Aufgabe 1 – Design-Muster  
 *-> Erläutern Sie, was man unter einem Design-Muster versteht. Welchem Zwecken dienen sie? Wo liegt der*

*Unterschied zu OOA- und Architekturmustern ?*

Ein Design-Muster ist eine Lösungsvorlage für immer wiederkehrende Probleme. Da man für viele Probleme oftmals die gleichen Entwurfslösungen hat, ist es sinnvoll darüber nachzudenken, welchem Muster ein Problem entspricht und das entsprechende Entwurfsmuster anzuwenden. Insbesondere, wenn sich ein Muster bewährt hat, lohnt es sich, es anderen Entwicklern zur Nachnutzung bereit zu stellen. Während die OOA ihre Analysemodelle auf einer anwendungsbezogenen Ebene erstellt und die Architekturmuster die Software im Groben entwerfen, konzentrieren sich die Design-Muster auf den Entwurf im Kleinen. Es werden insbesondere alle notwendigen Klassenbeziehungen, die in den vorigen Stufen der Entwicklung noch ignoriert wurden /werden mussten, miteinbezogen.

Aufgabe 2 – Adapter (Designmuster)

*-> Beschreiben Sie, was man unter dem Designmuster Adapter versteht. In welchen Situationen sollte man es*

*anwenden?*

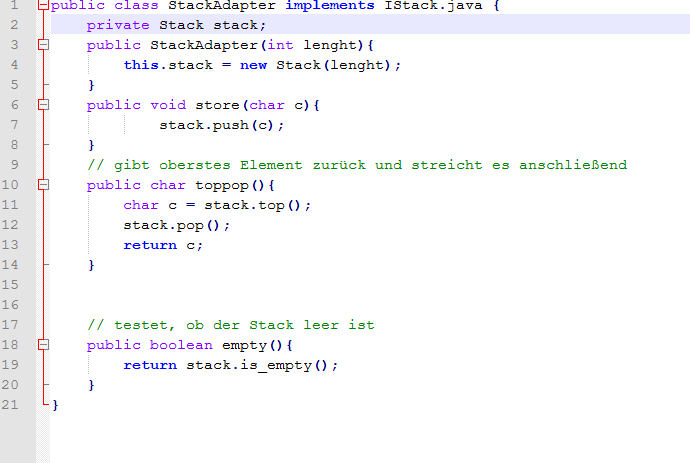
Das Designmuster „Adapter“ hat den Sinn, die Schnittstelle einer Altsystem-Klasse (Adaptierte Klasse) in eine andere Schnittstelle (Adapter) umzuwandeln, die wiederum von der neuen, aufrufenden Klasse erwartet wird. Ziel dabei ist, dass sowohl in der aufrufenden Klasse (Klient) als auch in der Altsystem-Klasse so wenig wie möglich Änderungen gemacht werden müssen, sondern dass alle Anpassungen und Umwandlungen in der neuen Adapterklasse stattfinden. Die aufrufende Klasse und die Altsystemklasse arbeiten zusammen, ohne dass es irgendwelcher Änderungen an ihnen bedarf. Der Adapter arbeitet dabei auch mit den Unterklassen der adaptierten Klasse zusammen und muss für andere Ziele neugeschrieben werden. Unter einem „Ziel“ versteht man hierbei die Schnittstelle, welche der Klient nutzen soll und die vom Adapter realisiert werden.

|  |  |  |
| --- | --- | --- |
| Klient | Ziel | Adaptierte Klasse |
|  | | |

Aufgabe 3 – Stackadapter

*->* *Implementieren Sie die Klasse StackAdapter unter Verwendung der gegebenen Klassen „StringReverse“, „IStack“*

*Und „Stack“*

*.*

Aufgabe Z4 – Strategie (Designmuster)

*-> Erläutern Sie das Designmuster Strategie. Gehen Sie dabei auf die beteiligten Akteure ein. Wann wird dieses*

*Muster eingesetzt?*

Ein Strategiemuster nutzt man um eine Nutzerklasse (Klient) von zur Laufzeit austauschbaren Algorithmen abzukoppeln. Der Klient greift dabei auf einen sogenannten Kontext zu, der die vom Klient gewünschten Operationen durchführt, ohne dass der Klient selbst bemerkt, auf welche Weise dies geschieht. Der Kontext greift dabei auf ein Regelwerk zurück, um zur Laufzeit zu entscheiden, welche Strategie er nutzen möchte, um eine Anfrage des Klienten zu erfüllen. Die konkrete Strategie ist ein Objekt einer von einer Basisklasse „Strategie“ abgeleiteten Klasse, sodass der Kontext lediglich die von der Basisklasse bereitgestellte Algorithmusschnittstelle nutzen und keine etwaigen Besonderheiten der konkreten Strategie bedenken muss.

Eine Einsatzmöglichkeit bietet sich zum Beispiel in einem Kartenspiel, in welchem eine Künstliche Intelligenz „am Zuge“ ist und anhand eines Regelwerks entsprechend entscheidet, wie ihr Zug vollzogen werden soll, die eingehenden Daten wären zum Beispiel die oben liegende (n) Karte(n) und das eigene Blatt, konkrete Strategien wären vielleicht „Bedienen“ oder andere regelkonforme Zugarten.

Aufgabe Z5 – Taschenrechner

*-> Grundrechenarten mit dem Muster Strategie angehen…*

