

README

EPR_07_Team

Abgabe für die Teamaufgabe EPR_07
8532653, Hoffmann, 8488705, Singh

1. Analyse:

Das Programm wird objektorientiert entwickelt.

Zu den notwendigen Funktionalitäten gehören:

1. Einlesen einer CSV-Datei, welche die angebotenen Produkte enthält. Diese Informationen sollen dann innerhalb des Programms genutzt werden können.
2. Bestellungen sollen pro Tisch vermerkt werden.
 - unter Einbeziehung von Nachbestellungen und Stornierungen
 - Sonderwünschen sollen berücksichtigt werden
 - wenn etwas hinzugewünscht wird soll sich der Preis um 1€ erhöhen
 - wenn etwas weggelassen wird bleibt der Preis unverändert
3. Erstellen einer Rechnung für den Bezahlvorgang
 - Format der Rechnung soll eine TXT-Datei sein, welche die Bestellungen enthält und den Gesamtpreis

Die bereitgestellte CSV.Datei wird automatisch bei Programmstart eingelesen.

Die Bedienung erfolgt über die Konsole, worüber auch Hinweise zum Status der jeweils gewählten Vorgänge ausgegeben werden. Eingaben müssen auf Korrektheit überprüft werden und fehlerhafte Eingaben dürfen nicht zum Absturz des Programms führen. Zudem wird ein aussagekräftiges UI gewünscht und eine Funktion zum Beenden des Programms.

Weitere Annahmen:

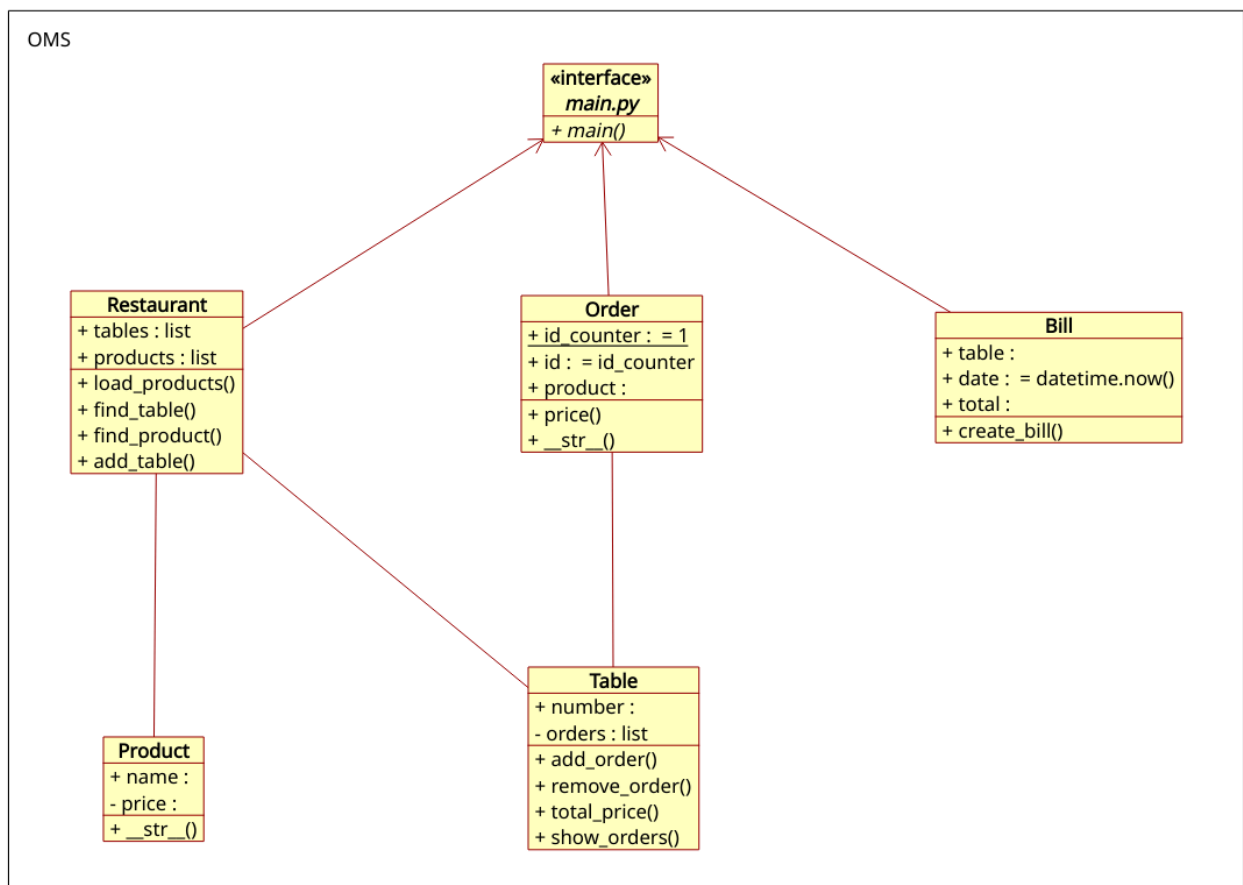
Die Produkte unterscheiden sich eindeutig im Namen.
Abgesehen von den erstellten Rechnungen wird nicht gespeichert.
Also keine Wiederaufnahme der Arbeit bei Neustart. (alle Werte bei Start auf Ausgangszustand)

Zerlegung:

Umsetzung als Klassen, deren Instanzen als Objekte in Listen gespeichert und daraus aufgerufen, bzw. angesteuert werden können.

Jedes Modul enthält für die Übersicht eine einzelne Klasse, die können, wo notwendig, importiert werden.

Alles wird über eine main.py Datei funktional zusammengeführt und kann darüber gestartet werden.



Ausgabe:

Die Konsole, worüber das Programm gesteuert wird, gibt Hinweise zum Status der jeweils gewählten Vorgänge aus.

Die Ausgabe der Rechnungen erfolgt im Programmordner.

Misc Infos:

- Der Gesamtpreis wird auf zwei Nachkommastellen gerundet.
- Keyword "extra" für +1€ bei Sonderwünschen, die etwas hinzufügen
- Der Inhalt der CSV-Datei muss das korrekte Format aufweisen
- Die im Lauf des Programms vergebenen IDs sind fortlaufend
- weitere Funktionen geplant, jedoch nicht bis zur Deadline final umsetzbar

2. Benutzerhandbuch

Nach erfolgreichem Start des Programms werden automatisch die Daten aus food.csv eingelesen.

Dem Nutzer werden anschließend die acht vorhandenen Auswahlmöglichkeiten in der Konsole angezeigt.

Main Menue

1. Create new table
2. Show tables
3. Remove table
4. Take order
5. Show orders
6. Remove order
7. Create bill
8. Exit OMS

Choose an option [1-8]:

Im Folgenden werden die verschiedenen Optionen im Detail erklärt:

1. Create new table

Mit der Eingabe "1" bekommt man die Möglichkeit eine Tischnummer einzugeben, wodurch im System ein Tisch zu dieser Nummer registriert wird.

Es ist darauf zu achten, dass nur ganze Zahlen erlaubt sind!

Bei einer falschen Eingabe wird man erneut aufgefordert eine Tischnummer anzugeben.

Nach erfolgreichem registrieren des Tisches bekommt man über

die Konsole eine Bestätigung: "Table *number* added."

Anschließend wird man zurück in das Hauptmenü geleitet.

(Aktuell gibt es keine Möglichkeiten mehrere Tische gleichzeitig zu registrieren!)

2. Show tables

Die Eingabe "2" erlaubt es eine Liste anzuzeigen, die alle registrierten und damit belegten Tische enthält.

Anschließend wird man zurück in das Hauptmenü geleitet.

3. Remove table

Mit der Eingabe "3" lassen sich registrierte/belegte Tische entfernen.

(Achtung: Das entfernen von einem Tisch löscht auch die darin aufgenommenen Bestellungen!)

Man wird anschließend dazu aufgefordert die Nummer des zu entfernenden Tisches einzugeben. Das erfolgreiche Entfernen wird anschließend in der Konsole bestätigt: "Table number *number* is no longer occupied."

Anschließend wird man zurück in das Hauptmenü geleitet.

(Aktuell gibt es keine Möglichkeiten mehrere Tische gleichzeitig zu entfernen!)

4. Take order

Die Eingabe "4" erlaubt es eine Bestellung aufzunehmen.

Zunächst wird man dazu aufgefordert die Nummer des Tischen einzugeben, für den eine Bestellung aufgegeben werden soll.

Anschließend kann das bestellte Produkt anhand des Namens eingegeben werden. (Achtung: Es gilt hierbei die Groß- und

Kleinschreibung zu beachten! Außerdem wird vom System der exakte Name verlangt, wie er in food.csv hinterlegt wurde.)

(Aktuell gibt es keine Möglichkeiten mehrere Bestellungen gleichzeitig aufzunehmen! Daher muss "Take order für jede Bestellung einzeln aufgerufen werden.")

Daraufhin bekommt der Nutzer die Möglichkeit sonderwünsche einzugeben. Diese müssen durch "," voneinander getrennt werden.

Soll etwas zur Bestellung hinzugefügt werden ist das Schlüsselwort "extra" einzugeben (e.g. "extra Käse"). Damit wird dem Preis bei der Berechnung 1€ hinzugefügt. Fehlt das Schlüsselwort "extra" so bleibt der Preis des Produktes unverändert (e.g. "ohne Ketchup").

Das erfolgreiche Aufnehmen der Bestellung in der Konsole bestätigt: "Order added to table *number*. Order was recorded."

Anschließend wird man zurück in das Hauptmenü geleitet.

(Hier sind nach aktuellem Stand die Eingabe von mehreren Sonderwünschen gleichzeitig möglich. Müssen wie erwähnt durch "," getrennt werden!)

5. Show orders

Mit der Eingabe "5" können die registrierten Bestellungen an einem Tisch angezeigt werden.

Hierfür verlangt das System anschließend die Tischnummer.

Danach werden alle Bestellungen des gewünschten Tisches in der Konsole nacheinander aufgelistet.

Anschließend wird man zurück in das Hauptmenü geleitet.

6. Remove order

Die Eingabe "6" erlaubt es eine bereits registrierte Bestellung zu entfernen. Dafür muss die Nummer des Jeweiligen Tisches

eingegeben werden an den die Bestellung entfernt werden soll. Daraufhin wird die ID der zu entfernenden Bestellung verlangt. (Achtung: Die IDs sind unabhängig von den registrierten Tischen fortlaufend. Um sicher zu gehen, dass die richtige ID für den jeweiligen Tisch eingegeben wird, empfiehlt es sich zuvor die Funktion "Show orders" zu verwenden. So werden dem Nutzer die Bestellungen mit den dazugehörigen IDs des gewählten Tisches angezeigt.)

Nach Eingabe der ID, der zu entfernenden Bestellung, bekommt der Nutzer über die Konsole eine Bestätigung: "Order *number* has been removed from table *number*."

Anschließend wird man zurück in das Hauptmenü geleitet. (Aktuell gibt es keine Möglichkeiten mehrere Bestellungen gleichzeitig zu entfernen!)

7. Create Bill

Mit der Eingabe "7" lässt sich eine Rechnung erstellen, die alle Bestellungen an einem Tisch enthält, sowie den Gesamtpreis berechnet. Die erstellte Rechnung wird als TXT-Datei abgespeichert. Nach eingabe der gewünschten Tischnummer bekommt der Nutzer über die Konsole die Bestätigung: "Bill saved under *billtablenumber_date and time.txt*

Created and saved bill for table 1."

Anschließend wird man zurück in das Hauptmenü geleitet. (Aktuell gibt es keine Möglichkeiten Rechnungen mehrere Tische gleichzeitig zu erstellen!)

8. Exit OMS

Die Eingabe "8" erlaubt es das Programm zu schließen. Mit der Nachricht "Closing OMS." wird das Programm beendet.

(Achtung: Der Zustand des Programms wird bei Beenden nicht gespeichert. Bei einem erneuten Start wird das Programm vollkommen neu initialisiert.)

3. README

Das Programm „Order-Management-System“ stellt Funktionen für ein einfaches Verwaltungssystem bereit.

Es erlaubt für das Verwalten von Bestellungen pro Tisch in einem Restaurant und kann Rechnungen als TXT-Datei abspeichern

Installation

Installiert sein muss ein Python-Interpreter 3.10 oder neuer mit den Bibliotheken und Support-Programmen der Standard-Installation von www.python.org .

Alle restlichen notwendigen Dateien sind im Ordner OMS enthalten, welcher auf die Datei zum Start des Programms enthält.

Start des Programms

Man starte das Programm „Order-Management-System“ in der für das jeweilige Betriebssystem üblichen Art und Weise, aus der Interpreter-Shell oder in einer IDE, z.B. IDLE.

Weitere Angaben zur Dokumentation siehe 1. Analyse und bei Fragen zur Handhabung siehe 2. Benutzerhandbuch.

bekannte Bugs:

Testfälle sind unter der enthaltenen Datei "testcases.pdf" zu finden.

food.csv

- Die Preise der CSV-Datei müssen als ein für Python zulässigen float eingetragen werden. Andernfalls sorgt eine falsche Hinterlegung des Preises zu Fehlern im Programm.
- Das passende Format innerhalb der CSV-Datei muss eingehalten werden (d.h. Name des Produkts in die erste Spalte und Preis in die vierte.)