INVESTIGATION

A £120 order wrongly accepted against a £100 limit

WRITTEN AGAINST	events.audit · 12 record(s) · 13:28:45.417->00:00:01.000
AUTHORED VIEW	all event types
LOG	events.audit - loaded snapshot; log unchanged-metadata; graph unchanged-met...
CREATED	2026-11-01T13:45:13.501675Z
RENDERED	2026-11-01 13:45:13.523

NARRATIVE - THE AUTHOR'S ACCOUNT, NOT LOG EVIDENCE

Controlled demonstration; Codex knew the seeded fault before this investigation. This report records actual Analyser observations, not a blind discovery. Source supplied separately is unverified by the Analyser as the executed binary. Native EvidenceRun/run.py executes tests; Analyser does not replay the application.

NARRATIVE - THE AUTHOR'S ACCOUNT, NOT LOG EVIDENCE

Requirement: approve exactly when valuePence <= limitPence. Faulty LimitCheck.java line15 instead uses >. Retain return true: both outcomes must propagate to Decision. Amounts are integer pence.

Record #3

```
event: Order{quantity=10, unitPricePence=1200, limitPence=10000}
1. orderValue  thread=main  method=onOrder  annotation=OnEventHandler  quantity=10
unitPricePence=1200  valuePence=12000
2. limitCheck  thread=main  method=check  annotation=OnTrigger  limitPence=10000  withinLimit=true
3. decision  thread=main  method=decide  annotation=OnTrigger  outcome=ACCEPTED  reason=within limit
```

WHAT IS WRONG - RECORD #3

Seeded known fault: this event records valuePence=12000, limitPence=10000, withinLimit=true and outcome=ACCEPTED. The stated requirement is value <= limit.

LIKELY CAUSE / SUGGESTED FIX

LimitCheck.java line 15 uses > in the faulty source. Replace > with <=; retain return true so accepted and rejected outcomes both reach Decision. Confirm with the native runner and boundary cases, not by the graph alone.

Record #3

```
event: Order{quantity=10, unitPricePence=1200, limitPence=10000}
1. orderValue  thread=main  method=onOrder  annotation=OnEventHandler  quantity=10
unitPricePence=1200  valuePence=12000
2. limitCheck  thread=main  method=check  annotation=OnTrigger  limitPence=10000  withinLimit=true
3. decision  thread=main  method=decide  annotation=OnTrigger  outcome=ACCEPTED  reason=within limit
```

NARRATIVE - THE AUTHOR'S ACCOUNT, NOT LOG EVIDENCE

Scope: synthetic known-seeded fault, ordinary unsigned audit, no proof of binary identity or universal correctness. Lifecycle begins at wall time before test clock changes to1000ms, so time-order warnings are expected; this investigation uses record indexes, not time comparisons. Seven matched inputs are rerun by the native harness after repair; see separate corrected CSV/report.