Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Институт информационных технологий

Надежность программного обеспечения

Лабораторная № 4

«Тестирование программного обеспечения в SilkTest»

Вариант № 3

Выполнил: студент гр. 581061 Фут Д.С.

Проверила: Бахур Н.И.

Минск 2017

Лабораторная № 4

«Тестирование программного обеспечения в SilkTest»

Вариант № 3

**Цель работы:** Научиться работать с программным обеспечением SilkTest. Произвести тестирование прикладной программы «Калькулятор» с помощью программного обеспечения SilkTest.

**Задание:** Разработать тесты для проверки выполнения операции в двоичной системе счисления над 4-байтными операндами для калькулятора вида «Программист».

Тест-кейс №1 – Проверка выполнения операции умножения в случае переполнения:

– Запустить приложение;

– Переключиться на вид «Программист».

– Переключиться на операцию «Bin - 4 байта»;

– Проверить доступность кнопок для двоичной системы счисления (доступны кнопки: 0, 1; недоступны кнопки: 2, 3, 4, 5, 6, 7, 8, 9, А, В, С, D, Е, F);

– текущее значение в поле ввода сбросить в 0;

– получить значения координат из внешних данных (X и Y);

– переместить калькулятор по полученным координатам;

– Ввести первый операнд (1111 1111 1111 1111 1111 1111 1111 1111);

– Набрать знак операции «\*»;

– Ввести второе слагаемое (1);

– Нажать клавишу «=»;

– Сравнить результаты. **Ожидаемый результат:** 1111 1111 1111 1111 1111 1111 1111 1111.

Тест-кейс №2 – Проверка выполнения операции умножения над отрицательными числами:

– Запустить приложение;

– Переключиться на вид «Программист»;

– Переключиться на операцию «Bin - 4 байта»;

– Проверить доступность кнопок для двоичной системы счисления (доступны – кнопки: 0, 1; недоступны кнопки: 2, 3, 4, 5, 6, 7, 8, 9, А, В, С, D, Е, F);

– текущее значение в поле ввода сбросить в 0;

– получить значения координат из внешних данных (X и Y);

– переместить калькулятор по полученным координатам;

– Набрать первый операнд (1010);

– Нажать клавишу +/- для смены знака;

– Проверить перевод первого операнда в дополнительный код (1111 1111 1111 1111 1111 1111 1111 0110);

– Набрать знак операции «\*»;

– Набрать второй операнд (1011);

– Нажать клавишу +/- для смены знака;

– Проверить перевод второго операнда в дополнительный код (1111 1111 1111 1111 1111 1111 1111 0101);

– Нажать клавишу «=»;

– Сравнить результаты. **Ожидаемый результат: 1101110**.

Тест-кейс №3 – Проверка выполнения операции умножения между отрицательным и положительным числами:

– Запустить приложение;

– Переключиться на вид «Программист»;

– Переключиться на операцию «Bin - 4 байта»;

– Проверить доступность кнопок для двоичной системы счисления (доступны – кнопки: 0, 1; недоступны кнопки: 2, 3, 4, 5, 6, 7, 8, 9, А, В, С, D, Е, F);

– текущее значение в поле ввода сбросить в 0;

– получить значения координат из внешних данных (X и Y);

– переместить калькулятор по полученным координатам;

– Набрать первый операнд (1010);

– Нажать клавишу +/- для смены знака;

– Проверить перевод первого операнда в дополнительный код (1111 1111 1111 1111 1111 1111 1111 0110);

– Набрать знак операции «\*»;

– Набрать второй операнд (1111 1111 1111 1111 1111 1111 1111 1111);

– Нажать клавишу «=»;

– Сравнить результаты. **Ожидаемый результат: 1010**.

Дополнительные требования к созданию тестов

1) Разработать следующие тестовые сценарии или тестпланы (Test Plan):

– запуск только Тест-кейса №1;

– запуск только Тест-кейса №2;

– запуск только Тест-кейса №3;

– запуск Тест-кейсов №1, №2, №3.

2) Все тест-кейсы создать как Data-Driven Testcase, используя Excel-файл для выбора тестовых данных.

Формат Excel-файла:

|  |  |
| --- | --- |
| Х | Y |
| 10 | 20 |

3) Общую функциональность тест-кейсов поместить в общий AppState.

По результатам выполнения задания были написаны 3 тест-кейса, 4 тест-плана, фрейм и общий апп стейт.

Код фрейма и апп стейта:

[ ] const wMainWindow = App

[ ]

[+] window MainWin App

[ ] tag "Калькулятор"

[ ]

[ ] // Blue Express screen navigation keys

[ ] LIST OF STRING lsNavKeys = NULL

[ ]

[ ]

[+] window MainWin Калькулятор

[ ] locator "/MainWin[@caption='Калькулятор']"

[+] Menu Вид

[ ] locator "Вид"

[+] MenuItem Программист

[ ] locator "Программист"

[+] RadioList Bin

[ ] locator "Bin"

[+] RadioList X4Байта

[ ] locator "4 байта"

[+] PushButton X15

[ ] locator "[@priorlabel='15'][6]"

[+] PushButton X155

[ ] locator "[@priorlabel='15'][5]"

[+] PushButton X0

[ ] locator "[@priorlabel='0'][22]"

[+] PushButton X027

[ ] locator "[@priorlabel='0'][27]"

[+] PushButton X154

[ ] locator "[@priorlabel='15'][4]"

[+] PushButton X021

[ ] locator "[@priorlabel='0'][21]"

[+] PushButton X026

[ ] locator "[@priorlabel='0'][26]"

[+] PushButton X153

[ ] locator "[@priorlabel='15'][3]"

[+] PushButton X020

[ ] locator "[@priorlabel='0'][20]"

[+] PushButton X025

[ ] locator "[@priorlabel='0'][25]"

[+] PushButton X017

[ ] locator "[@priorlabel='0'][17]"

[+] PushButton X016

[ ] locator "[@priorlabel='0'][16]"

[+] PushButton X015

[ ] locator "[@priorlabel='0'][15]"

[+] PushButton X014

[ ] locator "[@priorlabel='0'][14]"

[+] PushButton X013

[ ] locator "[@priorlabel='0'][13]"

[+] PushButton X012

[ ] locator "[@priorlabel='0'][12]"

[+] PushButton X024

[ ] locator "[@priorlabel='0'][24]"

[+] PushButton X1

[ ] locator "[@priorlabel='1'][32]"

[+] PushButton X01

[ ] locator "[@priorlabel='0'][39]"

[+] StaticText X11111111111111111111111111111

[ ] locator "11111111111111111111111111111111"

[+] MenuItem Обычный

[ ] locator "Обычный"

[+] PushButton X02

[ ] locator "[@priorlabel='0'][30]"

[+] StaticText X11111111111111111111111111110

[ ] locator "11111111111111111111111111110110"

[+] StaticText X111111111111111111111111111101

[ ] locator "11111111111111111111111111110101"

[+] PushButton X05

[ ] locator "[@priorlabel='0'][5]"

[+] StaticText X1101110

[ ] locator "1101110"

[-] StaticText X1010

[ ] locator "1010"

[ ]

[+] appstate StartState ()

[-] recording

[ ] Калькулятор.SetActive()

[ ] Калькулятор.Вид.Click(MB\_LEFT, 25, 21)

[ ] Калькулятор.Программист.Pick()

[ ] Калькулятор.Bin.Select("Bin")

[ ] Калькулятор.X4Байта.Select("4 байта")

[-] Калькулятор.X15.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Enabled", true}

[-] Калькулятор.X155.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Enabled", true}

[-] Калькулятор.X0.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Enabled", false}

[-] Калькулятор.X027.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Enabled", false}

[-] Калькулятор.X154.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Enabled", false}

[-] Калькулятор.X021.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Enabled", false}

[-] Калькулятор.X026.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Enabled", false}

[-] Калькулятор.X153.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Enabled", false}

[-] Калькулятор.X020.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Enabled", false}

[-] Калькулятор.X025.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Enabled", false}

[-] Калькулятор.X017.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Enabled", false}

[-] Калькулятор.X016.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Enabled", false}

[-] Калькулятор.X015.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Enabled", false}

[-] Калькулятор.X014.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Enabled", false}

[-] Калькулятор.X013.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Enabled", false}

[-] Калькулятор.X012.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Enabled", false}

[ ] Калькулятор.X024.Click()

[-] if( GetTestcaseState() == TCS\_EXITING )

[ ] Калькулятор.Обычный.Pick()

[ ] Калькулятор.X024.Click()

[ ]

Тест-кейс №1:

[ ]

[+] // \*\*\* DATA DRIVEN ASSISTANT Section (!! DO NOT REMOVE !!) \*\*\*

[ ] use "datadrivetc.inc"

[ ] use "test1.t"

[ ]

[ ] // \*\*\* DSN \*\*\*

[ ] STRING gsDSNConnect = "DSN=Silk DDA Excel;DBQ=C:\study\study\_2.2\НПО\_13.02.17\_В5\npo4.xls;UID=;PWD=;"

[ ]

[ ] // \*\*\* Global record for each testcase \*\*\*

[ ]

[-] type REC\_DATALIST\_DD\_Test1 is record

[ ] REC\_Лист1\_ recЛист1\_ //Лист1$,

[ ]

[ ] // \*\*\* Global record for each Table \*\*\*

[ ]

[-] type REC\_Лист1\_ is record

[ ] INTEGER X //X,

[ ] INTEGER Y //Y,

[ ]

[ ] // \*\*\* Global record containing sample data for each table \*\*\*

[ ] // \*\*\* Used when running a testcase with 'Use Sample Data from Script' checked \*\*\*

[ ]

[-] REC\_Лист1\_ grTest\_Лист1\_ = {...}

[ ] 61 // X

[ ] 312 // Y

[ ]

[ ] // \*\*\* End of DATA DRIVEN ASSISTANT Section \*\*\*

[ ]

[-] testcase DD\_Test1 (REC\_DATALIST\_DD\_Test1 rData) appstate StartState

[ ] integer i

[-] recording

[ ] Калькулятор.SetActive()

[ ] Калькулятор.Move(rData.recЛист1\_.X, rData.recЛист1\_.Y)

[-] for (i=0;i<32;i++)

[ ] Калькулятор.X155.Click()

[ ] Калькулятор.X1.Click()

[ ] Калькулятор.X155.Click()

[ ] Калькулятор.X01.Click()

[-] Калькулятор.X11111111111111111111111111111.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Text", "11111111111111111111111111111111"}

Тест-кейс №2:

[ ]

[+] // \*\*\* DATA DRIVEN ASSISTANT Section (!! DO NOT REMOVE !!) \*\*\*

[ ] use "datadrivetc.inc"

[ ] use "test2.t"

[ ]

[ ] // \*\*\* DSN \*\*\*

[ ] STRING gsDSNConnect = "DSN=Silk DDA Excel;DBQ=C:\study\study\_2.2\НПО\_13.02.17\_В5\npo4.xls;UID=;PWD=;"

[ ]

[ ] // \*\*\* Global record for each testcase \*\*\*

[ ]

[-] type REC\_DATALIST\_DD\_test2 is record

[ ] REC\_Лист1\_ recЛист1\_ //Лист1$,

[ ]

[ ] // \*\*\* Global record for each Table \*\*\*

[ ]

[-] type REC\_Лист1\_ is record

[ ] INTEGER X //X,

[ ] INTEGER Y //Y,

[ ]

[ ] // \*\*\* Global record containing sample data for each table \*\*\*

[ ] // \*\*\* Used when running a testcase with 'Use Sample Data from Script' checked \*\*\*

[ ]

[-] REC\_Лист1\_ grTest\_Лист1\_ = {...}

[ ] 24 // X

[ ] 225 // Y

[ ]

[ ] // \*\*\* End of DATA DRIVEN ASSISTANT Section \*\*\*

[ ]

[-] testcase DD\_test2 (REC\_DATALIST\_DD\_test2 rData) appstate StartState

[-] recording

[ ] Калькулятор.SetActive()

[ ] Калькулятор.Move(rData.recЛист1\_.X, rData.recЛист1\_.Y)

[ ] Калькулятор.X155.Click()

[ ] Калькулятор.X15.Click()

[ ] Калькулятор.X155.Click()

[ ] Калькулятор.X15.Click()

[ ] Калькулятор.X02.Click()

[-] Калькулятор.X11111111111111111111111111110.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Text", "11111111111111111111111111110110"}

[ ] Калькулятор.X1.Click()

[ ] Калькулятор.X155.Click()

[ ] Калькулятор.X15.Click()

[ ] Калькулятор.X155.Click()

[ ] Калькулятор.X155.Click()

[ ] Калькулятор.X02.Click()

[-] Калькулятор.X111111111111111111111111111101.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Text", "11111111111111111111111111110101"}

[ ] Калькулятор.X05.Click()

[-] Калькулятор.X1101110.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Text", "1101110"}

Тест-кейс №3:

[ ]

[+] // \*\*\* DATA DRIVEN ASSISTANT Section (!! DO NOT REMOVE !!) \*\*\*

[ ] use "datadrivetc.inc"

[ ] use "test3.t"

[ ]

[ ] // \*\*\* DSN \*\*\*

[ ] STRING gsDSNConnect = "DSN=Silk DDA Excel;DBQ=C:\study\study\_2.2\НПО\_13.02.17\_В5\npo4.xls;UID=;PWD=;"

[ ]

[ ] // \*\*\* Global record for each testcase \*\*\*

[ ]

[-] type REC\_DATALIST\_DD\_test3 is record

[ ] REC\_Лист1\_ recЛист1\_ //Лист1$,

[ ]

[ ] // \*\*\* Global record for each Table \*\*\*

[ ]

[-] type REC\_Лист1\_ is record

[ ] INTEGER X //X,

[ ] INTEGER Y //Y,

[ ]

[ ] // \*\*\* Global record containing sample data for each table \*\*\*

[ ] // \*\*\* Used when running a testcase with 'Use Sample Data from Script' checked \*\*\*

[ ]

[-] REC\_Лист1\_ grTest\_Лист1\_ = {...}

[ ] 7 // X

[ ] 45 // Y

[ ]

[ ] // \*\*\* End of DATA DRIVEN ASSISTANT Section \*\*\*

[ ]

[-] testcase DD\_test3 (REC\_DATALIST\_DD\_test3 rData) appstate StartState

[ ] integer i

[-] recording

[ ] Калькулятор.SetActive()

[ ] Калькулятор.Move(rData.recЛист1\_.X, rData.recЛист1\_.Y)

[ ] Калькулятор.X155.Click()

[ ] Калькулятор.X15.Click()

[ ] Калькулятор.X155.Click()

[ ] Калькулятор.X15.Click()

[ ] Калькулятор.X02.Click()

[-] Калькулятор.X11111111111111111111111111110.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Text", "11111111111111111111111111110110"}

[ ] Калькулятор.X1.Click()

[-] for (i=0;i<32;i++)

[ ] Калькулятор.X155.Click()

[ ] Калькулятор.X05.Click()

[-] Калькулятор.X1010.VerifyProperties({...})

[ ] ""

[-] {...}

[ ] {"Text", "1010"}

Тест-план 1:

[ ] script: test1.g.t

[ ] ddatestcase: DD\_Test1

[ ] ddatestdata: ALL\_ROWS\_FOR\_ALL\_TABLES

Тест-план 2:

[ ] script: test2.g.t

[ ] ddatestcase: DD\_test2

[ ] ddatestdata: ALL\_ROWS\_FOR\_ALL\_TABLES

Тест-план 3:

[ ] script: test3.g.t

[ ] ddatestcase: DD\_test3

[ ] ddatestdata: ALL\_ROWS\_FOR\_ALL\_TABLES

Тест-план 4:

[ ] script: test1.g.t

[ ] ddatestcase: DD\_Test1

[ ] ddatestdata: ALL\_ROWS\_FOR\_ALL\_TABLES

[ ] script: test2.g.t

[ ] ddatestcase: DD\_Test2

[ ] ddatestdata: ALL\_ROWS\_FOR\_ALL\_TABLES

[ ] script: test3.g.t

[ ] ddatestcase: DD\_Test3

[ ] ddatestdata: ALL\_ROWS\_FOR\_ALL\_TABLES

В ходе этой работы были разработаны тесты для проверки выполнения операции в двоичной системе счисления над 4-байтными операндами для калькулятора вида «Программист». Все тесты отработали без ошибок, багов в приложении обнаружено не было.