1 可编辑文本框与不可编辑文本框是否易于区分?  
(一般将不可编辑文本框置灰)

3.2.2 光标选中的可编辑文本框是否有明显显示?(如文本框底色由白色变为蓝色)

【注意】对于在文本框中输入的错误数据,程序一般有以下3种处理方式:

- 不允许输入,没有任何提示。
- 输入后立即给出提示要求重新输入。
- 单击窗体中的‘确定’或‘保存’或‘提交’按钮以后,程序再检验数据的正确性,不正确就给出提示要求重新输入。

在设计文档中没有特别注明需采用哪种处理方式时,无论哪种方式,只要能正确验证数据就可以。

4 举例说明:略

#### 6.2.2 Up-down 控件文本框

1 作用:通过控件的上下箭头,选择不同的值。

2 状态:可用、不可用。

3 测试点:

3.1 直接输入或上下箭头选择;

3.2 边界值

3.3 默认值

3.4 输入非法数据

3.5 若该控件不可用,是否有标识?且是否真的不可用?

4 举例说明:略

#### 6.2.3 组合列表框(下拉列表框)

1 作用:下拉列表中显示一组数据,选中某一条数据,该数据就返回到框中。

2 状态：可用、不可用。

3 测试点：

3.1 条目内容是否正确？（根据需求说明书确定其内容）

3.2 条目功能是否实现？（有些程序要求在获得条目内容的同时，获得该条目对应的编号，但是编号在窗体上不显示，此时就要在数据库中查看结果是否正确？）

3.3 是否能输入数据？（一般程序不允许输入数据。）

3.4 若该控件不可用，是否有标识？且是否真的不可用？

4 举例说明：略

#### 6.2.4 列表框

1 作用：列表框中显示一组数据，选中某一条/或某几条数据，程序进行某种处理。

2 状态：可用、不可用。

3 测试点：

3.1 条目内容是否正确？（根据需求说明书确定其内容）

3.2 条目功能是否实现？

3.3 滚动条是否可以滚动？（针对列表框内容较多时）

3.4 条目内容宽度超过列表框的宽度时，鼠标指针位于该条目时是否可以完整显示？

3.5 是否允许多选？（若允许，要分别检查按 Shift 选中、按 Ctrl 选中条目和直接用鼠标选中多项条目时的情况。）

3.6 若该控件不可用，是否有标识？且是否真的不可用？

4 举例说明：略

#### 6.2.5 命令按钮

1 作用：实现规定的功能。

2 状态：可用、不可用。

### 3 测试点：

- 3.1 可操作按钮功能是否实现？
- 3.2 对可能造成数据无法恢复的操作是否提供确认信息？（如删除等操作）
- 3.3 对不符合业务要求的输入数据是否有相应的处理方法？
- 3.4 对非法的输入或操作是否给出足够的提示说明，让用户明白错误出处？
- 3.5 若该按钮不可用，是否有标识？且是否真的不可用？

### 4 举例说明：略

#### 6.2.6 单选按钮（单选框）

- 1 作用：同一组中只能选择一个。
- 2 状态：可选（被选中、不被选中）、不可选。
- 3 测试点：
  - 3.1 同一组中，是否只能选中一个？
  - 3.2 各项功能是否能正确完成？
  - 3.3 是否有默认被选中的选项？
  - 3.4 可选和不可选项是否易于区分？（一般将不可选项置灰）
  - 3.5 不可选项是否限制不能被选中？

### 4 举例说明：

如性别组的单选按钮，可选项包括：男、女、未说明，默认为男。

#### 6.2.7 复选框

- 1 作用：可同时选中多项。
- 2 状态：可选（选中、未被选中）、不可选。
- 3 测试点：

- 3.1 是否可以同时全部选中?
- 3.2 是否可以同时部分选中?
- 3.3 是否可以都不选中?
- 3.4 各种选中情况下功能的实现?
- 3.5 是否有默认被选中的选项?
- 3.6 可选和不可选项是否易于区分? (一般将不可选项置灰)
- 3.7 不可选项是否限制不能被选中?

4 举例说明: 略。

#### 6.2.8 滚动条

1 作用: 在较多内容情况下, 可以通过拖动显示内容。

2 测试点:

- 2.1 是否能被拖动?
- 2.2 拖动滚动条时, 屏幕的刷新情况? (是否能及时刷新? 是否有乱码?)
- 2.3 拖动滚动条时, 信息的显示情况?
- 2.4 滚动条的上下按钮是否可用?
- 2.5 滚动条的大小是否会根据显示信息的长、宽度及时变换?
- 2.6 滚动条的位置是否能根据选中内容的位置及时移动?
- 2.7 是否能用鼠标滚轮控制滚动条?

3 举例说明: 略

#### 6.3 各种控件混合使用时的测试

- 1 控件间的相互作用。
- 2 Tab 键的顺序。(一般是从上到下, 从左到右。)
- 3 热键的使用。



4 Enter 键和 ESC 键的使用。

5 控件组合后功能的实现。

**【注意】**测试过程中，应遵循由简到繁的原则，先进行单个控件功能的测试，确保实现无误后，再进行多个控件的功能组合的测试。