

DAY 2 2026年9月9日（水）

テストで裏を取る

Claude Code にテストを書かせ、出てきたものを仕様に照らして採否できる状態を目指します。仕込まれた3件のバグは先にテストで赤を出してから直し、309行のレガシーコードは出力を固定したまま段階的に整理します。

この日の演習

演習は上から順に進みます。全員が終わるのを待ってから次へ進むので、詰まったら手を挙げてください。早く終わった方には各演習に発展課題があります。

ねらい

TODO のまま残っている遷移テストを Claude Code に書かせ、値と例外型まで見ているかを確認してから採用します。

さわるファイル

書く

src/test/java/com/example/order/service/OrderServiceImplTest.java

TODO 4件をテストに置き換える

書く

src/main/java/com/example/order/service/OrderServiceImpl.java

遷移ルールの確認と一時的な書き換え

読む

docs/生成物チェックリスト.md

採否の基準。既知バグの赤は除く

手順

- 1 実行** Day1 のアプリは起動したままで構わないので、2枚目の統合ターミナルで handson フォルダを開き、`.\mvnw.cmd test -Dtest=OrderServiceImplTest` を打ちます。Tests run: 9, Failures: 1 で終わるのが配布時点の正常な状態で、この1件は仕込みなので今日は直しません。
- 2 書く** claude を起動し、`@OrderServiceImplTest.java` と `@OrderServiceImpl.java` を渡して `changeStatus` の TODO を実装させます。異常系は例外型まで確認する、`save` は `verify` する、この2条件を必ず添えます。
- 3 答え合わせ** 差分を1メソッドずつ読み、`assertThat` が値を比較しているか、`assertThatThrownBy` が `InvalidOrderStateException` を指定しているかを見ます。足りなければ条件を足して出し直し、採用したら `.\mvnw.cmd test -Dtest=OrderServiceImplTest` をもう一度打ちます。
- 4 書く** CONFIRMED から CANCELLED への変更が成功する前提のテストを1本足して、もう一度走らせます。緑になった人は `validateStatusTransition` の CONFIRMED の case から CANCELLED を一時的に外し、赤を再現してから戻してください。

できたら

- ☐ Tests run の合計が配布時の9件から、自分が足した本数ぶん増えている
- ☐ 異常系のテストが `assertThatThrownBy` で例外型まで指定されている
- ☐ 赤の内訳を言える。配布時からの DELIVERED キャンセルが1件、CONFIRMED から CANCELLED は D1-5 を取り込んでいれば緑、未了なら2件目の赤になる

考えること

CANCELLED からの遷移が全部不可であることを、遷移先を1つだけ試して確かめたとと言えるか。

AI の出方

テストメソッドの本数も名前も毎回変わります。黙って @Disabled が付いて赤が隠れることがあるので、付いていたら外して赤のまま見てください（自分で理由コメント付きで付けるのは構いません）。正常系で `when(orderRepository.save(any(Order.class)))` のスタブが抜けると、保存結果をログに出す行で `NullPointerException` になるので、そのときは `save` のスタブを足してと言い直します。

D2-1+ `createOrder` の @Nested に残る TODO 3件（数量0、単価が負、納品予定日が過去）を同じ手順で埋めます。バリデーションは `OrderCreateRequest` の注釈と Controller の @Valid でしか動きません。単体テストはサービスを直接呼ぶので検証は走らず、例外を期待して書くと3件とも落ちます。落ちる前提で書いてから、サービス層に検証を足すか Controller 経由のテストに変えるかを Claude Code に切り分けさせてください。

くわしく（背景・詰まったときの対処）

`OrderServiceImplTest` は `findById`・`createOrder`・`changeStatus`・`cancelOrder`・`findAll` の代表ケースが正常系・異常系とも実装済みで、`changeStatus` の @Nested に4件の TODO コメントが残っています。

CONFIRMED から SHIPPED、SHIPPED から DELIVERED、CANCELLED からの遷移は全部不可、同一ステータスへの遷移は不可、の4件です。テスト基盤は組んであるので @ExtendWith や @Mock を書き足す必要はありません。AssertJ の `assertThat` と `assertThatThrownBy`、Mockito の `when` と `verify`、既存テストと同じ書き方に揃えるだけです。追記先は既存ファイルで、新規ファイルは作りません。

`./mvnw.cmd test -Dtest=OrderServiceImplTest` は9件中1件が赤です。Day2 で初めてテストを走らせると赤が1件出ますが、これは配布時点で仕込んである失敗で、環境の不備ではありません。DELIVERED の受注をキャンセルできてしまう穴が `cancelOrder` に残っていて、例外が出ずにキャンセルが通り、スタブしていない `save` が `null` を返すので、続く行で `NullPointerException` になります。画面に出るのは `expecting ...`

`InvalidOrderStateException but was ... NullPointerException` の行です。Day3 の題材なので、今日は直しません。走らせる前に配布時点の状態を一度見ておくと、自分が足したテストの失敗と切り分けやすくなります。

@ を打つと候補が出るので、ファイル名でもパスでも選べます。実行結果の読み方には癖がひとつ。Maven のコンソールに出るのはクラス単位の集計行と落ちたテストの名前だけで、通ったテストのメソッド名は表示されません。名前まで確かめたいときは `target/surefire-reports/TEST-com.example.order.service.OrderServiceImplTest$ChangeStatusTest.xml` を開き、`testcase name` に自分が足したメソッド名が並ぶのを見てください。

同一ステータスへの遷移には専用のチェック文がありませんが、どの case も自分自身を遷移先に挙げていないので例外が出ます。このテストは配布時点でも緑です。もうひとつ、CONFIRMED から CANCELLED が成功

する前提のテストは、Day1 の D1-5 を済ませていなければ落ちます。落ちたときは validateStatusTransition の CONFIRMED の case を開いて、SHIPPED しか許していないことを自分で確かめてください。生成物の誤りなのか実装の穴なのかは、ここが分かれ目です。採否の基準は docs/生成物チェックリスト.md の B-5 に書いてあります。

ねらい

仕込まれた3件のバグを、直す前にテストで赤くしてから潰します。

さわるファイル

書く

src/main/java/com/example/order/buggy/BuggyOrderCalculator.java
exercises から移動。修正もここ

新規

src/test/java/com/example/order/buggy/BuggyOrderCalculatorTest.java
生成させるテストの置き場所

手順

1 実行

PowerShell (claude を起動していたら Ctrl+D か /exit で一度抜けます) で New-Item -ItemType Directory -Force src\main\java\com\example\order\buggy を打ち、続けて Move-Item exercises\day2\BuggyOrderCalculator.java src\main\java\com\example\order\buggy\ を打ちます。Maven は src/main/java の下しか読まないの、この一手がないとテストがコンパイルで止まります。

2 書く

claude を起動して /gen-test @BuggyOrderCalculator.java だけを送り、出てきたテストの置き場所が src/test/java/com/example/order/buggy/ でなければ「buggy パッケージに置いて」と1行足します。数量10ちょうどのケースが無ければ「数量10ちょうども見て」と足す、というふうに1回に1つずつ条件を積みみます。

3 答え合わせ

各 assertThat の期待値が仕様の値か、いまの実装が返す値かを1件ずつ見て、calculateDiscount(10, 50000) が 500000 になっていたら 475000 に手で直します。claude を抜けて .\mvnw.cmd test -Dtest=BuggyOrderCalculatorTest を打ち、AssertJ が2行に割って出す expected: 475000 と but was: 500000 を読みます (BUILD FAILURE で終わりますが想定どおりです)。

4 書く

比較演算子、total の初期値、日付比較の3か所をまとめて直し、もう一度走らせて全部緑になるのを見ます。緑になったら比較演算子だけ > に戻して赤が戻るのを確かめ、>= に直し直して緑に戻してから終わります。

できたら

- ❑ 修正前の実行で expected: 475000 と but was: 500000 の2行 (AssertJ は改行して出します) を自分の画面で読んだ
- ❑ 存在しない顧客の合計と ORD-003 の遅延検出も赤く出ている (Failures と Errors のどちらに出るかは書き方で変わる)
- ❑ 修正後は Failures も Errors も 0 になり、java
src\main\java\com\example\order\buggy\BuggyOrderCalculator.java の出力から「バグ発生!」が消えて、遅延1件 (ORD-003) と存在しない会社の合計0円が表示される

考えること

10個ちょうどで割引が乗らないのは、仕様の読み違いか実装の書き間違いか。そう判断した根拠はどこにあるか。

AI の出方

期待値を仕様ではなく現在の実装値で埋めてくることがあります。生成したテストが最初から全部緑なら、それはバグごと固定された合図なので「期待値は仕様の値で書いて。いまの実装が返す値に合わせないで」と言い直してください。getCustomerOrderTotal は戻り値が null になるので、int で受ける書き方だと Failures ではなく Errors 側に出ます。

D2-2+ 直した3か所に再発防止のテストを足します。境界ちょうどの 10・20・50、複数受注を持つ東京電機工業の合計、SHIPPED の ORD-001 が遅延に入らないこと。演算子を > に戻したときに赤で気づけるのは境界ちょうどの3点で、手前と直後 (9・11、19・21、49・51) は > でも >= でも同じ値を返すため、こちらは修正で別の境界を壊していないかの確認に使います。あわせて、3か所をまとめてではなく1か所ずつ直して都度テストを走らせる進め方も試してください。

くわしく (背景・詰まったときの対処)

BuggyOrderCalculator は受注5件を内部に持つ単体クラスで、公開メソッドは3つです。仕様は Javadoc に書いてあります。数量10個以上で5%、20個以上で10%、50個以上で15%の割引。指定顧客の受注が1件もなければ合計0円。納品予定日の当日にまだ出荷されていない受注も遅延に数える。この3つを頭に入れてからテストを書きます。

置き場所には一手いります。このクラスは exercises/day2 にあり、Maven は src/main/java の下しかコンパイルしないため、テストクラスから参照できません。移さずに走らせると「シンボルを見つけられません」でコンパイルが止まります。移したあとは修正もそのファイルに対して行い、exercises 側には戻しません。

期待値は電卓で先に出しておきます。突き合わせる基準がないと、生成物の粗は見つけにくくなります。数量 10・単価50000なら5%引きで475000円、いまの実装が返すのは500000円。getCustomerOrderTotal は合計の初期値が null のままなので、該当受注ゼロの顧客では null がそのまま返ります。メソッド自体は例外を投

げません。NullPointerException になるのは戻り値を int で受けたときで、自動アンボクシングで null を数値に展開しようとして落ちます。期待値0で突き合わせるテストは値が合わずに Failures、int で受けるテストは Errors 側に出ます。

/gen-test はこのプロジェクトに同梱したコマンドで、実体は .claude/commands/gen-test.md です。handson フォルダを開いた状態で claude を起動すると、/ge まで打てば候補に出ます。テストが1件でも赤いと Maven は BUILD FAILURE で終わりますが、今日はそれが狙いどおりの状態なので、Results: の下の Tests run と Failures、Errors の行だけ読んでください。

修正後の確認は java src\main\java\com\example\order\buggy\BuggyOrderCalculator.java です。main に自己診断が入っていて、直る前は「バグ発生!」が3か所、直ったあとは割引と遅延が「正常」、存在しない会社の合計が0円と出ます。

休憩

[10min] ここで一度手を止めます。詰まっている方はこの間に声をかけてください。

ねらい

309行の帳票サービスを、出力が変わらないことを毎回確かめながら段階的に読みやすくします。

さわるファイル

書く

exercises/day2/LegacyReportService.java

整理する対象。309行

新規

before.txt

handson 直下。整理前の出力

新規

after.txt

handson 直下。fc.exe で比べる

手順

- 1 実行** VSCode の統合ターミナルで handson フォルダを開き（claude を起動していたら一度抜けます）、`java exercises\day2\LegacyReportService.java > before.txt` を打ちます。中身を開いて「合計件数: 8件」「合計金額(税込): 10,571,000円」の行を確認します。
- 2 書く** claude を起動し、@exercises/day2/LegacyReportService.java の `generateMonthlyReport` のローカル変数名だけを変えるよう頼みます。`d.get("qty")` のようなキー文字列は触らないこと、ロジックは変えないことを条件に入れます。
- 3 実行** claude を抜けて `java exercises\day2\LegacyReportService.java > after.txt` を打ち、`fc.exe before.txt after.txt` で比べます。同じ日のうちなら「相違点は検出されませんでした」が出ます。
- 4 書く** 続けて 1.1 を `TAX_INCLUDED_RATE`、1000000 と 2000000 を `HIGH_VALUE_THRESHOLD` と `URGENT_THRESHOLD` に置き換え、定数の宣言と本文の置換は2回に分けて頼みます。もう一度 after.txt を作って fc.exe で比べ、差が出たらそこが壊した場所なので直前の差分に戻ります。

できたら

- ☐ `fc.exe before.txt after.txt` が、段階を終えるたびに相違点なしを返す
- ☐ `generateMonthlyReport` の `d / q / p / val / tmp / cnt / st / dt` が意味の分かる名前になっている（ループの `i` はそのままでもよい）
- ☐ 1.1 は3メソッド全部、1000000 と 2000000 は `generateHighValueAlert` が定数名に置き換わり、java 実行で合計金額10,571,000円と高額アラート1件が変わっていない

考えること

変数名の置き換えと定数化、先にやるならどちらが安全か。理由を1文で。

AI の出方

頼んでいない範囲まで一度に直してることがあります。段階1で Map のキー文字列まで書き換えられると出力が変わり、fc.exe に差が出ます。「generateMonthlyReport のローカル変数だけ。キー文字列とロジックは触らないで」と範囲を絞り直してください。

D2-3+ Map<String, Object> を OrderRecord クラスに置き換えます。影響範囲が広いので Planモード (Shift+Tab) で計画を先に出させ、OrderRecord の定義と addOrder だけを先に通してから、レポートメソッドを1つずつ移します。1メソッド移すごとに fc.exe で比べると、差が出たときの原因が1か所に絞れます。

くわしく（背景・詰まったときの対処）

LegacyReportService は309行、出力は月次レポートと顧客別サマリーと高額アラートの3本です。動いてはいます。ただ、変数が d, q, p, val, tmp, cnt のまま、税率の 1.1 と高額判定の 1000000 と緊急判定の 2000000 が直接埋まっていて、明細は line = line + ... の連結で組み立てられ、月次と顧客別で集計処理がほぼ丸ごと重複しています。まず出力を固定してください。

java exercises\day2\LegacyReportService.java > before.txt を打つと、月次は合計件数8件・合計金額 10,571,000円・平均金額1,321,375円、顧客別（東京電機）は2件で1,584,000円、高額アラートは1件と出ます。この数字が最後まで変わらなければ、内部をどう書き換えても振る舞いは守れています。比較に使うのは fc.exe。PowerShell で fc とだけ打つと別のコマンドの別名として解釈されるので、.exe まで付けてください。

定数名には注意がいります。1.1 は税率ではなく税込にする倍率なので、OrderServiceImpl にある TAX_RATE = 0.10 と同じ名前を付けると、値の意味が違う定数が2つできます。TAX_INCLUDED_RATE のように別の名前にするか、TAX_RATE = 0.10 を宣言して本文を subtotal * (1 + TAX_RATE) に書き換えるか、どちらかを選んでください。後者に振っても出力は変わらないので、fc.exe でそのまま確かめられます。

置換の範囲も先に決めておきます。1.1 は generateMonthlyReport と generateCustomerSummary と generateHighValueAlert の3か所、1000000 と 2000000 は generateHighValueAlert だけです。変数名のほう は generateMonthlyReport に絞り、ループの i は1文字のままで構いません。

順番は、差分が小さく後ろに響かないものから進めます。変数名、次にマジックナンバーの定数化、余裕があれば文字列連結と共通処理の抽出です。この時間内に型安全なクラスへの置き換えまで届かなくて構いません。1段階ごとに比較を挟む進め方が残れば、この演習で狙ったところは足ります。なお、このファイルは

src/main/java の下にないので `.\mvnw.cmd compile` の対象外です。コンパイルの確認は java の単一ファイル実行が兼ねます。

到達チェック

この日の終わりに、自分で確かめてください。全部埋まっていなくても構いません。

- ☐ 生成されたテストを、そのまま採用せずに指示し直した場面が1回以上あった
- ☐ OrderServiceImplTest の Tests run が配布時の9件から、自分が足した本数ぶん増えるのを確認した
- ☐ BuggyOrderCalculator の3件のバグに対応するテストが、赤から緑に変わるのを自分の画面で見た
- ☐ BuggyOrderCalculator の自己診断から「バグ発生!」の行が消えた
- ☐ fc.exe の比較で、整理の前後に差が出ないことを段階ごとに確認した
- ☐ テストの赤が生成AIの誤りなのか実装の穴なのかを、コードの該当箇所を指して説明できる

つまづいたとき

よく出る詰まりどころです。当てはまるものがなければ、その場で声をかけてください。

コマンドをどこに打つのかわかりません

VSCode の統合ターミナルで handson フォルダを開いた状態で打ちます。Day1 から `.\mvnw.cmd spring-boot:run` を動かしたままでも構いません。その場合はターミナルの + ボタンで2枚目を開き、そちらで打ちます。`.\mvnw.cmd` も `java` も、このフォルダにいないと動きません。claude を起動しているあいだは入力欄が Claude Code のものになるので、PowerShell のコマンドを打つ前に `Ctrl+D` か `/exit` で一度抜けてください。逆に `/gen-test` のようなスラッシュコマンドは、claude を起動した中で打ちます。

BUILD FAILURE と出ます

テストが1件でも赤いと Maven はビルド失敗として終わります。今日は赤を先に出すのが目的なので、Results: の下の Tests run と Failures、Errors の行だけ読んでください。BUILD FAILURE の文字自体は異常ではありません。

生成されたコードが正しいか判断できません

最初に期待値だけ見てください。仕様の値ではなく、いまの実装が返す値で書かれていたら、そのテストはバグごと固定します。ここを外すと後の判断が全部ずれます。ほかは、assert が値を比較しているか、異常系が例外の型まで指定されているか。Java の細部を追わなくても差分の見た目で見分かります。判断がつかないところは採用せず、条件を足して出し直せば済みます。

配布時点で `.\mvnw.cmd test` が赤いです

DELIVERED の受注をキャンセルできてしまう穴が `cancelOrder` に残してあります。`.\mvnw.cmd test` では10件中1件 (`OrderApplicationTests` の1件を含む)、`.\mvnw.cmd test -Dtest=OrderServiceImplTest` なら9件中1件が赤い状態が正常です。Day3 で扱うので、今日は直さずそのままにしてください。環境が動くかどうかだけ確かめたいときは `.\mvnw.cmd compile` を打ちます。テストを走らせないので、この赤に邪魔されずに BUILD SUCCESS が出ます。clean は付けません。アプリを起動したままだと target の削除に失敗することがあります。

BuggyOrderCalculatorTest がコンパイルで止まります

「シンボルを見つけられません」が出ているはずですが、Maven は `src/main/java` の下しかコンパイルしないので、`exercises/day2` に置いたままのクラスはテストから見えません。
`src\main\java\com\example\order\buggy` を作って移してから走らせてください。

fc.exe で出力日の行だけ差が出ます

出力日は実行日で変わるので、日をまたいで比べたときはその行だけ差が出ます。ほかの行に差がなければ振る舞いは守れています。同じ日のうちに比べれば「相違点は検出されませんでした」が出ます。

