



РАЗРАБОТКА И ТЕСТИРОВАНИЕ КОНСОЛЬНОГО ПРИЛОЖЕНИЯ



Выполнили студенты группы ИС-33:
Киятова Дана, Тугуз Самира,
Клыков Александр, Навольнев Эдуард.



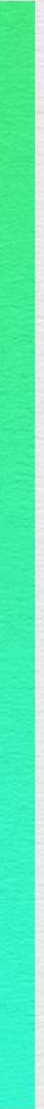
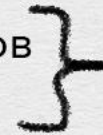
{ Цель проекта:

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.



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.

Для написания условий были применены некоторые математические правила:

- Определение существования треугольника.
- Определение вида треугольника.

Приложение выводит на консоль сообщение об ошибке, если введены какие-либо символы, нули или отрицательные числа. В коде программы предусмотрен оператор try-catch,

Для вычислений используется формула нахождения площади треугольника.

- формула Герона

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





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.

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.





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.



Workspace

Status: Active

+ Add filter

User	Status	Type
<div><div>KI</div><div>Dana kiyatovadana@mail.ru</div></div>	✓	Regular
<div><div>SA</div><div>Alex sasha.klykov.00@mail.ru</div></div>	✓	Regular
<div><div>NA</div><div>Eduard navolnev@inbox.ru</div></div>	✓	Regular

Рисунок 1. – Совместная работа в «Qase.io»

В начале, было создано 3 тестовых набора:

Каждый из модулей содержит в себе несколько тест-кейсов



Рисунок 2. – Структура тестового репозитория

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		<input type="text" value="Search for defects"/>	Status: Open + Add filter	
Defect	Author	Assignee	Severity	
 Проверка треугольника на равнобедренность (вещественные числа) D-6 • created 3 hours ago	 Alex	Unassigned	 Critical	
 Проверка на равнобедренный треугольник D-5 • created 3 hours ago	 Alex	Unassigned	 Critical	
 Проверка на равносторонний треугольник D-4 • created 3 hours ago	 Alex	Unassigned	 Normal	

Рисунок 3. – Дефекты

← 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 4. - Пример содержания дефекта 1

← 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

← 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 6. - Пример содержания дефекта 3

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

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.

Участники команды обнаружили свои сильные и слабые стороны, а также проявляли свои лучшие качества в процессе работы. Такие как: ответственность, продуктивность, дружелюбность, гибкость.

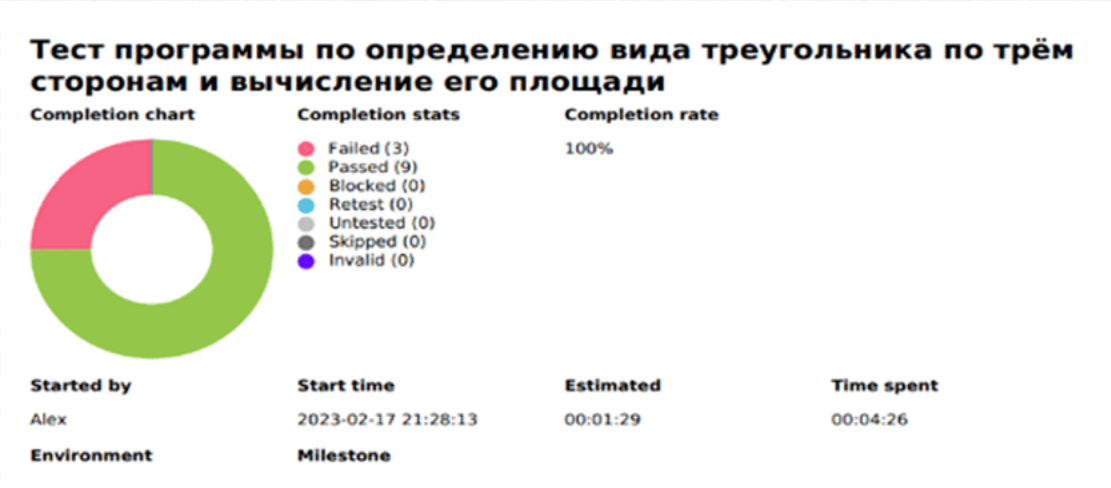


Рисунок 7. – Результаты тестирования