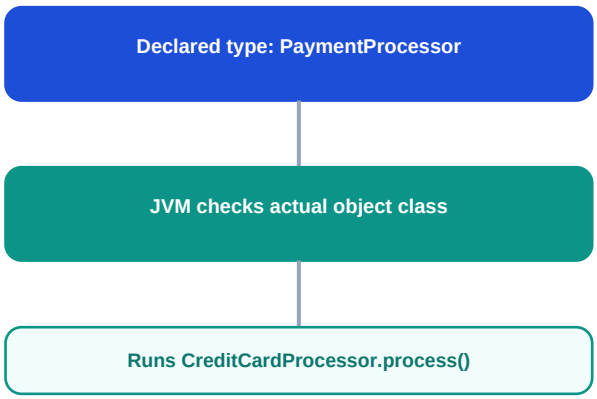
Polymorphism & Liskov Substitution Review Checklist

Catch Liskov Substitution violations, missing @Override annotations, and silent-failure overrides before they ship.

1. Overloading vs. Overriding

Type	Mechanism	Resolved When?
Compile-time (static)	Overloading — same name, different params	At compile time, by declared type
Runtime (dynamic)	Overriding — subclass redefines parent method	At runtime, by actual object class

2. Dynamic Dispatch at a Glance



Overloading = compile-time

(decided by declared type)

Overriding = runtime

(decided by actual object on the heap = dynamic dispatch)

3. Liskov Substitution Red Flags

- ■ Does the override silently do LESS than the parent guaranteed (e.g. an early return that skips required logic)?
- ■ Does the override throw a NEW exception type the caller was never built to catch?
- ■ Is @Override missing from any method meant to override a parent — a typo'd signature creates a silent overload instead?
- ■ Does a Page Object or adapter override skip calling super() when the parent's behavior should still apply?
- ■ Would swapping this implementation for another one that shares its interface change caller behavior unexpectedly?
- ■ Is any code referencing a concrete class directly instead of its interface, defeating extensibility?

4. The Fix Pattern

```
// Before: silently drops the transaction
if (amount > 50000) { return; }

// After: fails loud, matches the interface contract
if (amount > 50000) {
  throw new TransactionLimitException(...);
}
```

> **Rule of thumb:** a class is only truly substitutable for its parent if callers can't tell the difference except by the result improving. If behavior silently changes, it's not polymorphism working — it's a Liskov violation waiting to be a defect.
