

Aufgabe 1

Purja Abdshahzadeh

Tank - und Bezahlvorgang:

Entwurfsmuster : **State Pattern**, weil

1. Klare Phasen im Tankprozess:

Die Anwendung hat **eindeutige Zustände**, die sich klar voneinander unterscheiden:

- **Sorte wählen** (B1)
- **Tankvorgang starten/beenden** (B2)
- **Rabattoption auswählen** (B3)
- **Bezahlvorgang** (B4)

2. Zustandsübergänge logisch abbilden:

- Ein Tankvorgang kann **nicht beendet** werden, ohne dass er vorher gestartet wurde.
- Nach dem Tanken folgt die "Rabattwahl" (optional), dann der "Bezahlvorgang".
Das State Pattern sorgt dafür, dass solche **Übergänge kontrolliert** und **validiert** werden.

Liste der Implementierten Methoden:

int letztesPreisUpdateIstWieAlt(Timestamp t);

- berechnet die Zeit „t - letztes PreisUpdate“ (Ergebnis in sec)

void wartenAufUpdate();

- blockiert System bis neues Preisupdate verfügbar

void waehleSorteAus();

- speichert gewählte Kraftstoffsorte (1-4)

double getanktesVolumenErmitteln();

- ermittelt die aktuell getankte Menge in Litern

void weiterTanken();

- erlaubt Fortsetzung des Tankvorgangs bei < 5L

void bezahlvorgangStarten();

- initiiert den Bezahlvorgang wenn $\geq 5L$ getankt wurden

double EcKartenGebührBerechnen()

- 0,5% Gebühr wird auf den Gesamt Preis berechnet

void waehleRabatt();

- speichert gewähltes Rabattsystem (Payback/ADAC)

double berechneRabattiertenPreis();

- kalkuliert Endpreis mit gewähltem Rabatt

double berechnePreis();

- kalkuliert Endpreis ohne Rabatt

`void pinEingabe();`

- ermöglicht Eingabe der PIN

`void fehlermeldungAnzeige();`

- zeigt Fehlermeldung bei falscher PIN

`void transaktionBestaetigen();`

- schließt Bezahlvorgang erfolgreich ab