

Майкопский государственный гуманитарно-технический колледж федерального
государственного бюджетного образовательного учреждения высшего образования
«Адыгейский государственный университет»

ОТЧЁТ ПО ПРАКТИЧЕСКОМУ ЗАДАНИЮ

РАЗРАБОТКА И ТЕСТИРОВАНИЕ КОНСОЛЬНОГО ПРИЛОЖЕНИЯ

Выполнили студенты группы
ИС-33:

Киятова Дана, Тугуз Самира,

Клыков Александр, Навольнев
Эдуард.

2023

СОДЕРЖАНИЕ

Введение

1 Консольное приложение «Определение вида треугольника по трём сторонам и вычисление его площади»

2 Выбор инструмента автоматизированного тестирования

2.1 Краткое описание инструмента автоматизированного тестирования

2.2 Преимущества и недостатки «Qase.io»

3 Тестирование консольного приложения с помощью «Qase.io»

Заключение

Список использованных источников

Приложения

A – Код консольного приложения «Определение вида треугольника по трём сторонам и вычисление его площади»

ВВЕДЕНИЕ

Цель проекта: Создание аналитического отчета по программе "Определение вида треугольника по трём сторонам и вычисление его площади."

Роли участников: В работе над проектом принимали участие 4 человека, каждый из которых выполнял важную роль, наиболее подходящую имеющимся навыкам. А именно:

- Тимлид(Дана). Распределение ролей и задач между участниками проекта, помощь в решении проблем, организация продуктивной деятельности;
- Разработчик консольного приложения(Дана). Пишет код консольного приложения;
- Отвечающий за оформление документации(Самира). Составление отчёта по проделанной работе и анализ результатов;
- Тестировщики(Эдуард и Александр). Отвечают за разработку тест-кейсов для проверки качества ПО, составление подробных отчетов для разработчиков. Создание тест-плана и его прогон.

1 КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ «ОПРЕДЕЛЕНИЕ ВИДА ТРЕУГОЛЬНИКА ПО ТРЁМ СТОРОНАМ И ВЫЧИСЛЕНИЕ ЕГО ПЛОЩАДИ»

В качестве тестируемой программы было взято консольное приложение «Определение вида треугольника по трём сторонам и вычисление его площади», входными данными которого являются три числа. Данные числа интерпретируются как длины сторон треугольника.

Выходные данные – информация о том, является ли треугольник равнобедренным, равносторонним или неравносторонним, площадь полученного треугольника. Предусмотрен ввод вещественных чисел.

Для написания условий были применены некоторые математические правила:

- Определение существования треугольника. Треугольник существует только тогда, когда сумма любых двух его сторон больше третьей. Требуется сравнить каждую сторону с суммой двух других. Если хотя бы в одном случае сторона окажется больше или равна сумме двух других, то треугольника с такими сторонами не существует.
- Определение вида треугольника. Один из признаков равнобедренного треугольника — две стороны равны. Разносторонний – все стороны разные. Равносторонний - все стороны равны.

Приложение выводит на консоль сообщение об ошибке, если введены какие-либо символы, нули или отрицательные числа. В коде программы предусмотрен оператор `try-catch`, чтобы не было аварийного завершения программы при возникновении ошибок. В случае если в блоке `try` обнаружена ошибка, выполняется блок `catch`, который задаёт обработчики для исключений.

Для вычислений используется формула нахождения площади треугольника.

$S = \sqrt{p \cdot (p - a) \cdot (p - b) \cdot (p - c)}$, $P = (a+b+c)/2$ - формула Герона, где a, b, c - стороны треугольника, p - полупериметр.

Для создания приложения была использована среда программирования Visual Studio 2019 и язык программирования C#.

2 ВЫБОР ИНСТРУМЕНТА АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ

2.1 Краткое описание инструмента автоматизированного тестирования

Qase.io –это облачная система управления тестированием, которая позволяет командам эффективно хранить и систематизировать тестовую информацию по продукту, а также организовать совместную работу. Это способствует повышению производительности тестирования.

Легко управлять тестовыми наборами, составлять планы тестирования и выполнять тестовые прогоны.

После изучения нескольких инструментов автоматического управления тестированием, мы остановились на «Qase.io». Выбор нами данного инструмента автоматизированного тестирования обусловлен сочетанием в себе удобного интерфейса, широкого функционала, большого выбора различных обзоров и обучающих статей/видеороликов для ознакомления с данной системой.

2.2 Преимущества и недостатки «Qase.io»

Самый большой плюс – возможность организации совместной работы команды. В нашем случае это очень помогло при проведении тестирования. Каждый мог параллельно с кем-то выполнять свою задачу, при это не мешая друг другу.

Другие преимущества «Qase.io», которые может выделить наша команда:

- Для каждого шага есть поле для вводных данных, а также ожидаемый результат, что позволяет сделать более широкое покрытие тест-кейсами, а также более подробно описать каждый шаг, а самое главное – что мы от него ожидаем;

- У каждого тест-кейса может быть 5 состояний: Passed, Failed, Blocked, Skipped, Invalid;
- Программа бесплатна до 3-х пользователей;
- Достаточно много параметров у каждого тест кейса;
- Тестовый репозиторий: выстраивание тестов в отдельные группы, объединенные логически;
- Составление шагов для кейсов, установка приоритета и серьезности;
- Трекинг времени по каждому тесту;
- Автоматическое заведение дефектов.

Недостатки:

Минусов и неудобств в процессе работы с программой не было обнаружено.

Приглашать людей в проект можно отправив пригласительную ссылку на почту с комментарием или без, все очень просто. Так мы и сделали.

В приведенном ниже рисунке можно увидеть, что мы работали вместе. (Рис. 1)

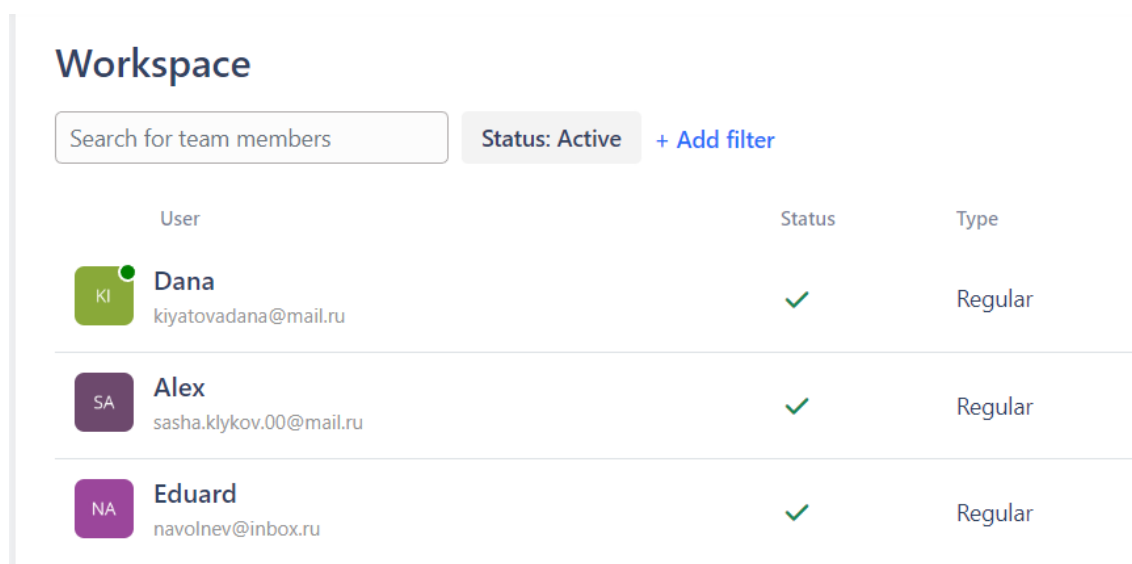


Рисунок 1 – Совместная работа в «Qase.io»

3 ТЕСТИРОВАНИЕ КОНСОЛЬНОГО ПРИЛОЖЕНИЯ С ПОМОЩЬЮ «QASE.IO»

В начале, было создано 3 тестовых набора:

- Проверка на существование треугольника;
- Определение типа треугольника;
- Проверка на вводимые данные.

Каждый из модулей содержит в себе несколько тест-кейсов.(Рис.1)



Рисунок 2 - Структура тестового репозитория

Тесты направлены на выявления дефектов, связанных с неправильным извлечением исходных данных во внутренние переменные или последующим использованием этих переменных. Выполняется проверка отсечения отрицательных значений, символов и нулевых значений.

Следующим этапом тестирования стал прогон всех тестовых наборов. Из 12 тест-кейсов было выявлено 3 успешных тестовых случаев, все ошибки были занесены в дефекты. (Рис. 3)

Defects

[Create new defect](#)

Status: Open

[+ Add filter](#)

Defect	Author	Assignee	Severity
Проверка треугольника на равнобедренность (вещественные числа) D-6 • created 1 week ago	Alex	Unassigned	Critical
Проверка на равнобедренный треугольник D-5 • created 1 week ago	Alex	Unassigned	Critical
Проверка на равносторонний треугольник D-4 • created 1 week ago	Alex	Unassigned	Normal

Рисунок 3 - Дефекты

В первом выявленном дефекте отражена логическая ошибка, которая возникла из-за неправильно продуманного условия определения вида треугольника. Были введены данные ($a=5$ $b=2$ $c=5$), в консоли должно было вывестись, что треугольник является равнобедренный и соответствующая ему площадь. Но вывелась информация ещё о том, что треугольник разносторонний. (Рис. 4)

← Проверка на равнобедренный треугольник

[→ In Progress](#)[✓ Resolve](#)[✗ Invalid](#)[Assign to me](#)[✎ Edit](#)

Description

сумма двух сторон больше третьей - треугольник существует

Равнобедренный треугольник. Площадь: 4,90

Разносторонний треугольник. Площадь: 4,90

Треугольник является равнобедренным, но никак не разносторонним. Ошибка в коде при проверке на разносторонний треугольник. Данная ошибка вводит пользователей в заблуждение.

Рисунок 4 - Пример содержания дефекта 1

Следующий дефект - частный случай. Введенные данные - $a=6$ $b=6$ $c=6$. Не все свойства, принадлежащие равностороннему треугольнику, подходят равнобедренному, поэтому вывод некорректный. (Рис. 5)

← Проверка на равносторонний треугольник

[→ In Progress](#)[✓ Resolve](#)[✗ Invalid](#)[Assign to me](#)[✎ Edit](#)

Description

Треугольник является равносторонним. Выводится информация о том, что треугольник и равнобедренный, но это частный случай и выводить эту информацию необязательно.

Рисунок 5 - Пример содержания дефекта 2

Третья ошибка похожа на первый случай. Введенные данные - $a=2,5$ $b=2,5$ $c=4,5$. (Рис. 6)

← Проверка треугольника на равнобедренность (вещественные числа)

[→ In Progress](#)[✓ Resolve](#)[✗ Invalid](#)[Assign to me](#)[✎ Edit](#)

Description

Вывод: сумма двух сторон больше третьей - треугольник существует

Равнобедренный треугольник. Площадь: 2,45

Разносторонний треугольник. Площадь: 2,45

Вещественные числа вычисляются. Треугольник не разносторонний. Ввод пользователей в заблуждение.

Рисунок 6 - Пример содержания дефекта 3

Ошибки схожи между собой, везде был выявлен некорректный вывод, который не соответствовал ожидаемому результату. Причиной являются неправильно прописанные условия в коде программы.

В первом тестовом наборе не было обнаружено ошибок, чего нельзя сказать о втором и третьем.

Были выявлены логические ошибки, но все вычисления являются правильными.

ЗАКЛЮЧЕНИЕ

В ходе выполнения проектного задания нашей командой была достигнута основная цель проекта и выполнены все задачи.

Процесс тестирования начинается с разработки плана тестирования на основе технического задания, так как тестирование направлено на выявление проблем, связанных с несоответствием разрабатываемого программного продукта требованиям.

Проведена аналитическая работа по выполнению программы "Определение вида треугольника по трём сторонам и вычисление его площади". Обнаружены ошибки, выявленные путем успешного автоматизированного тестирования, которые не были ожидаемы при разработке программы.

Были произведены сбор полученных данных в отчёте, определение объёма и срок работы, разработка тестируемой программы и самих тестов. Наиболее трудоемким и ответственным процессом оказалось построение тестовых наборов и тестовых случаев.

Также мы освоили инструмент «Qase.io», с помощью которого осуществлялось тестирование. Он является одной из самых простых в освоении систем управления тестированием, что позволило не тратить время на изучение всех возможностей тестовой системы. Благодаря управлению ролями и разрешениями для пользователей системы, участники команды обнаружили свои сильные и слабые стороны, а также проявляли свои лучшие качества в процессе работы. Такие как: ответственность, продуктивность, дружелюбность, гибкость.

Тест программы по определению вида треугольника по трём сторонам и вычисление его площади

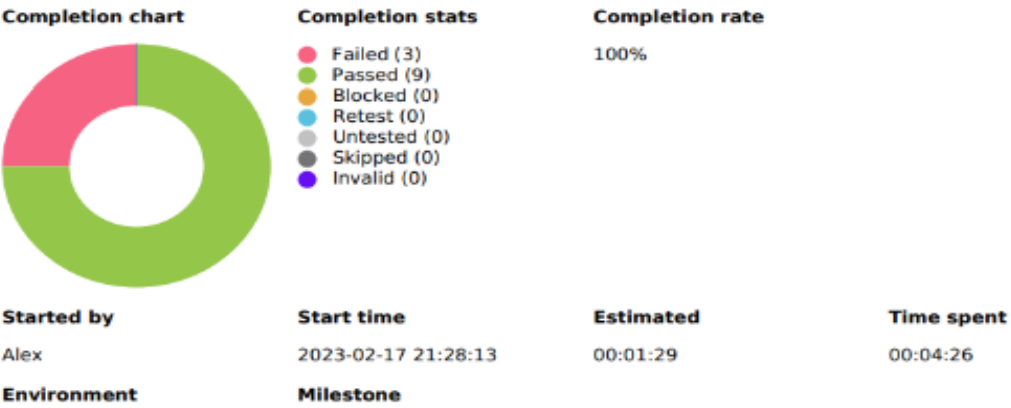


Рисунок 7 - Результаты тестирования

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Топ-12 лучших систем управления тестированием 2020 // software-testing.ru URL: <https://www.software-testing.ru/library/around-testing/management/3457-top-12-best-test-management-systems-2020> (дата обращения: 14.02.2023).

2 Обзор системы управления тестированием Qase // medium.com URL: <https://medium.com/@nik3r/обзор-системы-управления-тестированием-qase-267bfcd8513c> (дата обращения: 14.02.2023).

3 How to Determine if Three Side Lengths Are a Triangle // wikihow URL: https://translated.turbopages.org/proxy_u/en-ru.ru.3ac0d8eb-64023c57-67a9f21a-74722d776562/https/www.wikihow.com/Determine-if-Three-Side-Lengths-Are-a-Triangle (дата обращения: 15.02.2023).

ПРИЛОЖЕНИЕ А

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace OurProject
8 {
9     Ссылка: 0
10    internal class Program
11    {
12        Ссылка: 0
13        static void Main(string[] args)
14        {
15            try
16            {
17                Console.WriteLine("Введите сторону a = ");
18                double a = Convert.ToDouble(Console.ReadLine());
19                Console.WriteLine("Введите сторону b = ");
20                double b = Convert.ToDouble(Console.ReadLine());
21                Console.WriteLine("Введите сторону c = ");
22                double c = Convert.ToDouble(Console.ReadLine());
23                double S;
24                double p = (a + b + c) / 2;
25                S = Math.Sqrt(p * (p - a) * (p - b) * (p - c));
26                if (a <= 0 || b <= 0 || c <= 0)
27                {
28                    Console.WriteLine("Сторона не может быть меньше или равной 0. Треугольник не существует.");
29                }
30                else if (a + b > c && a + c > b && b + c > a)
31                {
32                    Console.WriteLine("сумма двух сторон больше третьей - треугольник существует");
33                    if (a == b && b == c && a == c)
34                    {
35                        Console.WriteLine("Равносторонний треугольник. Площадь: {0:f2}", S);
36                    }
37                    if (a == b || b == c || a == c)
38                    {
39                        Console.WriteLine("Равнобедренный треугольник. Площадь: {0:f2}", S);
40                    }
41                    if (a != b || b != c || a != c)
42                    {
43                        Console.WriteLine("Разносторонний треугольник. Площадь: {0:f2}", S);
44                    }
45                }
46                else Console.WriteLine("сумма двух сторон меньше третьей - треугольник не существует");
47                Console.ReadKey();
48            }
49            catch (Exception e)
50            {
51                Console.WriteLine(e.Message + " Символы или пробелы недопустимы.");
52                Console.ReadLine();
53            }
54        }
55    }
56 }
57
58
59
60
61
62
63
```

Рисунок 1 – Код консольного приложения