**Практическая работа №6**

Тема: «Разработка и тестирование программного продукта»

Задание: разработать и протестировать программный продукт.

Чек-лист:

|  |  |  |
| --- | --- | --- |
| № | Требование | Не/выполнено |
| 1 | В программе имеется графический интерфейс, с которым пользователь может взаимодействовать | + |
| 2 | Программа принимает целые числа (стороны треугольника), после чего определяет его вид | + |
| 3 | В программе работает защита от ввода нуля | + |
| 4 | В программе работает защита от ввода отрицательных значений | + |
| 5 | В программе работает защита от ввода символов | + |
| 6 | Интерфейс программы прост в обращении и корректно выполняет свои функции | + |
| 7 | При работе, программа не зависает и не вылетает | + |
| 8 | Если пользователь введет некорректное значение, то программа выдаст сообщение с конкретной причиной ошибки | + |
| 9 | Пользователь может продолжить работу в программе после определения треугольника по первым заданным параметрам | + |

Тестовый план:

|  |  |  |  |
| --- | --- | --- | --- |
| Функциональные элементы | | | |
| № | Элемент | Вид теста | Риск |
| 1 | Программа исправно определяет вид треугольника | White-box, Black-box | Риск возможен в том случае, если в алгоритме допущена неточность |
| 2 | Текст-боксы принимают в себя только целые числа | White-box, Negative | Вероятность того, что программа примет что-то кроме целых чисел – минимальна |
| 2 | Кнопка «Рассчитать» работает исправно | White-box, Black-box | Риск возможен, если в алгоритме допущена ошибка |
| 4 | Защита от ввода некорректных значений работает исправно | Negative | Риск возможен, если не предусмотрен какой-либо вариант некорректных значений |
| Нефункциональные элементы | | | |
| 1 | В программе есть интерфейс, с которым можно взаимодействовать | GUI-тестирование, юзабилити | Риск может быть связан с тем, что пользователю не понравится дизайн самого приложения |
| 2 | Ответ ясен и понятен для пользователя | GUI-тестирование, опрос | Опять же, риск может быть связан с эстетическими предпочтениями пользователя |
| 3 | Алгоритм работы с программой интуитивно понятен для пользователя | Юзабилити, опрос | Риск минимален, ведь каждый интерактивных элемент подписан и дает необходимую для работы информацию |
| 4 | Программа не зависает | юзабилити | Риск возникновения зависит от ПК, на котором приложение будет запущено |

Тестирование White-box:

|  |  |
| --- | --- |
| Тест | Результат |
| Ввод целых чисел | Успех |
| Нажатие на кнопку «Рассчитать» | Получен ответ, на основании введенных ранее чисел |
| Продолжить работу после получения ответа | Программа позволяет изменить значения на другие и повторить алгоритм |

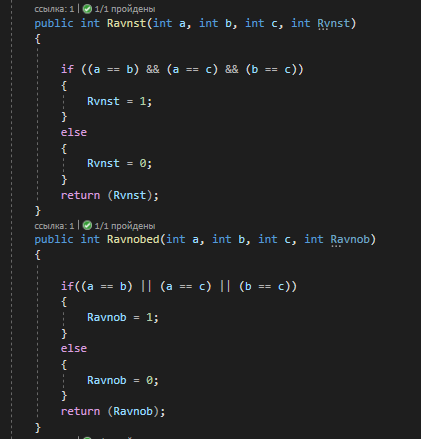
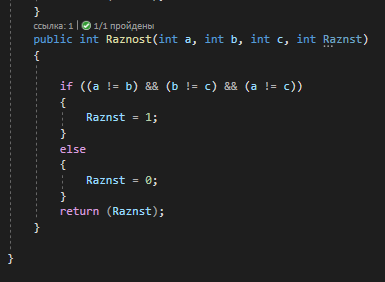
Данный вид тестирования подтверждается ранее составленным чек-листом.

Тестирование Black-box:

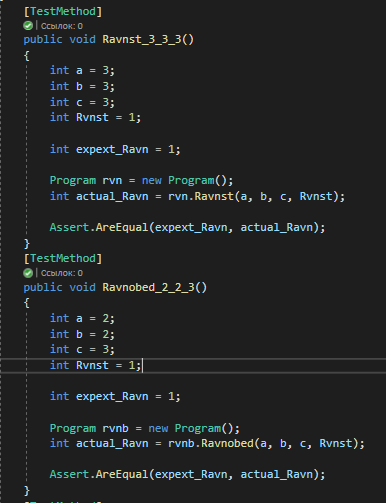
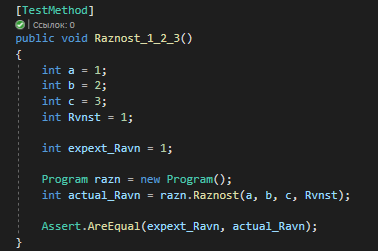
|  |  |
| --- | --- |
| Тест | Результат |
| Ввод отрицательных чисел | Отрицательные числа просто невозможно ввести |
| Ввод нулей | Программа выдаст сообщение с конкретной причиной ошибки |
| Попытка ввода некорректных значений | В программа прописана теорема сторон треугольника, что повлекло за собой вывод ошибки с конкретным сообщением |
| Нажать на кнопку «Результат» без ввода значений | Программа выдаст сообщение об ошибке |

Positive и Negative:

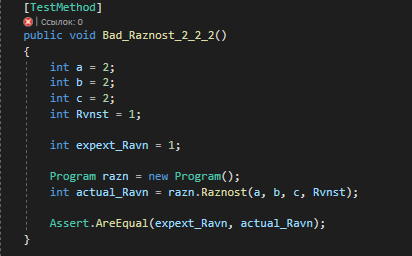
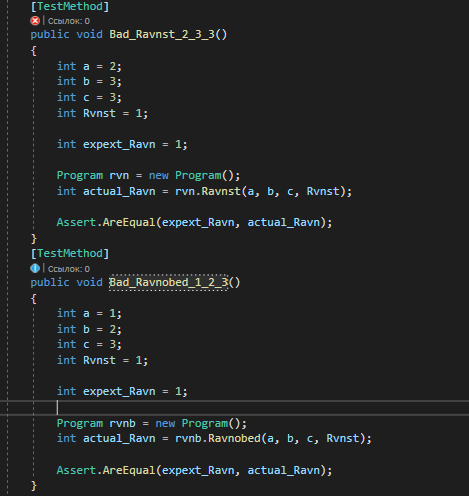
При вызове класса Form – вся логика приложения расписана в обработчики нажатия на кнопки, следовательно такой код не годится для модульного тестирования, поэтому был создан отдельный класс, где были прописаны функции, содержащие логику приложения.

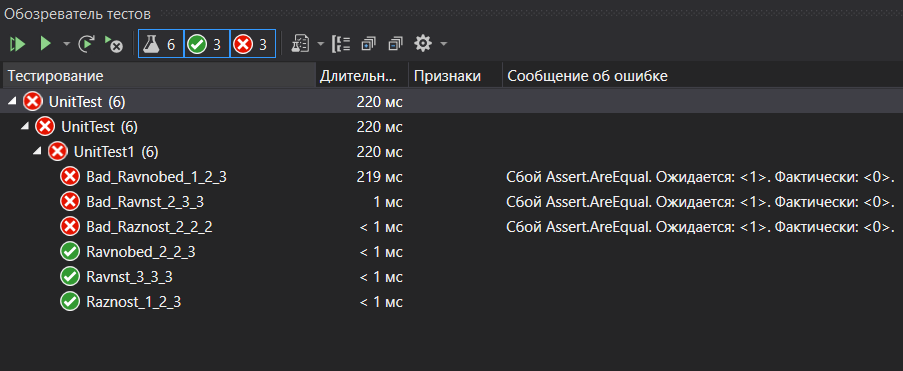
Класс программы, где содержатся функии

Тесты по позитивному сценарию



Тесты по негативному сценарию



GUI & Ux-testing:

|  |  |
| --- | --- |
| UX | |
| Тест | Результат |
| Пользователь хочет ввести значения | Успех |
| Пользователь нажимает на кнопку «Рассчитать» | Кнопка работает |
| Пользователь хочет увидеть результат работы программы | В правом верхнем углу появился ясный ответ |
| Пользователь случайно ввел неверные значения | Программа вывела сообщение об ошибке, где указана конкретная проблема |

GUI – интерфейс дружелюбен к пользователю, он содержит минимум необходимых для работы элементов, которые расположены по определенным областям окна приложения, дизайн прост и не вызывающ, а поля и кнопки подписаны для удобства пользователя.