

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Отчет по лабораторной работе № 5
по предмету

Надежность программного обеспечения на тему:

«Работа в среде автоматизированного тестирования TestComplete»

Выполнил
Студент гр. 051001

Шуляк А.В.

Проверил

Деменковец Д.В.

Минск, 2022

В ходе данной лабораторной работы было протестировано приложение для вычисления и постановки ЭЦП, ее проверки по алгоритму DSA. В качестве параметров для алгоритма были взяты значения, сохраненные в виде таблицы Excel (для каждого теста собственный лист).

Тесты

- 1) input_check (тестирование введения некорректных параметров)
 - a) TestNoParameters — тестирование постановки и проверки подписи при отсутствии параметров
 - b) TestCharParameters — тестирование постановки и проверки подписи при заполнении полей не числовыми значениями
 - c) TestPNotPrime — тестирование постановки и проверки подписи при не удовлетворении условия простоты параметра p
 - d) TestQNotPrime — тестирование постановки и проверки подписи при не удовлетворении условия простоты параметра q
 - e) TestHInterval — тестирование постановки и проверки подписи при не удовлетворении условия принадлежности h интервалу $(1, p - 1)$
 - f) TestTestQNotP_1 — тестирование постановки и проверки подписи при не удовлетворении условия делимости $(p - 1)$ на q
 - g) TestXInterval — тестирование постановки подписи при не удовлетворении условия принадлежности x интервалу $(0, q)$
 - h) TestKInterval — тестирование постановки подписи при не удовлетворении условия принадлежности k интервалу $(0, q)$
- 2) signature_check (тестирование постановки и проверки подписи)
 - a) TestEmptyFile — тестирование постановки и проверки подписи при удовлетворении параметрами всех условий в случае работы с пустым файлом
 - b) TestBSUIR — тестирование постановки и проверки подписи при удовлетворении параметрами всех условий в случае работы с файлом, содержащем текст «BSUIR»
 - c) TestWrongFormat — тестирование проверки подписи при передаче файла без подписи
 - d) TestWrongSignature — тестирование проверки подписи при некорректной подписи

Скрипты модулей

1. DDTWork (модуль для работы с DDT)

```
def get_values(sheet_name, values):
    DDT.ExcelDriver("../Resources/TestSequence.xlsx", sheet_name)
    result = {}
    for value in values:
        result[value] = DDT.CurrentDriver.Value[value]
    DDT.CloseDriver(DDT.CurrentDriver.Name)
    return result
```

2. TestSamples (модуль с шаблонами для проведения тестирования)

```
import DDTWork

def startDSA():
    TestedApps.Dsa.Run(1)

def stopDSA():
    TestedApps.Dsa.Close()

def test_parameters_sign(excel_sheet, alies, file=False):
    if file:
        res = DDTWork.get_values(excel_sheet, ["p", "q", "h", "x", "k", "file"])
    else:
        res = DDTWork.get_values(excel_sheet, ["p", "q", "h", "x", "k"])
```

```

DSA = alies
frmDSA = DSA.frmDsa
frmDSA.rbSign.Select()
tb = frmDSA.tbP
tb.Text = res["p"]
tb = frmDSA.tbQ
tb.Text = res["q"]
tb = frmDSA.tbH
tb.Text = res["h"]
tb = frmDSA.tbX
tb.Text = res["x"]
tb = frmDSA.tbK
tb.Text = res["k"]
frmDSA.gbRadiobtnSign.btnChoice.Click()
if file:
    filepath = aqFileSystem.ExpandFileName("../Resources\TestFiles\\"+res["file"])
    test_choose_file(filepath, alies)

def test_parameters_check_sign(excel_sheet, alies, file_s=False):
    if file_s:
        res = DDTWork.get_values(excel_sheet, ["p", "q", "h", "y", "file_s"])
    else:
        res = DDTWork.get_values(excel_sheet, ["p", "q", "h", "y"])

    DSA = alies
    frmDSA = DSA.frmDsa
    frmDSA.rbCheckSign.Select()
    tb = frmDSA.tbP
    tb.Text = res["p"]
    tb = frmDSA.tbQ
    tb.Text = res["q"]
    tb = frmDSA.tbH
    tb.Text = res["h"]
    tb = frmDSA.tbY
    tb.Text = res["y"]
    frmDSA.gbRadiobtnSign.btnChoice.Click()
    if file_s:
        filepath = aqFileSystem.ExpandFileName("../Resources\TestFiles\\"+res["file_s"])
        test_choose_file(filepath, alies)

def test_dlg_warning(static, static_wnd_caption, alies):
    DSA = alies
    dlg = DSA.dlg_
    code = aqObject.CheckProperty(dlg, "WndCaption", cmpEqual, "Предупреждение")
    Aliases.Dsa.dlg_.Window("Button", "OK", 1).ClickButton()

def test_choose_file(filepath, alies):
    Aliases.Dsa.Window("#32770", "Open", 1).Window("ComboBoxEx32", "", 1).Window("ComboBox", "", 1).SetText(filepath)
    Aliases.Dsa.Window("#32770", "Open", 1).Window("Button", "&Open", 1).ClickButton()

def test_sign_values(sheet_excel, alies):
    DSA = alies
    values_check = DDTWork.get_values(sheet_excel, ["y", "g", "hash", "r", "s"])
    res = aqString.Compare(DSA.frmDsa.tbY.Text, values_check["y"], True)
    if res != 0:
        Log.Error("y is incorrect")
    res = aqString.Compare(DSA.frmDsa.tbG.Text, values_check["g"], True)
    if res != 0:
        Log.Error("g is incorrect")
    res = aqString.Compare(DSA.frmDsa.tbR.Text, values_check["r"], True)
    if res != 0:
        Log.Error("r is incorrect")
    res = aqString.Compare(DSA.frmDsa.tbS.Text, values_check["s"], True)
    if res != 0:
        Log.Error("s is incorrect")
    res = aqString.Compare(DSA.wndDSA.gbSignResult.tbHM.Text, values_check["hash"], True)
    if res != 0:
        Log.Error("H(M) is incorrect")

def test_check_sign_values(sheet_excel, alies):
    DSA = alies
    values_check = DDTWork.get_values(sheet_excel, ["g", "hash", "r", "s", "w", "v"])
    res = aqString.Compare(DSA.frmDsa.tbG.Text, values_check["g"], True)
    if res != 0:

```

```

    Log.Error("g is incorrect")
res = aqString.Compare(DSA.frmDsa.tbHMC.Text, values_check["hash"], True)
if res != 0:
    Log.Error("H(M') is incorrect")
res = aqString.Compare(DSA.frmDsa.tbRC.Text, values_check["r"], True)
if res != 0:
    Log.Error("r is incorrect")
res = aqString.Compare(DSA.frmDsa.tbSC.Text, values_check["s"], True)
if res != 0:
    Log.Error("s is incorrect")
res = aqString.Compare(DSA.frmDsa.tbW.Text, values_check["w"], True)
if res != 0:
    Log.Error("w is incorrect")
res = aqString.Compare(DSA.frmDsa.tbV.Text, values_check["v"], True)
if res != 0:
    Log.Error("v is incorrect")

```

3. Tests (тестирование некорректно введенных параметров)

```

import TestSamples

def test_no_sequence_sign():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_sign("KeyNone", DSA)
    TestSamples.test_dlg_warning(DSA.dlg_.Static, "Введите", DSA)

def test_no_parameters_check_sign():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_check_sign("KeyNone", DSA)
    TestSamples.test_dlg_warning(DSA.dlg_.Static, "Введите", DSA)

def test_chars_sign():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_sign("Chars", DSA)
    TestSamples.test_dlg_warning(DSA.dlg_.Static2, "не числовое значение", DSA)

def test_chars_check_sign():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_check_sign("Chars", DSA)
    TestSamples.test_dlg_warning(DSA.dlg_.Static2, "не числовое значение", DSA)

def test_sign_p_not_prime():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_sign("p_not_prime", DSA)
    TestSamples.test_dlg_warning(DSA.dlg_.Static3, "p не простое число", DSA)

def test_check_sign_p_not_prime():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_check_sign("p_not_prime", DSA)
    TestSamples.test_dlg_warning(DSA.dlg_.Static3, "p не простое число", DSA)

def test_sign_q_not_prime():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_sign("q_not_prime", DSA)
    TestSamples.test_dlg_warning(DSA.dlg_.Static4, "q не простое число", DSA)

def test_check_sign_q_not_prime():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_check_sign("q_not_prime", DSA)
    TestSamples.test_dlg_warning(DSA.dlg_.Static4, "q не простое число", DSA)

def test_sign_q_not_p_1():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_sign("q_p-1", DSA)
    TestSamples.test_dlg_warning(DSA.dlg_.Static5, "q не является делителем (p - 1)", DSA)

def test_check_sign_q_not_p_1():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_check_sign("q_p-1", DSA)
    TestSamples.test_dlg_warning(DSA.dlg_.Static5, "q не является делителем (p - 1)", DSA)

def test_sign_h_not_interval():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_sign("h_interval", DSA)
    TestSamples.test_dlg_warning(DSA.dlg_.Static6, "h не в интервале (1, p - 1)", DSA)

def test_check_sign_h_not_interval():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_check_sign("h_interval", DSA)

```

```

    TestSamples.test_dlg_warning(DSA.dlg_.Static6, "h не в интервале (1, p - 1)", DSA)
def test_sign_x_not_interval():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_sign("x_interval", DSA)
    TestSamples.test_dlg_warning(DSA.dlg_.Static7, "x не в интервале (0, q)", DSA)
def test_sign_k_not_interval():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_sign("k_interval", DSA)
    TestSamples.test_dlg_warning(DSA.dlg_.Static8, "k не в интервале (0, q)", DSA)

```

4. TestSignature (тестирование постановки и проверки подписи)

```

import TestSamples

def test_sign_empty_file():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_sign("emptyfile", DSA, True)
    TestSamples.test_sign_values("emptyfile", DSA)
    DSA.frmDsa.gbRadiobtnSign.btnSave.Click()
    filepath = aqFileSystem.ExpandFileName("../Resources\TestFiles\empty_s.txt")
    TestSamples.test_choose_file(filepath, DSA)

def test_check_sign_empty_file():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_check_sign("emptyfile", DSA, True)
    TestSamples.test_check_sign_values("emptyfile", DSA)
    aqObject.CheckProperty(DSA.frmDsa.lblResult, "Text", cmpContains, "Подпись верна")

def test_sign_BSUIR():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_sign("BSUIR", DSA, True)
    TestSamples.test_sign_values("BSUIR", DSA)

    DSA.frmDsa.gbRadiobtnSign.btnSave.Click()
    filepath = aqFileSystem.ExpandFileName("../Resources\TestFiles\BSUIR_s.txt")
    TestSamples.test_choose_file(filepath, DSA)

def test_check_sign_BSUIR():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_check_sign("BSUIR", DSA, True)
    TestSamples.test_check_sign_values("BSUIR", DSA)
    aqObject.CheckProperty(DSA.frmDsa.lblResult, "Text", cmpContains, "Подпись верна")

def test_wrong_format_check_sign():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_check_sign("wrong_format", DSA, True)
    TestSamples.test_dlg_warning(DSA.dlg_.Static9, "Подпись не может быть проверена", DSA)

def test_wrong_signature():
    DSA = Aliases.Dsa
    TestSamples.test_parameters_check_sign("wrong_signature", DSA, True)
    aqObject.CheckProperty(DSA.frmDsa.lblResult, "Text", cmpContains, "Подпись не верна")

```

Результаты тестирования

TestComplete - D:\Homework\HPO\lab5\testcomplete\Test_DSA.pjs

File Edit View Test Debug Tools Help

New Open Recent Save Undo Redo Copy Paste Find Replace Run Stop Show Mobile Screen

Project Workspace Object Browser

Project Explorer

Test_DSA

- Execution Plan
- KeywordTests
 - input_check
 - TestChars
 - TestInterval
 - TestInterval
 - TestNoParameters
 - TestNoPrime
 - TestNoP_1
 - TestNoPrime
 - TestInterval
 - signature_check
 - TestEmptyFile
 - TestEmptyFile
 - TestWrongFormat
 - TestWrongSignature
 - HomeMapping
 - Script
 - DOTWork
 - Tests
 - TestSamples
 - TestSignature
- Project Suite Logs
 - Test_DSA Logs
 - Test_DSA 12/21/2022 1:18:33 AM

Log Items

- Test_DSA
 - StartApplication [Script...]
 - TestNoParameters [Key...]
 - TestCharParameters [Ke...]
 - TestNoPrime [Keywor...]
 - TestNoPrime [Keywor...]
 - TestInterval [Keywor...]
 - TestNoP_1 [Keywor...]
 - TestInterval [Keywor...]
 - TestInterval [Keywor...]
 - TestInterval [Keywor...]
 - TestEmptyFile [Keywor...]
 - TestEmptyFile [Keywor...]
 - TestBSUR [Keywor...]
 - TestWrongFormat [Key...]
 - TestWrongSignature [Ke...]
 - StopApplication [Script...]

Project Log

Status	No	Name	Start Time	End Time	Run Time	Details
✓	1	StartApplication [ScriptTestSamples - startDSA]	1:18:33	1:18:36	0:00:02	Details
✓	2	TestNoParameters [KeywordTests - TestNoParameters]	1:18:36	1:18:38	0:00:02	Details
✓	3	TestCharParameters [KeywordTests - TestChars]	1:18:38	1:18:40	0:00:01	Details
✓	4	TestNoPrime [KeywordTests - TestNoPrime]	1:18:40	1:18:42	0:00:01	Details
✓	5	TestInterval [KeywordTests - TestInterval]	1:18:42	1:18:44	0:00:01	Details
✓	6	TestNoP_1 [KeywordTests - TestNoP_1]	1:18:44	1:18:45	0:00:01	Details
✓	7	TestInterval [KeywordTests - TestInterval]	1:18:45	1:18:47	0:00:01	Details
✓	8	TestInterval [KeywordTests - TestInterval]	1:18:47	1:18:48	0:00:00	Details
✓	9	TestInterval [KeywordTests - TestInterval]	1:18:48	1:18:49	0:00:00	Details
✓	10	TestEmptyFile [KeywordTests - TestEmptyFile]	1:18:49	1:18:53	0:00:04	Details
✓	11	TestBSUR [KeywordTests - TestBSUR]	1:18:53	1:18:57	0:00:04	Details
✓	12	TestWrongFormat [KeywordTests - TestWrongFormat]	1:18:57	1:18:59	0:00:01	Details
✓	13	TestWrongSignature [KeywordTests - TestWrongSignature]	1:18:59	1:19:00	0:00:01	Details
✓	14	StopApplication [ScriptTestSamples - stopDSA]	1:19:00	1:19:00	0:00:00	Details

Summary Log Details

Bookmarks Search/Replace Results To Do

CAPS NUM SCL