Solutions

2.1

- e.g. `test_return_twice_raises` ✓.
- Named for the behaviour, not the method it calls ✓.

2.2

- The behaviour-named test survives ✓.
- The method-named test breaks on the rename while pinning nothing that changed ✓.

2.3

- Single responsibility principle ✓.
- One reason to change per class: loans change for loan rules, fines for fine rules ✓.

3.1

- Test 1: arrange a 20-day loan, act `renew(loan, 7)`, assert the result is 27 ✓.
- Test 2: arrange a 20-day loan, act `renew(loan, 11)`, assert it raises the limit error ✓.
- Test 3: assert the loan still reads 20 after the refusal ✓.
- Each assert pins one promised outcome – the value, the raise, the unchanged state ✓.
- Each test builds its own loan: no shared mutable state, any order runs green ✓.

3.2

- They still pass, untouched – behaviour preserved while structure moved ✓.
- The move follows the single responsibility principle ✓.
- Coupling went down (callers depend on less); cohesion went up (one purpose per class) ✓.
- Asserting private internals couples tests to storage – the move would have broken them ✓.