МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»  
(Самарский университет)

Институт информатики и кибернетики\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра программных систем\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОТЧЕТ**  
  
по лабораторной работе №4

«Создание баг-репортов»

по дисциплине «Тестирование и отладка ПО»

Обучающаяся в группе 6301-020302D Соколова Алёна

Преподаватель Лобанков Антон Алексеевич

Самара 2024

**Оглавление**

[Баг-репорты 2](#_Toc181052006)

[ID дефекта: DEF-001 2](#_Toc181052007)

[ID дефекта: DEF-002 4](#_Toc181052008)

[ID дефекта: DEF-003 5](#_Toc181052009)

[ID дефекта: DEF-004 6](#_Toc181052010)

[Заключение 8](#_Toc181052011)

**Цель работы**

Целью выполнения данной лабораторной работы является ознакомление с процессом создания баг-репортов для фиксации ошибок в программном обеспечении. В ходе работы необходимо изучить структуру и содержание баг-репортов, а также самостоятельно составить баг-репорты на основе тестирования программы, соответствующей индивидуальному заданию.

**Задание**

1. Используя тренажер https://playground.learnqa.ru/puzzle/triangle найдите все баги;
2. Описать баги в виде полноценных баг-репортов, заполнив все необходимые атрибуты отчета о дефекте;
3. Составить отчет о проделанной работе.

# Баг-репорты

# ID дефекта: DEF-001

*Заголовок:* Нулевые стороны определяются как равносторонний треугольник.

*Описание:* При проверке на равносторонний треугольник, если задать все стороны равными нулю, программа ошибочно определяет треугольник как равносторонний. Однако корректный треугольник должен иметь стороны, которые равны между собой и больше нуля.

*Классификация:*

*Тип дефекта:* Логическая ошибка;

*Серьезность:* Major (влияет на корректность основной функциональности);

*Приоритет:* Высокий (нуждается в исправлении, так как влияет на базовую логику определения треугольника).

*Предусловия:* у равностороннего треугольника все стороны равны, но не равны нулю.

*Шаги для воспроизведения:*

* + - 1. Определить кейсы с целыми числами (равносторонний треугольник);
      2. Дать нулевые значения каждой стороне;
      3. Запустить программу и определить тип треугольника.

*Ожидаемый результат****:*** Программа должна вывести, что треугольник не является равносторонним, так как стороны не могут быть нулевыми.

*Фактический результат****:*** Программа определяет треугольник как равносторонний, несмотря на то, что его стороны равны нулю.

*Дополнительно****:*** Проблема связана с отсутствием проверки на нулевые значения сторон. Необходимо уточнить валидацию, чтобы треугольник считался равносторонним только в случае, когда стороны равны и больше нуля.

*Вложения:* Скриншот, на котором показаны введенные нулевые значения для сторон треугольника и результат, в котором треугольник определен как равносторонний.

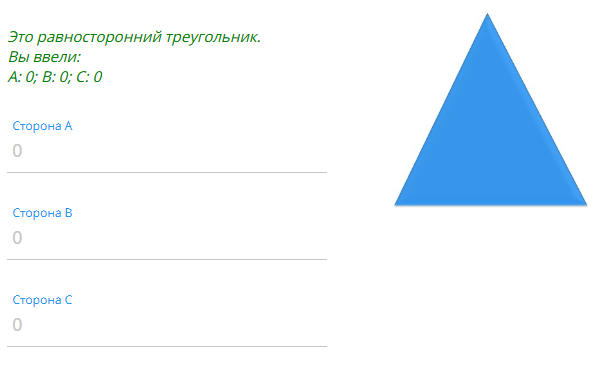


Рисунок 1 – Скриншот по баг-репорту 1

# ID дефекта: DEF-002

*Заголовок:* Форма неправильно работает с нецелыми числами.

*Описание:* При проверке работы программы с вещественными числами (нецелыми значениями) она ошибочно определяет треугольник как "НЕ треугольник" в случаях, когда должно быть определено, что это тупоугольный или прямоугольный треугольник.

*Классификация:*

*Тип* *дефекта:* Логическая ошибка;

*Серьезность:* Major (влияет на корректность основной функциональности);

*Приоритет:* Высокий (нуждается в исправлении, так как влияет на базовую логику работы с треугольниками).

*Предусловия:* Программа должна корректно обрабатывать вещественные числа для определения типа треугольника.

*Шаги воспроизведения:*

* + - 1. Определить кейсы с целыми числами (тупоугольный, равносторонний, прямоугольный, треугольник);
      2. Создать тестовые кейсы с нецелыми значениями для сторон треугольника, включающие тупоугольный и прямоугольный треугольник;
      3. Запустить программу для определения типа треугольника.

*Ожидаемый результат****:*** Программа правильно идентифицирует треугольники:

1. Тупоугольный треугольник (первое вложение);
2. Прямоугольный треугольник (второе вложение).

*Фактический результат****:*** Программа выводит, что это "НЕ треугольник" для обоих случаев.

*Дополнительно:* Необходима проверка корректной работы программы с вещественными числами, так как сейчас она может ошибочно не распознавать тип треугольника при их использовании.

*Вложения:*

1. Скриншот формы с введенными значениями для тупоугольного треугольника, на котором видно ошибочное определение результата;
2. Скриншот формы с введенными значениями для прямоугольного треугольника, также демонстрирующий ошибочный результат.

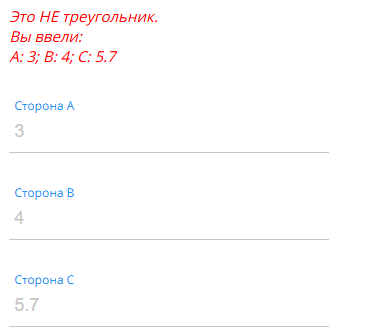


Рисунок 2 – Скриншот №1 по баг-репорту 2

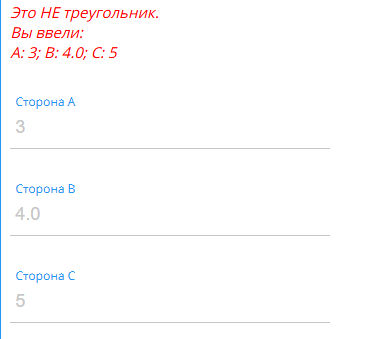


Рисунок 3 – Скриншот №2 по баг-репорту 2

# ID дефекта: DEF-003

*Заголовок:* Поле C не проверяется на заполненность.

*Описание:* При проверке заполненности полей формы программа не учитывает поле C. Если оставить поле C пустым, форма не выдает предупреждение о необходимости его заполнения и пытается определить тип треугольника, что приводит к некорректному результату.

*Классификация:*

*Тип* *дефекта:* Ошибка валидации;

*Серьезность:* Major (влияет на корректность основной функциональности);

*Приоритет:* Средний (исправление желательно для предотвращения ошибок в работе с неполными данными).

*Предусловия:* Поля A и B проверяются на заполненность, тогда как поле C также должно проверяться перед определением типа треугольника.

*Шаги воспроизведения:*

* + - 1. Открыть форму для ввода значений сторон треугольника;
      2. Ввести значения только в поля A и B, оставив поле C пустым;
      3. Запустить программу для определения типа треугольника.

*Ожидаемый результат****:*** Программа выводит предупреждение о необходимости заполнения всех полей, включая поле C, перед определением типа треугольника.

*Фактический результат****:*** Программа не выводит предупреждение и пытается определить тип треугольника, несмотря на пустое поле C.

*Дополнительно:* Требуется обновить логику валидации формы, чтобы проверять заполненность всех полей, включая поле C, перед выполнением определения типа треугольника.

*Вложения:* Скриншот формы, показывающий, что поле C оставлено пустым и при этом не отображается предупреждение о его заполнении, программа продолжает попытку определить тип треугольника.

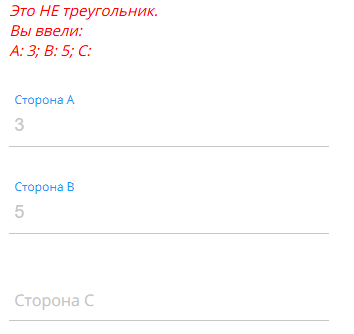


Рисунок 4 – Скриншот по баг-репорту 3

# ID дефекта: DEF-004

*Заголовок:* Форма уязвима к XSS при вводе тега <SCRIPT> в разных регистрах.

*Описание:* При вводе тега <SCRIPT> в верхнем регистре или в смешанном регистре форма не распознает это как попытку XSS-атаки и позволяет выполнить JavaScript-код. Это свидетельствует о недостаточной проверке тега <script>, так как фильтрация работает только при использовании нижнего регистра. Кроме того, после ввода тега <SCRIPT> в поле A программа отображает сообщение, в котором не упоминает поля B и C, что создает дополнительные уязвимости.

*Классификация:*

*Тип дефекта:* Уязвимость безопасности (XSS);

*Серьезность:* Critical (уязвимость позволяет исполнение вредоносного кода);

*Приоритет:* Высокий (необходимо исправить для защиты приложения от XSS-атак).

*Предусловия:* Форма проверяет поле на наличие подстроки <script> в нижнем регистре, но не обрабатывает теги, написанные в верхнем или смешанном регистре.

*Шаги воспроизведения:*

1. Открыть форму и ввести значение <SCRIPT>alert("XSS")</SCRIPT> в поле A (первую сторону);
2. Ввести любые значения в поля B и C;
3. Отправить форму.

*Ожидаемый результат:*

* + - 1. Форма распознает ввод как попытку XSS-атаки и предотвращает исполнение JavaScript-кода независимо от регистра в теге <script>;
      2. Программа должна отображать все введенные значения, включая A, B и C.

*Фактический результат:*

Форма не распознает ввод как XSS-атаку и позволяет исполнение JavaScript-кода при вводе тега <SCRIPT> в верхнем регистре;

При выводе результатов программа отображает только значение A: «Вы ввели A: ...», не упоминая о значениях B и C.

*Дополнительно:* Требуется обновить механизм фильтрации для проверки тега <script> вне зависимости от регистра символов, а также исправить логику отображения, чтобы она показывала все введенные значения.

*Вложения:* Скриншот формы изображен на рисунке 4 с введенным тегом <SCRIPT> в верхнем регистре, подтверждением выполнения JavaScript-кода и выводом, показывающим только значение A: «Вы ввели A: ...», без упоминания B и C.

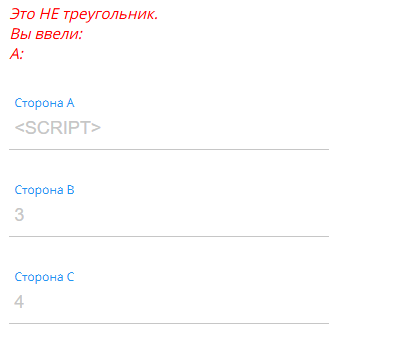


Рисунок 5 – Скриншот по баг-репорту 4

# Заключение

В процессе работы были изучены правила и структура составления баг-репортов. В результате тестирования приложения для генерации случайных чисел были составлены баг-репорты, фиксирующие выявленные ошибки и предлагающие пути их устранения. Эти отчеты помогут разработчикам улучшить качество приложения и обеспечить его корректную работу.