?



SEW7 4xHIT 23/24 / Modul 7b: Design-Patterns / [GK] 7b.4: Fast-Food (Observer-Pattern)

[GK] 7b.4: Fast-Food (Observer-Pattern)

✓ Done: View ✓ Done: Make a submission ✓ Done: Receive a grade

In einem Fast-Food-Restaurant soll der Fortschritt abgegebener Bestellungen für die Kund:innen übersichtlich dargestellt werden. Eine Bestellung kann folgende Zustände haben

- Neu: Die Bestellung ist eingelangt, wird aber noch nicht bearbeitet.
 In Bearbeitung: ein:e Mitarbeiter:in hat die Bearbeitung der Bestellung begonnen.
- Bereit: Die Bestellung kann von dem/der Kund:in abgeholt werden.
 Abgeholt: Die Bestellung kann aus den Systemen entfernt werden.

Die relevanten Systeme sind dabei:

- POS-Systeme: POS=Point of Sale; darüber können neue Bestellungen abgegeben werder
- Bestellungs-Anzeigen: darüber werden alle Bestellnummern, die Neu/In Bearbeitung oder Bereit sind angezeigt. Kund:innen können den Unterschied zwischen Neu und In Bearbeitung nicht sehen, und da Bestellnummern chronologisch aufsteigend vergeben werden, sollen diese auch so sortiert dargestellt werden.
- Küchen-Anzeigen: darüber sehen Mitarbeiter:innen die Neuen Bestellungen und können diese zur Bearbeitung auswählen und danach als Bereit markieren. Sobald eine Bestellung in Bearbeitung ist, wird sie hier nicht mehr dargestellt. die Neuen Bestellungen sollen komplett (nicht nur
- Bestellnummern) dargestellt werden, und ebenfalls chronologisch.

 Kassa-Terminals: darüber werden abgeholte Bestellungen als Abgeholt marklert

(Tipp: es kann sinnvoll sein, zusätzlich zu diesen Systemen eine zentrale "Bestellungs-Liste" oder Ähnliches vorzusehen. Ob du so etwas brauchst hängt aber von deinem Design ab und ist dir überlassen.)

Verwende dafür (gemeinsam mit dem EK-Teil) dieses Github-Repository: https://classroom.github.com/a/rNxfjdsh

GKÜ

Erstelle einen Entwurf (UML-Klassendlagramm) das diese Funktionalität umsetzt nach dem Observer-Muster mittels Push- oder Puli-Ansatz. Eine Bestellung soll zumindest eine (fortlaufende) Nummer und einen Preis haben. Implementiere die entsprechenden Klassen in Java.

Erstelle in einer main-Methode Tests, die realistische, nicht komplett triviale Abläufe testen. Es soll zumindest folgendes vorkommen:

- Bei mehreren Bestellungen wird in ankommender Reihenfolge mit der Bearbeitung begonnen, aber die Bestellungen werden in einer anderen Reihenfolge als Bereit markiert.
 Mehreren Bestellungen werden in ankommender Reihenfolge als Bereit markiert aber in einer anderen Reihenfolge Abgeholt.

GKV

sich durch diese Anpassung so wenig wie möglich ändern.

Ändere und erweitere den UML-Entwurf und die Implementierung entsprechend

Statt den Code in der Main-Methode zu testen und die Ergebnisse immer manuell überprüfen zu müssen, sind automatisierte Tests mit JUnit natürlich viel besser geeignet. Setze deine Tests auf diese Weise um. Testprotokoli ist keines notwendig.

Abgabe & Beurteilung

Trage zur Abgabe einen direkten Github-Link auf den letzten Commit in deinem Repository ein, Dieser Link müsste also so aussehen; https://github.com/TGM-HIT/sew7-2324-fast-food-github-username>/commit/ccommit-id>;

Dein abgegebener Commit muss den Sourcecode sowie das UML als png-Datei beinhalten. Die Beurteilung erfolgt in einem Abgabegespräch.

Edit submission Remove submission

Submission status

Submission status	Submitted for grading	
Grading status	Graded	
Last modified	Wednesday, 17 January 2024, 12:43 PM	
File submissions	Bohaczyk_Observer_link 17 January 2024, 12:43 PM	
Submission comments	Comments (0)	

Feedback

Grade	vollståndig erfüllt
Graded on	Wednesday, 17 January 2024, 1:00 PM
Graded by	RW Rateiner-Magor Walter
Foodback comments	

240117: GK