КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ТАРАСА ШЕВЧЕНКА

Кафедра інтелектуальних та інформаційних систем

Лабораторна робота № 9

з дисципліни

“ Технологія створення програмних продуктів ”

Виконав студент

групи КН- 22

Пашковський Павло Володимирович

Київ-2020

Задача 3.1

Есть программа, которая интерпретирует три целых числа, вводимых с клавиатуры, как длины сторон треугольника и выводит сообщение, о том, какой это треугольник: равносторонний, равнобедренный или неравносторонний. Напишите на листе бумаги тесты (последовательности входных данных и ожидаемые результаты), которые, как вам кажется, будут адекватно проверять эту программу.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Критерий формирования теста | № теста | Исходные данные | | | Ожидаемый результат |
| a | b | c |
| Проверка равностороннего треугольника | 1 | 3 | 3 | 3 | Равносторонний |
| Проверка равнобедренного треугольника | 2 | 5 | 5 | 4 | Равнобедренный |
| 3 | 5 | 4 | 5 | Равнобедренный |
| 4 | 4 | 5 | 5 | Равнобедренный |
| Проверка неравностороннего треугольника | 5 | 6 | 7 | 8 | Неравносторонний |
| Любая из сторон равна 0 | 6 | 0 | 0 | 0 | Сообщение об ошибке |
| 7 | 0 | 0 | 1 | Сообщение об ошибке |
| 8 | 0 | 1 | 0 | Сообщение об ошибке |
| 9 | 1 | 0 | 0 | Сообщение об ошибке |
| 10 | 0 | 1 | 1 | Сообщение об ошибке |
| 11 | 1 | 0 | 1 | Сообщение об ошибке |
| 12 | 0 | 0 | 1 | Сообщение об ошибке |
| Любая из сторон отрицательна | 13 | -3 | -8 | -4 | Сообщение об ошибке |
| 14 | -3 | -8 | 4 | Сообщение об ошибке |
| 15 | -3 | 8 | -4 | Сообщение об ошибке |
| 16 | 3 | -8 | 4 | Сообщение об ошибке |
| 17 | -3 | 8 | 4 | Сообщение об ошибке |
| 18 | 3 | -8 | 4 | Сообщение об ошибке |
| 19 | 3 | 8 | -4 | Сообщение об ошибке |
| Проверка на существование тругольника (Сумма двух сторон треугольника должна быть больше третьей стороны) | 20 | 1 | 2 | 10 | Сообщение об ошибке |
| 21 | 5 | 10 | 5 | Сообщение об ошибке |
| 22 | 10 | 4 | 5 | Сообщение об ошибке |

Задача 3.2

В программе «Деканат» переменная, обозначающая количество студентов в группе, может принимать значения от 1 до 30. Составьте тестовый набор для этой переменной.

По методу эквивалентных разбиений формируем три класса:

1. ;
2. ;
3. .

По методу граничных значений:

1. ;
2. .

Задача 3.3

В компьютерной обучающей системе тестовые задания для контроля знаний берутся из файла типа .txt, где каждое задание занимает одну строку. Для формирования теста указывается имя файла, количество заданий в тесте и количество вариантов теста (не более 10). Варианты должны различаться не менее чем тремя заданиями. Если заданий в файле недостаточно для реализации этого требования, выдается сообщение «Недостаточно заданий», если файл не найден – «файл отсутствует». Увидеть составленный вариант теста для контроля знаний можно, нажав на соответствующую кнопку «Вариант №». Составьте тестовые наборы для проверки перечисленных функций.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Критерий формирования теста | № теста | Исходные данные | | | Ожидаемый результат |
| Имя файла | Количество заданий | Количество вариантов |
| Неправильный тип файла | 1 | Test.doc | 20 | 10 | Сообщение об ошибке |
| Недопустимое количество вариантов | 2 | Test.txt | 20 | 11 | Сообщение об ошибке |
| Большое количество совпадений заданий | 3 | Test.txt | 20 | 10 | Сообщение об ошибке |
| Малое количество заданий | 4 | Test.txt | 10 | 10 | Недостаточно заданий |
| Файл не найдено | 5 | Test.txt | 20 | 10 | Файл отсутствует |
| Количество заданий = 0 | 6 | Test.txt | 0 | 10 | Недостаточно заданий |
| Количество вариантов = 0 | 7 | Test.txt | 20 | 0 | Сообщение об ошибке |
| Верные данные | 8 | Test.txt | 20 | 10 | Показ файла |

Задача 3.4

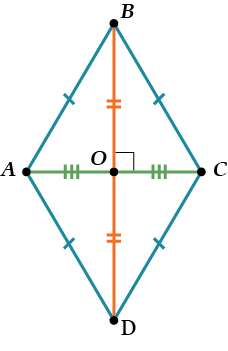
Используя методы эквивалентного разбиения и граничных значений, составьте тестовые наборы для проверки перечисленных ниже функций программы "Геометрические фигуры". Протестируйте указанную программу.

В программе "Геометрические фигуры" (файл geometry.exe) после ее запуска пользователю предоставляется выбрать вариант геометрической программы. Для выбора пользователь должен нажать на кнопку с названием нужного варианта. Предусмотрены следующие варианты:

Вариант 3. Четырехугольник задается координатами вершин. Программа проверяет, является ли он ромбом.

Для доказательства нам потребуются два правила:

1. Противоположные стороны параллельны (AB || CD и AD || BC);
2. Соседние стороны равны (AB = BC или BC = CD или CD = AD или AD = AB).



По методу эквивалентных разбиений формируем один правильный класс эквивалентности (все условия выполняются) и один неправильный (не все условия выполняются). Откуда генерируем 7 тестов:

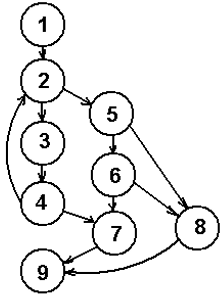
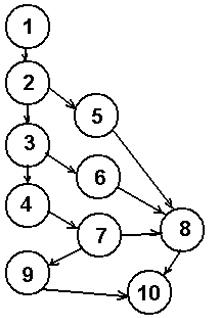
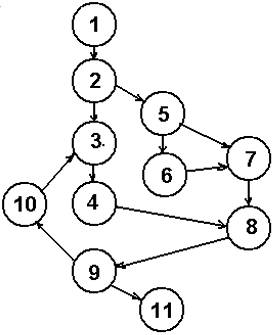
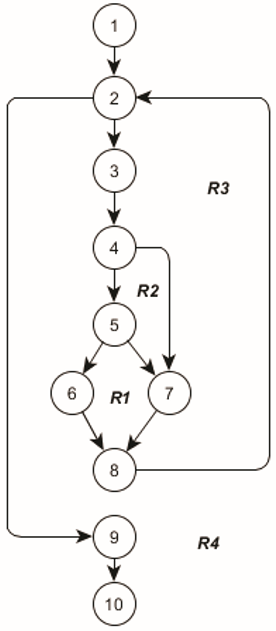
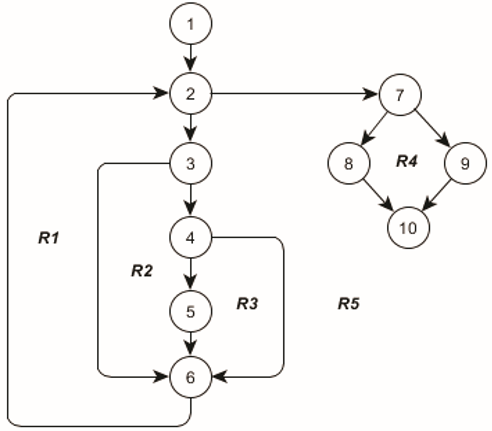
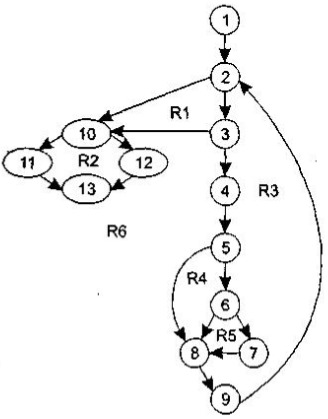
1. Условия выполняются;
2. Условия поочередно не выполняются (или же ни одно условие не выполняется).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Критерий формирования теста | № теста | Исходные данные | | | | Ожидаемый результат |
| A | B | C | D |
| Проверка на ромб | 1 | (0;2) | (4;0) | (-2;6) | (2;4) | Ромб |
| Проверка на параллелограмм | 2 | (0;2) | (4;0) | (-3;8) | (1;6) | Фигура не является ромбом |
| Проверка на четырехугольник | 3 | (-2;1) | (3;-2) | (-1;1) | (2;2) | Фигура не является ромбом |
| Совпадение координат двух или больше точек | 4 | (0;0) | (0;0) | (-2;6) | (2;4) | Фигура не является ромбом |
| 5 | (0;0) | (4;0) | (0;0) | (2;4) | Фигура не является ромбом |
| 6 | (0;0) | (4;0) | (-2;6) | (0;0) | Фигура не является ромбом |
| 7 | (0;2) | (0;0) | (0;0) | (2;4) | Фигура не является ромбом |
| 8 | (0;2) | (0;0) | (-2;6) | (0;0) | Фигура не является ромбом |
| 9 | (0;2) | (4;0) | (0;0) | (0;0) | Фигура не является ромбом |
| 10 | (0;0) | (0;0) | (0;0) | (2;4) | Фигура не является ромбом |
| 11 | (0;0) | (0;0) | (-2;6) | (0;0) | Фигура не является ромбом |
| 12 | (0;0) | (4;0) | (0;0) | (0;0) | Фигура не является ромбом |
| 13 | (0;2) | (0;0) | (0;0) | (0;0) | Фигура не является ромбом |
| 14 | (0;0) | (0;0) | (0;0) | (0;0) | Фигура не является ромбом |

Задача 5.1

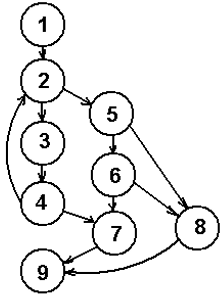
Определить цикломатическую сложность потоковых графов, представленных на рисунках ниже.

Цикломатическая сложность определяется по формуле: , где − количество дуг, − количество узлов потокового графа

1.  V(G) = 12 – 9 + 2 = 5
2.  V(G) = 12 – 10 + 2 = 4
3.  V(G) = 13 – 11 + 2 = 4
4.  V(G) = 12 – 10 + 2 = 4
5.  V(G) = 13 – 10 + 2 = 5
6.  V(G) = 17 – 13 + 2 = 6

Задача 5.2

В соответствии с концепцией максимально полного тестирования всех маршрутов программы определить независимые пути потоковых графов, представленных на рис. 1-6 с предыдущего задания.

1. 

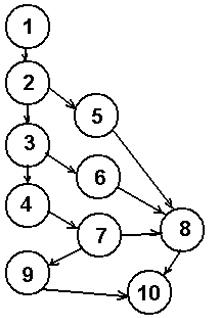
1-2-3-4-7-9

1-2-5-8-9

1-2-5-6-8-9

1-2-5-6-7-9

1-2-3-4-2-3-4-7-9

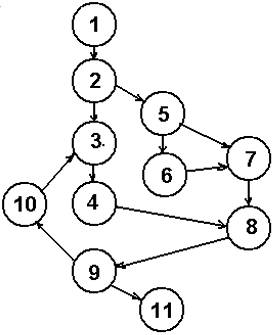
1. 

1-2-3-4-7-8-10

1-2-3-4-7-9-10

1-2-3-6-8-10

1-2-5-8-10

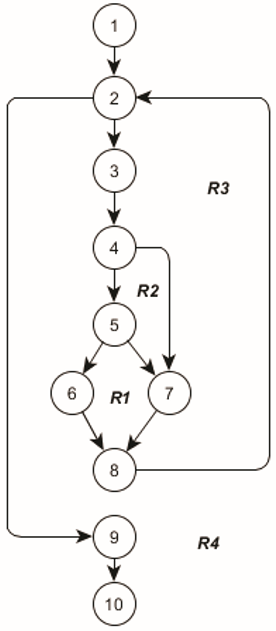
1. 

1-2-3-4-8-9-11

1-2-5-6-7-8-9-11

1-2-5-7-8-9-11

1-2-3-4-8-9-10-3-4-8-9-11

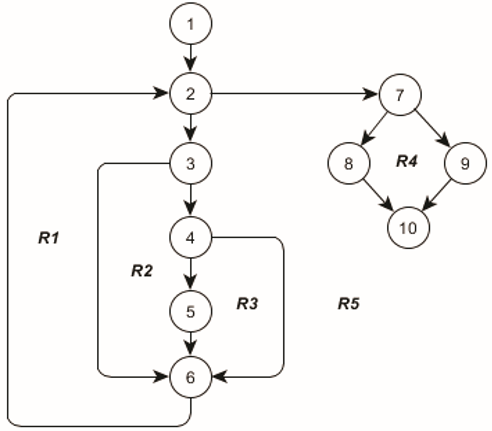
1. 

1-2-3-4-5-6-8-2-9-10

1-2-3-4-5-7-8-2-9-10

1-2-3-4-7-8-2-9-10

1-2-9-10

1. 

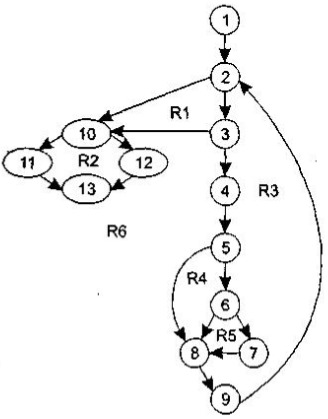
1-2-7-8-10

1-2-7-9-10

1-2-3-6-2-7-8-10

1-2-3-4-6-2-7-8-10

1-2-3-4-5-6-2-7-8-10

1. 

1-2-3-4-5-6-7-8-9-2-10-11-13

1-2-3-4-5-6-7-8-9-2-10-12-13

1-2-3-4-5-6-8-9-2-10-11-13

1-2-3-4-5-6-8-9-2-10-12-13

1-2-3-4-5-8-9-2-10-11-13

1-2-3-4-5-8-9-2-10-12-13

1-2-3-10-11-13

1-2-3-10-12-13

1-2-10-11-13

1-2-10-12-13

Задание 5.3

Для заданной процедуры составить потоковый граф, определить цикломатическую сложность потокового графа по каждой из трех формул и составить тестовые наборы по критерию покрытия маршрутов.

void procedure m (int a, int b, int &x);

{

if (a<=6) && (b<0)

{

x:=x+1;

}

if (a=7)

{

x:=x-1

}

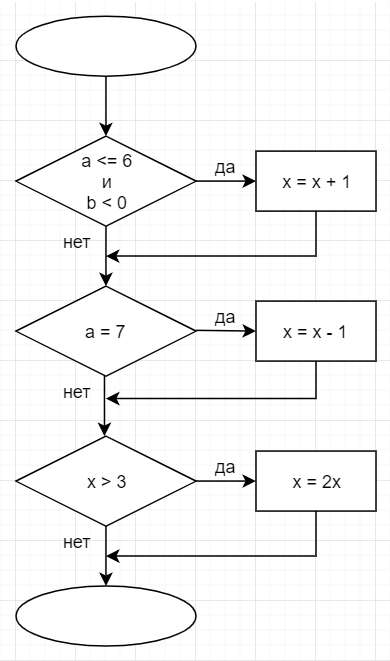
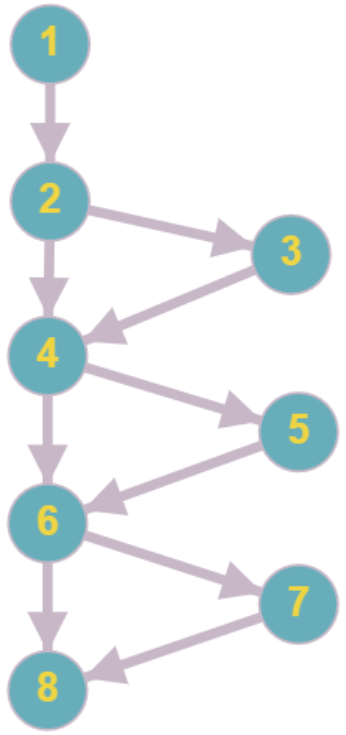
else if (x>3)

{

x=x\*2;

}

}

1. V(G) = количество регионов = 4
2. V(G) = 10 – 8 + 2 = 4
3. V(G) = 3 предикатных узла + 1 = 4

Тестовые наборы по критерию покрытия маршрутов:

a = 6, b = -1, x = 2

а = 7, b = -1, x = 2

а = 8, b = -1, х = 2

а = 8, b = -1, х = 4

Задание 5.4

Для заданной процедуры составить граф-схему и тестовые наборы для тестирования маршрутов по критериям:

1. покрытия операторов;
2. покрытия решений (переходов);
3. покрытия условий;
4. покрытия решений/условий;
5. комбинаторного покрытия условий.
6. a = 6, b = -1, х = 4

а = 7, b = -1, х = 2

а = 8, b = -1, х = 2

1. а = 6, b = -1, х = 2

а = 7, b = -1, х = 2

а = 8, b = -1, х = 2

а = 8, b = -1, х = 4

1. а = 6, b = -1, х = 2

а = 5, b = 0, х = 2

а = 7, b = -1, х = 2

а = 8, b = 0, х = 2

а = 8, b = -1, х = 4

1. а = 6, b = -1, х = 2

а = 5, b = 0, х = 2

а = 7, b = -1, х = 2

а = 8, b = 0, х = 2

а = 8, b = -1, х = 4

1. а = 6, b = -1, х = 2

а = 5, b = 0, х = 2

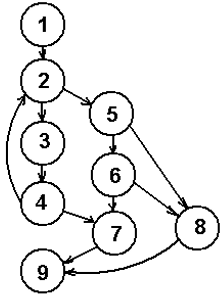
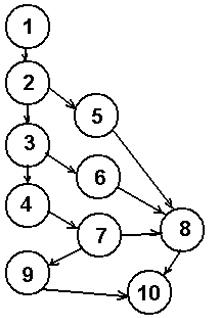
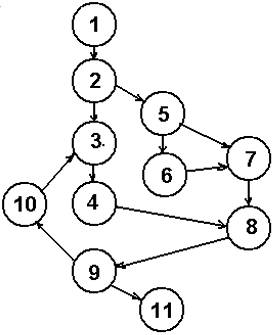
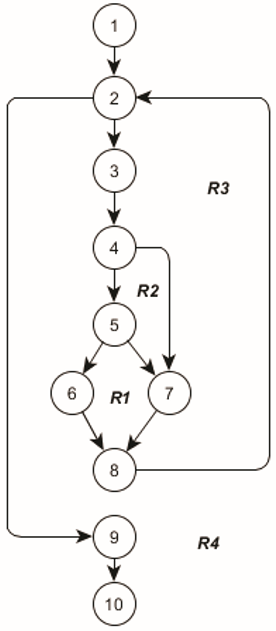
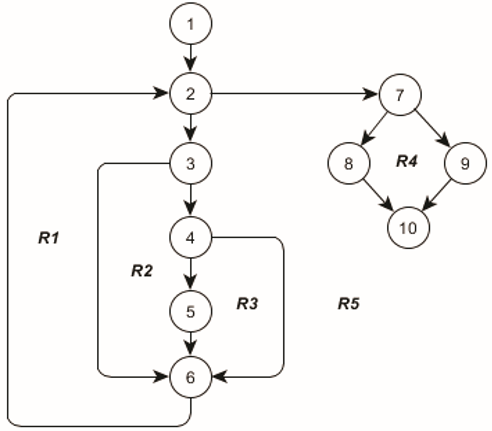
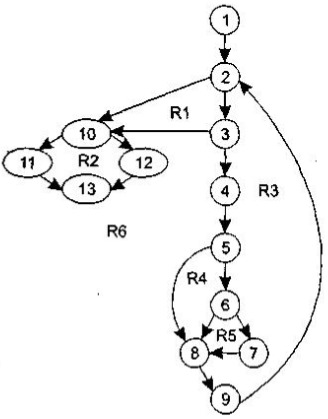
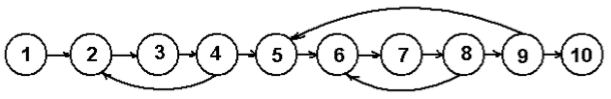
а = 7, b = -1, х = 2

а = 8, b = 0, х = 2

а = 8, b = -1, х = 4

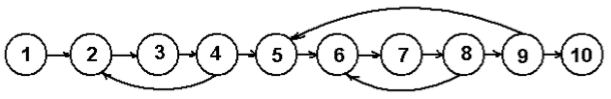
Задача 5.5

Определить типы циклов в потоковых графах, представленных на рисунках.

1.  Неструктурированный
2.  Неструктурированный
3.  Неструктурированный
4.  Вложенный
5.  Вложенный
6. Вложенный
7. Вложенный и объединенный

Задача 5.6

Сколько наборов тестов необходимо для тестирования программы, потоковый граф которой представлен на рисунке.



Выбираем самый внутренний цикл. Устанавливаем минимальные значения параметров других циклов.

Для внутреннего цикла проводим тесты простого цикла. Добавляем тесты для исключенных значений и значений, которые выходят за границы рабочего диапазона.

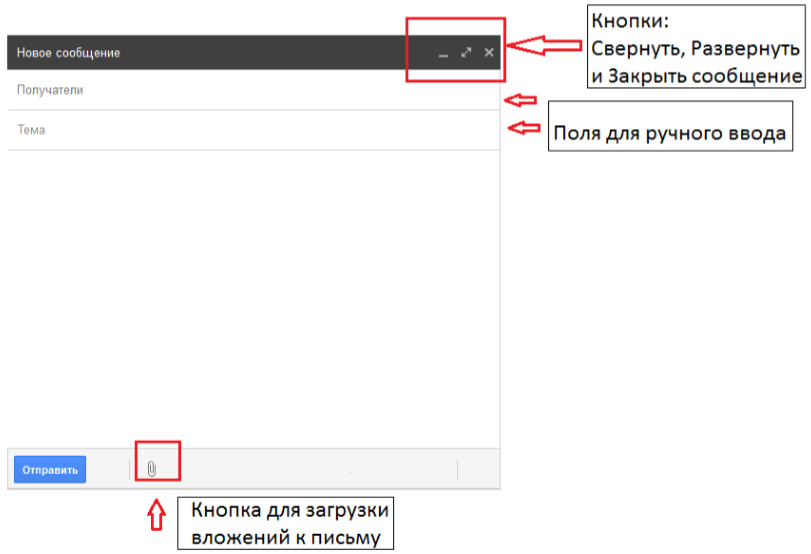
Переходим в следующий по порядку внешний цикл. Выполняем его тестирование. При этом сохраняем минимальные значения параметров внешнего цикла и типичные значения для вложенного цикла.

Для третьего независимого цикла выполняем тестирование как для простого.

Тест кейсы

Написать тест кейсы, позволяющие детально протестировать функционал (по возможности используя техники тест дизайна), соответствующие вашей теме.

Ваш проект «Реализация онлайн почтового клиента». Нужно описать тест кейсы на функционал: Отправка сообщения. Путь к форме отправки сообщения: нажатие кнопки «Новое сообщение» на панели инструментов. Дизайн с комментариями прилагается.



|  |  |  |  |
| --- | --- | --- | --- |
| № теста | № действия | Действие | Ожидаемый результат |
| 1 | 1 | Нажатие кнопки «Новое сообщение» | Открытие формы для написания и отправки сообщения  Все поля пусты |
| 2 | Заполнение полей «Получатели», «Тема», «Текст» и прикрепление файлов | Поля заполнены |
| 3 | Нажатие кнопки «Отправить» | Отправка сообщения  Вывод на экран уведомления «Письмо отправлено» |
| 2 | 1 | Нажатие кнопки «Новое сообщение» | Открытие формы для написания и отправки сообщения  Все поля пусты |
| 2 | Заполнение полей «Получатели», «Тема», «Текст» | Поля заполнены |
| 3 | Нажатие кнопки «Отправить» | Отправка сообщения  Вывод на экран уведомления «Письмо отправлено» |
| 3 | 1 | Нажатие кнопки «Новое сообщение» | Открытие формы для написания и отправки сообщения  Все поля пусты |
| 2 | Заполнение полей «Получатели», «Тема» | Не все поля заполнены |
| 3 | Нажатие кнопки «Отправить» | Отправка сообщения  Вывод на экран уведомления «Письмо отправлено» |
| 4 | 1 | Нажатие кнопки «Новое сообщение» | Открытие формы для написания и отправки сообщения  Все поля пусты |
| 2 | Заполнение полей «Получатели», «Текст» | Не все поля заполнены |
| 3 | Нажатие кнопки «Отправить» | Отправка сообщения  Вывод на экран уведомления «Письмо отправлено» |
| 5 | 1 | Нажатие кнопки «Новое сообщение» | Открытие формы для написания и отправки сообщения  Все поля пусты |
| 2 | Заполнение поля «Получатели» | Не все поля заполнены |
| 3 | Нажатие кнопки «Отправить» | Вывод на экран уведомления «Отправить это сообщение без темы и текста?»  Подтверждение отправки  Отправка сообщения  Вывод на экран уведомления «Письмо отправлено» |
| 6 | 1 | Нажатие кнопки «Новое сообщение» | Открытие формы для написания и отправки сообщения  Все поля пусты |
| 2 | Заполнение поля «Получатели» и прикрепление файлов | Не все поля заполнены |
| 3 | Нажатие кнопки «Отправить» | Вывод на экран уведомления «Отправить это сообщение без темы и текста?»  Подтверждение отправки  Отправка сообщения  Вывод на экран уведомления «Письмо отправлено» |
| 7 | 1 | Нажатие кнопки «Новое сообщение» | Открытие формы для написания и отправки сообщения  Все поля пусты |
| 2 | Заполнение поля «Текст» | Не все поля заполнены |
| 3 | Нажатие кнопки «Отправить» | Вывод на экран ошибки «Укажите как минимум одного получателя»  Сообщение не отправлено |
| 8 | 1 | Нажатие кнопки «Новое сообщение» | Открытие формы для написания и отправки сообщения  Все поля пусты |
| 2 | Заполнение полей «Получатели», «Тема», «Текст» и прикрепление файлов (превышен лимит получателей) | Поля заполнены |
| 3 | Нажатие кнопки «Отправить» | Вывод на экран уведомления «Слишком большое количество получателей»  Сообщение не отправлено |
| 9 | 1 | Нажатие кнопки «Новое сообщение» | Открытие формы для написания и отправки сообщения  Все поля пусты |
| 2 | Заполнение полей «Получатели», «Тема», «Текст» и прикрепление файлов (превышен размер текста) | Поля заполнены |
| 3 | Нажатие кнопки «Отправить» | Вывод на экран уведомления «Слишком много текста»  Сообщение не отправлено |
| 10 | 1 | Нажатие кнопки «Новое сообщение» | Открытие формы для написания и отправки сообщения  Все поля пусты |
| 2 | Заполнение полей «Получатели», «Тема», «Текст» и прикрепление файлов (превышен размер прикрепленного сообщения) | Поля заполнены |
| 3 | Нажатие кнопки «Отправить» | Вывод на экран уведомления «Слишком большой прикрепленный файл»  Сообщение не отправлено |

Вывод: В данной лабораторной работе я научилась составлять наборы тестовых данных для структурного тестирования. Освоила стратегию «черного ящика» и «белого ящика». Научилась писать тест кейсы на предоставленный с дизайнами функционал.