**Dokumentation: Mein erster Techstore-Versuch**

Ich fange bei der Dokumentation mal mit dem Resultat an.

Leider war das Projekt ergebnistechnisch ein kompletter Reinfall. Ich habe kein funktionierendes Programm vorzuweisen. Das Einzige, was ich anbieten kann, sind die Schlüsse, die ich da rausziehen konnte:

* Ich habe das Projekt viel zu ambitioniert gestaltet. Für meinen Wissensstand waren das einfach viel zu viele neue Herausforderungen, die mich im Lauf des Projekts sehr frustriert haben und viel Energie gekostet haben. Ich habe etwas Angst bekommen, als ich gehört habe, dass das Projekt zur Einschätzung unserer Tauglichkeit für AE verwendet wird und hatte das Gefühl ich musste damit "beweisen", dass ich das kann.
* Ich habe die Grundkonzepte der Methoden und Klassen nicht so gut verstanden, wie ich das anfangs angenommen habe. Ich hätte einfach vorher entweder noch mehr üben sollen oder das Projekt einfach ein wenig simpler gestalten sollen, damit ich einfach das als zusätzliche Übung der Basics nutzen könnte.
* Ich habe die Planung meines Projekts mit falschen Annahmen begonnen. Beispielsweise bin ich davon ausgegangen, dass ich durch das Erben Eigenschaften einfach von einer Klasse an die andere weitergeben kann und darauf dann alles aufgebaut habe. Kann man wohl doch nicht, so wie ich gedacht habe oder ich kann es einfach noch nicht.
* Ich habe zu viel Zeit für die Planung verbraucht. Hätte ich lieber fürs Wiederholen von Methoden und Klassen nutzen können.
* Ich habe Schwierigkeiten mit Zugriffsoperatoren. Muss ich auch nochmal wiederholen.
* Ich habe mich nicht getraut um Hilfe zu fragen. Dadurch, dass ich am Anfang viel Zeit mit Planung verbracht habe und sehr langsam mit dem Code vorangekommen bin, habe ich das Gefühl gehabt, dass ich nicht genug zum Vorzeigen hatte und mich dadurch nicht getraut habe zu fragen. Außerdem wollte ich erstmal probieren, ob ich das eingeständig hinbekomme und nicht nach jedem kleinen Problem gleich nachfragen.
* Ich habe ein Egoproblem, wenns um neue Sachen geht. Manchmal ist meine Frusttoleranz etwas gering, daher brauche ich einfach manchmal meine Zeit, um neue Konzepte zu verarbeiten und mir einfach die Zeit zu geben, um die Dinge zu lernen. Und wenn etwas einfach nicht funktionieren will, dann kann mich das gerne mal demotivieren.

Ich weiß nicht, ob es Sinn macht an dem Projekt weiterzuarbeiten, so wie es jetzt ist. Ich werde erstmal mein Verständnis der Basics weiter aufbauen und dann das Projekt mal so modifizieren, dass es einfach meinem Level entspricht und damit dann arbeiten. Wenns gut läuft, kann ich dann Stück für Stück neue Teile aus der Planung in das reduzierte Projekt einarbeiten.

Mehr als aus dem Scheitern zu Lernen, kann ich gerade leider nicht anbieten :(

 PS.: Nachdem wir die Info über das AE Jahresprojekt erhalten haben: Ich würde mich über ein kurzer Feedback freuen, ob das Projekt von dem Umfang her (so wie ich es anfangs geplant habe) entsprechend für ein Jahresprojekt wäre. Würde daraus erstmal eine Lite-Version machen, um einfach die Basics zu verinnerlichen und dann Stück für Stück weiter daran arbeiten.

Ein kleiner Techstore mit Konsolenoberfläche:

Bisherige Funktionen, die ich mir überlegt habe:

* Liste der Artikel mit Gliederungen in verschiedene funktionelle Bereiche (kann man gut Inheritance und Polymorphismus einbauen)
* Zugriff als Admin und Kunde
* Möglichkeit zur Nachbestellung als Kunde (das wird Bemerkung beim Admin abgespeichert)
* Zu jedem Artikel kann man sich Informationen ausgeben lassen
* Warenkorb, den man vor der Bestellung nochmal überprüfen und möglicherweise noch bearbeiten kann (weiß noch nicht, wie aufwendig das ist, aber probiere ich mal aus)
* Allgemeine Speicherung von Kundendaten und -bestellungen, um auch beim nächsten Start darauf zugreifen zu können.
* Ausgabe der formatierten Rechnung nach Bestellung
* Möglichkeit einen Rabattcode einzugeben, der dann auf Richtigkeit überprüft wird
* Andere Faktoren, die sich auf den Preis auswirken können (wie Mengenrabatt oder andere Sachen)

Je nach Aufwand und Fortschritt würde ich Funktionen hinzufügen oder nicht implementieren. Vielleicht fallen mir während der Arbeit noch Sachen ein.

Mir ist es aber auf jeden Fall wichtig da ein Paar Befehle auch mit reinzubringen, die wir bisher so vielleicht auch noch nicht gehabt haben.

Natürlich wird so gut es geht mit Klassen und Methoden gearbeitet. Lieber etwas mehr, als zu wenig.

Einzelne Funktionen:

* Anmeldung als Kunde oder Admin
  + Überstruktur mit Loop und if Abfrage für Admin oder Kunde
  + Hier besteht die große Frage mit Berechtigungen für die einzelnen Variablen/Methoden
* Artikeloberklasse, Untergruppen und einzelne Einzelartikel
  + Klassen für: Techsparte, Untergruppe (Kopfhörer, Monitore, Grafikkarten, Fernseher etc…welche Eigenschaften könnte ich noch nutzen?), Artikel (Artikelnummer, Bestand, Preis…),
  + Die einzelnen Artikel erben von ihrer Oberklasse Eigenschaften. Und dieser Ganze Ast befindet sich vielleicht noch in einem großen Ast des Rests des Unternehmens, der im Code nicht direkt angegeben wird.
  + Die Artikelnummern werden der Reihenfolge nach generiert.
* Wenn Bestand Untergrenze erreicht wird, Benachrichtigung beim Admin
  + Vergleich des IstBestands mit SollBestand nach Bestätigung des Kaufs. Hierbei soll ein Vergleich vom Käufer-Branch mit einer Variablen aus dem Admin-Branch stattfinden.
* Eine Möglichkeit Artikel in einen Warenkorb zu machen und das dann lokal zu speichern, möglicherweise auch mit Kundendaten
  + Hier wird die größte Herausforderung sein, die Daten abzuspeichern und wieder zugreifbar zu machen. Als extra könnte man noch Infos wie Lieferadresse hinzufügen, könnte aber aus Komfortgründen einfach zu anstrengend sein da ganze Adressen einzutippen
  + Die andere Herausforderung wäre es den Warenkorb in der Konsole bearbeitbar zu machen. Müsste aber mit mehreren Abfragen möglich sein.
* Es gibt’s Konditionen, die sich auf den Endpreis auswirken, wie Rabattcodes oder Mengenrabatte
  + Hier war die Idee, dass die Rabattcodes semi-randomisiert sind. Sie beinhalten das Schlagwort "rbtt", nach dem im Rabattcode gesucht wird. Entweder ist dieser Teil als ganzer String "rbtt" enthalten oder die Buchstaben sind einfach nur in dem Rabattcode vorhanden. In der Reihenfolge, aber nicht am Stück (bspw. 1**r**lkjh**b**ij**t**l**t**3i). Das wird im Laufe des Projekts entschieden, abhängig von der Gesamtschwierigkeit und -dauer.
* Nach Bestellung wird eine formatierte Rechnung ausgegeben
  + Hier kommt es hauptsächlich auf eine korrekte Formatierung an, dass das in der Konsole verständlich und lesbar aussieht.

Nachdem die Grundfunktionen bestimmt wurden, gilt es als nächstes die Grundstruktur des Codes zu überlegen. Da wird es jetzt eher nach dem Konzept learning by doing stattfinden. Ich werden die einzelnen Unterpunkte Stück für Stück implementieren und im Code versuchen die Funktion zu dokumentieren. Die große Herausforderung ist es, die Loops der einzelnen Aufgaben so zu verknüpfen, dass die Logik intakt bleibt. Und die einzelnen Artikelklassen so zu gestalten, dass es Sinn macht mit Erben und Polymorphismus.

Zu allererst müsste ich feststellen, was von den bisher überlegten Funktionen die wichtigsten sind. Prio Nr. 1 ist auf jeden Fall die Gruppen mit den einzelnen Artikeln zu implementieren. Ohne das macht alles andere keinen Sinn. Darauf folgt eine minimalistische Konsolenoberfläche mit Abfragefunktion vom Nutzer. Ist ja auch primär eine Bestelloberfläche. Preisberechnung und Rechnungsausgabe ist relativ simpel und müsste schnell implementierbar sein. Das dann alles in einem do while Loop, mir der Möglichkeit das Programm zu beenden oder neu zu starten. Alle anderen Funktionen wären nice to have, aber sind anfangs nicht für die absolute Grundfunktion essentiell. Als nächstes hätte ich wohl die Faktoren implementiert, die sich auf die Preisberechnung beziehen, wie der Rabattcode. Andererseits finde ich die Adminfunktion ganz spannend, die den Code gleich am Anfang komplett splitten würde und die Auswirkungen auf die Komplexität des Codes zu beobachten. Außerdem finde ich die Herausforderung eines bearbeitbaren Warenkorbs sehr interessant. Ich würde abhängig von der verfügbaren Zeit entscheiden, ob ich zusätzlich etwas implementiere, was wir so ähnlich schonmal gemacht haben.

Eine große Herausforderung wird auch das korrekte Implementieren von Berechtigungen der Klassen und Methoden. Gerade mit der Kunde/Admin Interaktion scheint es nach einem wichtigen Detail, ich bin aber noch nicht so sicher im Umgang mit ihnen (und der get/set-Thematik).

Als ich mit dem Code anfangen wollte, habe ich gemerkt, dass ich nicht ganz zufrieden bin mit der Unterteilung der Artikel. Um den größten Lernprozess zu haben, würde ich gerne versuchen drei Stufen der Untergliederung zu machen, habe aber bisher nur zwei sinnvolle gehabt und möchte nicht einfach nur so eine zusätzliche hinzufügen. Es sollte thematisch schon passen. Es ist aber auch nicht sinnvoll anzufangen, bevor das konkrete Konzept steht. Hierfür würde sich auch ein Klassendiagramm anbieten, ich bin mir aber noch nicht ganz sicher welche Methoden konkret gebraucht werden, werde daher das Klassendiagramm als Übung vom fertigen Produkt ableiten, bis ich irgendwann vertrauter mit beiden Aspekten bin.

Ich bleibe bisher bei einer simplen Unterteilung, wenn mir im Laufe des Projekts etwas Besseres einfällt, dann kann ich das nachträglich abändern.

Nun zu den Variablen und Methoden der Klassen:

Artikel:

|  |  |
| --- | --- |
| Methoden | Variablen |
| LegeInWarenkorb | Int bestand |
| GibInformationen | String bezeichnung |
| WoZuFinden | Double preis |
| BestandPrüfen | String lagerort (macht eher Sinn in der Obergruppe) |
| Nachbestellen | String artikelnummer |
|  | String artikeldetails |

Artikelobergruppe:

|  |  |
| --- | --- |
| Methode | Variablen |
| Konstruktor | String zuständigerMitarbeiter |
| UmzugInNeuenBereich | String lagerort |
|  |  |
|  |  |
|  |  |
|  |  |

Es wird einige Usereingaben geben. Ich frage mich, ob es Sinn mach, diese in einer Methode zu verpacken und damit Lines zu sparen.

Zuerst definiere ich die nötigen Variablen in den essentiellen Klassen. Ich werde das so vermerken, dass die Funktion der Klassen kurz im Code vermerkt wird und genauere Informationen, falls nötig, in der Dokumentation erscheint.

Ich habe mich für folgende Obergruppen (+ die jeweiligen Eigenschaften) und Einzelartikel entschieden

Monitore:

* Bildschirmgröße
* Refreshrate
* Anwendungsgebiet
* Einzelartikel:
  + 27', 60Hz, Office
  + 32', 120Hz, Grafikbearbeitung
  + 34', 240Hz, Gaming

Da ich mit diesen Variablen nicht rechnen werde, speichere ich sie alle als Strings ab. Bin mir noch nicht sicher, ob es Sinn macht, diesen jeweils separate Variablen zu geben oder es der Einfachheit halber sinnvoller ist als einen string zu definieren.

Grafikkarten:

* Maße
* Leistungsaufnahme
* Anwendungsgebiet
* Einzelartikel:
  + 200x120 mm, 130 W, Arme-Schlucker-Version
  + 300x150 mm, 250 W, AI- und Blockchain-Crusher
  + 450x220 mm, 500 W, Gaming-Monster

Handys:

* Speicher
* Displaygröße
* Einzelartikel:
  + 128 GB, 6'
  + 256 GB, 6,6
  + 1 TB, 7'

Kopfhörer:

* Marke
* Art
* Audiofokus
* Einzelartikel:
  + Sandchor, In-Ear, Klare Höhen
  + Beets by Dr.Bre, On-Ear, Tiefe Bässe
  + Böse, Over-Ear, Reiche Mitten

Falls mehr Zeit vorhanden sein sollte, kann ich eine Variable in einer Oberklasse hinzufügen, die in einer Unterklasse überschrieben wird (mit virtual und override). Mir ist aber bisher noch keine Idee gekommen, wie ich das nicht redundant einbauen kann.

Herausforderung: Ich möchte, dass die Klassen die Parameter an Unterklassen vererben. Ich wollte die Parameter der Klasse zuordnen, aber es sollte ja eher so sein, dass ich die Parameter dann beim Konstruktor verwende. Das war dann auch mein Fehler. Werde jetzt das gleiche versuchen im Konstruktor zu integrieren.

Die Variablen in den Konstruktor zu integrieren und die Klasse als abstrakt zu definieren, jetzt versuche ich die Parameter in den Konstruktor der Unterklasse durch : base() zu integrieren, er erkennt die Variablen aber noch nicht. Ich habe versucht mit die Zugriffe der Variablen und der Klassen zu variieren, aber das hat bisher noch nichts zur Lösung beigetragen. Die Variable vorher mit = 0 zu initialisieren hat auch nicht geholfen.

Nach langem Hin-und-Her und der Suche in unseren Übungsaufgaben und mit ChatGPT habe ich herausgefunden, dass es nicht ausreicht, wenn ich die Variablen mit dem base Befehl übernehme, sondern ich muss sie auch zusätzlich in der Klasse (oder in dem Fall im Konstruktor) initiieren. Jetzt stoße ich aber auf das Problem, dass ich in den untersten Klassen sehr viele Variablen als Parameter habe. Ich habe gehofft, dass das durch den base Befehl vielleicht kürzer und übersichtlicher bleibt. Naja, solange der Code funktioniert, kann ich mich später darum kümmern.

Ich habe in den Unterklassen mehrmals Variablen definiert, die den gleichen Namen haben. Ich muss aufpassen, dass ich da später keine Probleme haben werde.

Ich behalte erstmal alle Klassen in der Programmdatei, trenne sie aber zumindest visuell mit einem Kommentar.

Ich habe geschaut, ob es eine Möglichkeit gibt den Text in der Konsole zentriert auszugeben. ChatGPT gab eine Möglichkeit mit windowWidth, textWidth und setCursorPosition empfohlen. Könnte ich bei Möglichkeit zusätzlich implementieren.

Um eine Oberfläche für die Artikelauswahl zu machen, muss ich vorher eine Liste erstellen, in der alle Artikel vorhanden sind. Hierbei muss ich aber darauf achten, dass diese Liste homogen ist. Hierbei entsteht meine Probleme: Ich habe für die Artikel für unterschiedliche Unterklassen teilweise unterschiedliche Variablen verwendet, was es schwer machen könnte alles in eine Liste abzuspeichern. Eine Lösung wäre es jeweils eine Liste für alle Untergruppen zu machen und dann eine große Liste zu generieren, aus den Werten, die in allen Artikeln vorhanden sind. Ich weiß aber bisher noch nicht so wirklich, wie ich mit Listen umgehe, daher muss ich mich da erstmal einarbeiten.

Um die große Datenbank-Liste zu erstellen, muss ich auf die Listen in den einzelnen Klassen zugreifen. Das bereitet mir derzeit Probleme. Ich habe mit den Zugriffen herumprobiert, habe aber bisher keine Lösung finden können.

Nach langem Ausprobieren habe ich einfach den ganzen Code der Klassen in ChatGPT eingefügt und am Ende hat es gereicht einfach nur die eine Liste als Parameter in der Liste der anderen Klasse einzufügen und das hat das Problem bisher gelöst. Ich bin jetzt aber noch mehr durcheinander, was die jeweiligen Berechtigungen angeht, als davor.

Ich kann die große Liste nicht mit Objekten aus der Klasse Artikel erstellen, weil die Klasse abstrakt ist und damit keine Objekte initiiert werden können.

Die Liste mit einer gemeinsamen Datenbank mit allen Artikeln aber nur mit Bestand, Bezeichnung, Preis und Artikelnummer gebe ich erstmal auf. Ich hätte wahrscheinlich bei der Konzeptualisierung mir ein besseres Format überlegen können, damit es nicht zu so einem Monster am Schluss kommt. Ich würde dann bei der grafischen Darstellung einfach auf die einzelnen Listen zugreifen und nur auf die Eigenschaften, die bei allen vorhanden sind.

Wieder zurück bei der Programmierung der Nutzeroberfläche. Ich merke, dass ich bei jeder Nutzereingabe eine Auffangmechanik für Falscheingaben brauche. Ich habe zwar einen Code mit try int.TryParse und catch, aber das müsste ich dann bei jeder Nutzereingabe hinzufügen und überlege, ob eine Methode dafür Sinn macht. Habe das doch in einer Methode implementiert.

Ich bin mir nicht immer sicher, was wann mehr Sinn macht, eine Klasse oder einfach nur eine Methode.

Nachdem ich nun auch bei den Methoden in Methoden zu kämpfen hatte und mir unsicher war, wie ich Variablen richtig von einer Methode in die andere übergebe. Diese Methoden beziehen sich dann auch noch auf Objekte aus den Klassen, wo ich auch gemerkt habe, dass mir da das Wissen noch nicht ausreicht, um das so zu bewältigen, wie ich das anfangs angenommen habe.

Und dann habe ich auch einfach meine Frustgrenze für den Tag erreicht. Mein größtes Problem beim Programmieren ist mein Ego…

Am Ende des letzten Tages musste ich mir eingestehen, dass die Basis, auf dem ich dieses Projekt aufbauen wollte (das Konzept selbst und wohl auch mein Verständnis der Themen Methoden und Klassen) nicht gut genug waren, um da sinnvoll weiter daran zu arbeiten. Ich werde das Projekt erstmal abbrechen und versuchen das Konzept sinnvoller zu gestalten. Angepasst an meinen eigenen Fähigkeitsstand.