**1. Тестирование поля ввода возраста**

**Требования:**

* Только целые числа.
* Возраст ≥ 18.

**Тест-кейсы:**

| **ID** | **Шаги** | **Ожидаемый результат** | **Тип теста** |
| --- | --- | --- | --- |
| 1 | Ввести "20" | Данные принимаются | Позитивный |
| 2 | Ввести "18" | Данные принимаются | Позитивный |
| 3 | Ввести "17" | Ошибка: "Возраст < 18" | Негативный |
| 4 | Ввести "abc" | Ошибка: "Только числа" | Негативный |
| 5 | Ввести "20.5" | Ошибка: "Только целые числа" | Негативный |
| 6 | Оставить поле пустым | Ошибка: "Поле обязательно" | Негативный |

**2. Тестирование приложения для определения типа треугольника**

**Входные данные:** 3 целых числа (стороны a, b, c).

**Тест-кейсы:**

| **ID** | **Шаги** | **Ожидаемый результат** | **Тип теста** |
| --- | --- | --- | --- |
| 1 | **Ввод (**5 5 5**)** | "равносторонний" | Позитивный |
| 2 | **Ввод (**5 5 3**)** | "равнобедренный" | Позитивный |
| 3 | **Ввод (**3 4 5**)** | "разносторонний" | Позитивный |
| 4 | **Ввод (**0 5 5**)** | Ошибка: "Сторона ≤ 0" | Негативный |
| 5 | **Ввод (**1 2 3**)** | Ошибка: "Не треугольник" | Негативный |
| 6 | **Ввод (**"a" 2 3**)** | Ошибка: "Только числа" | Негативный |

**3. Попарное тестирование (Pairwise Testing) формы регистрации**

**Цель:** Проверить все возможные комбинации значений полей, минимизируя количество тест-кейсов за счет покрытия всех пар взаимодействий между параметрами.

**Параметры и значения:**

1. **ФИО** (обязательное):
   * Валидное (например, "Иванов Иван Иванович")
   * Невалидное (пустое, "123", спецсимволы)
2. **Пол** (обязательное):
   * М
   * Ж
3. **Возраст** (обязательное):
   * Валидное (18, 199, 30)
   * Невалидное (17, 200, "abc")
4. **Документ** (обязательное, зависит от пола):
   * Для **М**: "Паспорт", "Водительское удостоверение", "Военный билет"
   * Для **Ж**: "Паспорт", "Водительское удостоверение"
5. **Место работы** (необязательное):
   * Заполнено ("ООО Рога и копыта")
   * Пустое

**Список тест-кейсов (попарное покрытие):**

| **№** | **ФИО** | **Пол** | **Возраст** | **Документ** | **Место работы** | **Ожидаемый результат** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Валидное | М | 18 | Военный билет | Заполнено | Успешная регистрация |
| 2 | Валидное | Ж | 199 | Водительское удостоверение | Пустое | Успешная регистрация |
| 3 | Невалидное | М | 17 | Паспорт | Заполнено | Ошибка: "Возраст < 18" |
| 4 | Невалидное | Ж | 200 | Паспорт | Пустое | Ошибка: "Возраст > 199" |
| 5 | Валидное | М | 30 | Водительское удостоверение | Пустое | Успешная регистрация |
| 6 | Валидное | Ж | 18 | Военный билет | Заполнено | Ошибка: "Документ недоступен для Ж" |
| 7 | Пустое | М | 199 | Паспорт | Заполнено | Ошибка: "Поле ФИО обязательно" |
| 8 | Пустое | Ж | 18 | Водительское удостоверение | Пустое | Ошибка: "Поле ФИО обязательно" |

**Итог:** 8 тест-кейсов покрывают все значимые пары параметров

**4. SQL запрос для получения количества книг по жанрам, зарегистрированных после 12 апреля 2003 года:**

SELECT

g.name AS genre\_name,

COUNT(DISTINCT b.id) AS books\_count

FROM

genres g

JOIN

book\_genres bg ON g.id = bg.genre\_id

JOIN

books b ON bg.book\_id = b.id

JOIN

book\_reg br ON b.id = br.id

WHERE

br.date > '2003-04-12'

GROUP BY

g.name

ORDER BY

books\_count DESC;

Ожидаемый результат:

| **genre\_name** | **books\_count** |
| --- | --- |
| Русская классика | 2 |
| Зарубежные романы | 1 |
| Драма | 1 |

*Примечание: В выборку попадут книги с id 1, 3 и 5, так как только они имеют записи в таблице book\_reg с датами после 12.04.2003 (01.05.2006, 05.07.2004 и 30.09.2015 соответственно).*