Лабораторная работа №12

РЕГРЕССИОННОЕ ТЕСТИРОВАНИЕ

Цель: получить навыки выполнения регрессионного тестирования

Ход работы: изучить указания, выполнить типовое задание, выполнить задание для самостоятельной работы, оформить отчет.

Теория.

Под регрессионным тестированием понимают те виды тестов, которые проводятся с каждой новой версией программы.

Примеры видов тестов:

**Тест верификации багов:** Допустим, что тест с номером 3 выявил баг, что было зафиксровано и передано разработчику для исправления. Через определенное время Вы получили от разработчика новую версию программы, с информацией о том, что описнный баг испарвлен. Ваша задача - провести тест с номером 3 повторно - для того, чтобы убедиться, что баг действительно больше не проявляется. В случае успешного прохождения теста такой баг помечается как Verified, в противном случае - как re-do, о чем сообщается разработчику и передается на доработку. Проведение таких тестов является обязательным. Так как причин, из-за которых исправленный баг может сохраниться в программе - множество (от ошибочного описания, а, возможно, и понимания проблемы, до ошибочного утверждения о том, что исправление имело место).

**Тесты верификации версии:** Если хотя бы один из тестов верификации версии выявляет баг - то тестер возвращается к предыдущей (последней "рабочей"), дальнейшей тестирование новой версии не проводится, а информация об ошибке вносится в базу и отправляется разработчику. Таким образом, тесты верификации версии представляют собой краткий набор основных тестов функциональности.

**Тесты регрессии:** Допустим, что ранее проведенный тест № 2, который обеспечивал проверку в программе участка кода (назовем его условно кодом-А) не выявил ошибок в программе, и был отмечен как pass. В ходе разработки возникла необходимость изменить участок кода-А (например, при исправлении какого либо иного бага или же придания программе новой функциональности). В результате этот участок кода требует дополнительной проверки, что и будет сделано при повторном проведении теста № 2.

**Тест регрессии при закрытых багах:** Допустим, что тест № 3, выявивший баг, после исправления этого бага разработчиком был проведен повторно, при том успешно. Тест был отмечен как pass, а баг - как Verified. Такой баг откладывается "на полочку", "дело" закрыто. Такой баг и будет "закрытым". Допустим теперь, что в ходе разработки, участок кода, где был исправлен этот баг был изменен, или сменился разработчик, который случайно удалил "нашлепку" в коде исправлявшую этот глюк и показавшуюся ему лишней и т.п. В этом случае баг проявится снова. Что бы не допустить подобного бета-тестеру время от времени необходимо проводить тесты, выявлявшие ранее баги в измененном участке кода, исправление которых уже было проверено ранее и зафиксировано в базе.

Для наглядности при проведении Регрессионного тестирования можно использовать следующую таблицу:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № теста | № версии | № бага | № версии | № бага |  |  |
|  |  |  |  |  |  |  |

Количество столбцов соответствует количеству версий.

Например,

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № теста | Версия 1 | № бага | Версия 2 | № бага | Версия 3 | № бага |
| 1 | Pass |  |  |  |  |  |
| 2 | Pass |  |  |  |  |  |
| 3 | Fail | 1 | Pass | 1 - verified |  |  |
| 4 | Pass |  |  |  |  |  |
| 5 | Pass |  |  |  |  |  |
| 6 | Fail | 2 | Pass | 2 - verified |  |  |
| 7 | Fail | 3 | Fail | 3 - re-do | Pass | 3 - verified |
| 8 |  |  | Pass |  |  |  |
| 9 |  |  | Fail | 4 |  |  |
| 10 |  |  |  |  | Pass |  |

 Список багов: № 1 – вместо сложения двух переменных а и б реализовано умножение…

Тесты: № 1 – ввод в Textbox букв…..

**Комментарий.** В ходе тестирования среди первых тестов №№ 1, 2, 4, 5 были проведены успешно и отмечены как pass. Тесты № 3, 6, 7 выявили баги (fail) соответственно №№ 1, 2 и 3. В версии №2 разработчик сообщил, что баги №№ 1,2 и 3 исправлены. В ходе Тестов верификации багов (предполагается, что Тесты верификации версии прошли успешно) выяснилось, что тесты №№ 3 (выявил баг № 1) и 6 (баг № 2) прошли успешно (баги помечены как verified), а тест № 7 (баг № 3 ) - вновь выявил тот же баг, о чем сообщено разработчику (re-do). Кроме того во второй версии было продолжено тестирование и проведены тесты №№ 8 и 9. Тест № 8 прошел успешно, а тест № 9 выявил баг № 4. В третьей версии (тесты верификации версии также прошли успешно) разработчик повторно сообщил, что баг № 3 исправлен, что и подтвердило повторное проведение этого теста (тест - pass, баг - verified). Информации об исправлении бага № 4 в третьей версии от разработчика не поступало, поэтому этот тест верификации не проводился. Очередной тест № 10 багов не выявил (pass).

**ТИПОВОЕ ЗАДАНИЕ.**

Программа вычисляет корни квадратного уравнения.

Напомним вам о квадратных уравнениях вообще:

**Квадра́тное уравне́ние** — [алгебраическое уравнение](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5) общего вида

ax^2 + bx + c = 0,

где x — свободная переменная, a, b, c — [коэффициенты](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%8D%D1%84%D1%84%D0%B8%D1%86%D0%B8%D0%B5%D0%BD%D1%82), причём \quad a \ne 0.

Выражение ax^2+bx+c называют *квадратным трёхчленом*[[1]](https://ru.wikipedia.org/wiki/%D0%9A%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5#cite_note-.D0.AD.D0.BD.D1.86.D0.B8.D0.BA.D0.BB.D0.BE.D0.BF.D0.B5.D0.B4.D0.B8.D1.87.D0.B5.D1.81.D0.BA.D0.B8.D0.B9_.D1.81.D0.BB.D0.BE.D0.B2.D0.B0.D1.80.D1.8C_.D1.8E.D0.BD.D0.BE.D0.B3.D0.BE_.D0.BC.D0.B0.D1.82.D0.B5.D0.BC.D0.B0.D1.82.D0.B8.D0.BA.D0.B0.E2.80.9).

*Корень* — это значение переменной x, обращающее *квадратный трёхчлен* в ноль, а *квадратное уравнение* в верное равенство.

Элементы квадратного уравнения имеют собственные названия[[1]](https://ru.wikipedia.org/wiki/%D0%9A%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5#cite_note-.D0.AD.D0.BD.D1.86.D0.B8.D0.BA.D0.BB.D0.BE.D0.BF.D0.B5.D0.B4.D0.B8.D1.87.D0.B5.D1.81.D0.BA.D0.B8.D0.B9_.D1.81.D0.BB.D0.BE.D0.B2.D0.B0.D1.80.D1.8C_.D1.8E.D0.BD.D0.BE.D0.B3.D0.BE_.D0.BC.D0.B0.D1.82.D0.B5.D0.BC.D0.B0.D1.82.D0.B8.D0.BA.D0.B0.E2.80.9):

* a называют *первым* или *старшим* коэффициентом,
* b называют *вторым*, *средним* или *коэффициентом при x*,
* c называют *свободным членом*.

**Общая формула для вычисления корней**

Для нахождения корней квадратного уравнения  ax^2 + bx +c=0 в общем случае следует пользоваться приводимым ниже алгоритмом:

|  |  |  |  |
| --- | --- | --- | --- |
| **Вычислить значение**[**дискриминанта**](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BA%D1%80%D0%B8%D0%BC%D0%B8%D0%BD%D0%B0%D0%BD%D1%82)**квадратного уравнения: таковым для него называется выражение D=b^2 - 4ac** | | | |
| Условие | D > 0 | D = 0 | D < 0 |
| Число действительных корней | корней два | корень один (в некоторых контекстах говорят также о двух равных или совпадающих корнях — его. к тому же, называют **корнем кратности 2**) | делают вывод о том, что корней на множестве действительных чисел нет. |
| Формула | x_{1,2} = \frac{-b \pm \sqrt{b^2-4ac}}{2a}(1) | x_1 = x_2 =  -\frac{b}{2a} |  |

Теперь собственно о самом приложении.

Программистом была разработано windows-приложение **sourses**, реализующее поставленную задачу: расчет корней квадратного уравнения согласно формулы, описанной выше.

Вам, как тестировщику представлена версия sourses 1.0 (в одноименной папке)

Вы должны ее оттестировать и заполнить таблицу:

|  |  |  |
| --- | --- | --- |
| № теста | Версия 1.0 | № бага |
| 1 |  |  |
| 2 |  |  |

Список тестов: ….

Список багов:…..

Далее Вам, как тестировщику представлена версия sourses 1.1 (в одноименной папке)

Вы должны ее оттестировать и заполнить таблицу:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № теста | Версия 1.0 | № бага | Версия 1.1 | № бага |
| 1 |  |  |  |  |
| 2 |  |  |  |  |

Список тестов: ….

Список багов:…..

Далее Вам, как тестировщику представлена версия sourses 2.0 (в одноименной папке)

Вы должны ее оттестировать и заполнить таблицу:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № теста | Версия 1.0 | № бага | Версия 1.1 | № бага | Версия 2.0 | № бага |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |

Список тестов: ….

Список багов:…..

**Задание для самостоятельной работы:**

Аналогично типовому выполнить задание для самостоятельной работы согласно варианта: оттестировать и заполнить таблицы.

Приложение для тестирования имеет 2 версии.

Вариант кратно номеру в списке группы