Применение методов черного ящика для планирования тестов для ListBoxer

# Эквивалентное разбиение

Для начала нужно определиться, как всевозможные входные данные разбить на группы, каждая из которых обрабатывается аналогичным образом.

Входными данными для ListBoxer являются строки: буквенные, цифровые.

|  |
| --- |
| *Длина строки от 1 до 8 символов. Диапазон вводимых букв от « a » до «z» и от « A » до « Z ». Числа можно вводить в диапазоне от 0 до 9999.* |

Эти строки затем формируются в списки строк: буквенные, цифровые, смешанные, которые в свою очередь могут быть сохранены в виде файла на диске.

|  |
| --- |
| *В документе, описывающем ListBoxer, не указаны никакие ограничения на размер этого файла, имя, расширение и т.д.*  *Это относится к разряду ошибок «Недостаточные требования».*  *Поэтому не будем это учитывать при разбиении на классы эквивалентности.* |

Также, для списков можно выбрать фильтрацию (для каждого из режимов доступны свои фильтры).

|  |
| --- |
| *В цифровом режиме доступны следующие фильтры: «<none>», «All», «0-100», «101-200», «201-300» и «301-9999».*  *В алфавитном режиме доступны следующие фильтры: «<none>», «All», «a-m» и «n-z».*  *В смешанном режиме доступны все фильтры двух предыдущих режимов.* |

Таки образом мы получаем следующие классы эквивалентности:

1. Строки длиной от 1 до 8 символов, из диапазона a-m, A-M
2. Строки длиной от 1 до 8 символов, из диапазона n-Z, N-Z
3. Строки длиной больше 8 символов, из диапазона a-z, A-Z
4. Строки длиной от 1 до 8 символов, не из диапазона a-z, A-Z
5. Число в диапазоне от 0 до 100
6. Число в диапазоне от 101 до 200
7. Число в диапазоне от 201 до 300
8. Число в диапазоне от 301 до 9999
9. Число больше 9999
10. Число меньше 0

Теперь нужно выбрать по одному представителю каждого класса и написать с ними тесты.

1. Вводим bbbBBB
2. Вводим pppPPP
3. Вводим CCCCCCCCCC
4. Вводим ююю
5. 77
6. 123
7. 255
8. 2500
9. 5000
10. -3

Для этих входных значений расписываем тесты для добавления в список (Add to List) и для ранжирования (Range).

Готово!

# Анализ граничных значений

Для применения этого метода нам опять же понадобится разбиение на классы эквивалентности. Затем нужно проверить ListBoxer на граничных и приграничных значениях.

1. FFFFFFFF
2. FFFFFFF
3. FFFFFFFFF
4. F
5. FF
6. [пустая строка]
7. AA
8. BB
9. LL
10. MM
11. NN
12. OO
13. YY
14. ZZ
15. -1
16. 0
17. 1
18. 99
19. 100
20. 101
21. 102
22. 199
23. 200
24. 201
25. 202
26. 299
27. 300
28. 301
29. 302
30. 9998
31. 9999
32. 10000

Итого нам надо написать тесты для добавления и ранжирования для 32 входных значений.

# Тестирование таблицы решений

Составим таблицу решений для ListBoxer.

Надо выделить условия и действия системы.

Условия:

1. Выбран Alphabetic
2. Выбран Numeric
3. Список непустой
4. Строка введена

Действия:

1. Добавить в список
2. Очистить список
3. Сохранить в файл
4. Выбран Range = none
5. Выбран Range = All
6. Выбран Range = a-m
7. Выбран Range = n-z
8. Выбран Range = 0-100
9. Выбран Range = 101-200
10. Выбран Range = 201-300
11. Выбран Range = 301-9999
12. Выбран Sort Order = Ascending
13. Выбран Sort Order = Descending

Составляем таблицу решений.

Напротив каждого из условий и действий.

Условия:

|  |
| --- |
| Выбран Alphabetic |
| Выбран Numeric |
| Список непустой |
| Введена строка |

Действия:

|  |
| --- |
| Добавить в список |
| Очистить список |
| Сохраниить в файл |
| Выбран Range = none |
| Выбран Range = All |
| Выбран Range = a-m |
| Выбран Range = n-z |
| Выбран Range = 0-100 |
| Выбран Range = 101-200 |
| Выбран Range = 201-300 |
| Выбран Range = 301-9999 |
| Выбран Sort Order = Ascending |
| Выбран Sort Order = Descending |

Надо составить таблицу со всевозможными комбинациями условий (условие = 1, если оно выполняется, условие = 0, если оно не выполняется). Затем расставить 0 и 1 для каждого действия (возможно ли оно при выбранных условиях).

См. «Таблица решений» excel-файле.

Получилось 13 колонок с правилами, для которых надо составить тесты.

# Тестирование таблицы переходов

Такой способ планирования тестов больше подходит для систем, где имеются последовательности происходящих событий и условий, которые соответствуют этим событиям, а соответствующая обработка конкретной ситуации зависит от событий или условий, которые произошли раньше.

Как видно, ListBoxer – не совсем ТА программа.