PROTOKOLL – 08. NOVEMBER 2019

Was ist objektorientierte Programmierung (OOP)?

Die OOP setzt sich mit der Grundidee auseinander, die Architektur einer Software an die Wirklichkeit und die dort existierenden Gefüge/Gegebenheiten anzupassen bzw. sich an dieser zu orientieren. Für OOP charakteristisch ist die Kapselung von Daten und Code in Objekten, welche übersichtliche, wiederverwendbare Module darstellen. Dies reduziert nicht nur den Programmieraufwand, sondern zeichnet die OOP auch als besonders dynamisch aus.

Who is Alan Kay?

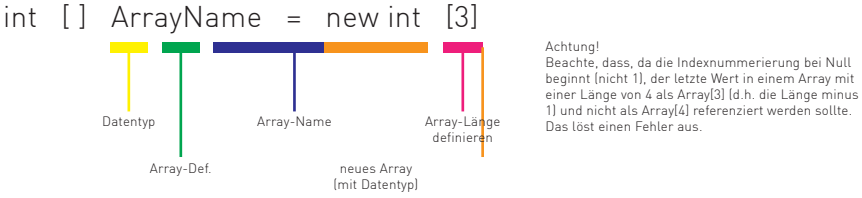
Alan Kay gilt als der Namensgeber für den Begriff „object oriented". Er fasste den Begriff „objektorientierter Programmierung" schlussendlich wie folgt zusammen:

„OOP bedeutet für mich nur Messaging, lokales Beibehalten und Schützen und Verbergen des Prozesszustands sowie spätestmögliche Bindung aller Dinge."

Das ist neu!

1. Arrays

Bei Arrays handelt es sich um einen Satz von Daten, die unter dem gleichen Namen abgelegt werden. Alle Datentypen können in dieser Form gespeichert werden. Jedes Datenstück in einem Array wird durch eine Indexnummer identifiziert, die seine Position in dem Array darstellt. Die Daten werden dabei von Null beginnend bis X durchnummeriert. Jedes Array hat eine variable Länge.



Der Vorteil hierbei ist, dass alle Werte zusammengefasst werden können und man nicht hunderte neue Variablennamen festlegen muss → Programmierer sind faul! Im Sinne der OOP werden Arrays zuerst **deklariert**, dann **erschaffen** und schlussendlich **zugewiesen**.

2. Array-Listen

Neben den Arrays gibt es die sogenannten Array-Listen. Ein Nachteil von Arrays besteht darin, dass man von Beginn an definieren muss, wie groß der zu speichernde Datensatz ist. Eine Erweiterung ist nicht möglich. Array Listen hingegen sind erweiterbar und es können jederzeit neue Ergänzungen angefügt werden. Sie erweisen sich als flexiblere Alternative.

3. Klassen

Unter einer Klasse versteht man in der OOP einen Bauplan bzw. ein abstraktes Modell, mit welchem ähnliche Objekte erschaffen werden können. Ziel ist die Abbildung von realen Objekten in Softwareobjekten mit der Auflistung von Attributen (Eigenschaften) und Methoden (Verhaltensweisen) dieser.

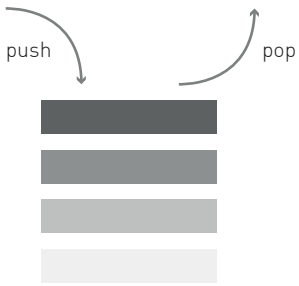
Beispiel: Auto

- was für Eigenschaften hat ein Auto? Größe, Farbe, Form, Marke...
- was für Methoden hat ein Auto? Geschwindigkeit, Startposition...
- dieses Auto kann man nun auf Grundlage des Bauplans beliebig oft neu erstellen

Eine Klasse kann keinen, einen oder mehrere unterschiedliche **Konstruktoren** besitzen. Sie dienen dazu, ein neu gebildetes Objekt einer Klasse in einen definierten Anfangszustand zu versetzen. Im Beispiel des Autos dieses an verschiedene Startpositionen zu setzen oder unterschiedliche Geschwindigkeiten zu definieren.

4. Stacks

Stacks werden auch Stapelspeicher genannt. Sie dienen dazu, Objekte zu speichern und für einen späteren Zeitpunkt aufrufbar zu machen. Zu speichernde Elemente werden übereinander gestapelt und in umgekehrter Reihenfolge vom Stapel genommen. Dies wird auch Last-In-First-Out-Prinzip (LIFO) genannt.



5. PGraphics

Mit den sogenannten PGraphics lassen sich große Zeichendimensionen erschaffen. Diese Klasse wird verwendert, wenn in einen Off-Screen-Grafikpuffer gezeichnet werden muss. Ein PGraphics-Objekt kann mit der Funktion createGraphics() erstellt werden.