

Ziel der Einreichung ist die Realisierung von Microservices. Dazu wurde eine Plattform zum Anbieten mehrerer Anwendungen gewählt. Ziel ist es, die Architektur so zu entwickeln, dass oft genutzte Methoden nicht mehrfach programmiert werden müssen, sondern als Service bereit gestellt werden und mit kleinen Änderungen in übersichtlichen Teilprogrammen das Angebot an die Erfordernisse angepasst werden kann.

1 Dateien zum Starten von Anwendungen

AOP_ausfuehrbar_Anwender.py startet die Anwendung aus Anwendersicht.

AOP_ausfuehrbar_Entwickler.py startet die Anwendung für Wartungszwecke.
Die Anwendung ist passwortgeschützt. Das Passwort lautet 1234.

2 Services zum Aufbau von GUI

AOP_Aufbau_Hauptmenue.py baut das GUI mit dem Hauptmenü auf. Dabei kann sowohl das Hauptmenü für den Anwender als auch das Hauptmenü für den Entwickler mit diesem Programm aufgebaut werden. Der notwendige Parameter wird aus der Datei, mit der die Anwendung gestartet wird, übergeben.

AOP_Aufbau_Anwendung.py baut das GUI für die gewählte Anwendung auf. Dabei ist das GUI für alle Anwendungen vom Grundaufbau her gleich.

AOP_Aufbau_Wartung.py baut das GUI für das gewählte Wartungsziel auf. Auch hier ist der Grundaufbau für alle Wartungsziele gleich.

AOP_Aufbau_Benutzereingabe.py baut ein GUI für die Abfrage einer Benutzereingabe auf.

Die GUI übergeben Benutzereingaben an weitere Programme.

3 Services zur Bereitstellung von grundlegenden oft gebrauchten Funktionen

AOP_Service_Alphabet.py erstellt eine Liste (Array) aller Buchstaben des Alphabets

AOP_Service_Benutzereingabe_einlesen.py liest Benutzereingaben ein und stellt sie als Variable zur Verfügung.

AOP_Service_Fenster_schließen.py enthält die Methode zum Schließen eines GUI. Es wird das jeweils aktive GUI geschlossen.

4 Services zur Bereitstellung von Layoutvorgaben

Die Layoutvorgaben sind Mikroservices, die ein Corporate Design ermöglichen.

AOP_Farben.py legt die farbliche Gestaltung der GUI fest.

AOP_Geometrie.py liefert sowohl Vorgaben für die Maße und Positionen der Elemente der GUI.

AOP_Spielfeld.py liefert Vorgaben für den Aufbau der Spielfelder.

Alle diese Services übernehmen Vorgaben aus anderen Services und stellen Vorgaben für andere Services bereit, sie sind also Schnittstellen.

5 Schnittstellen

AOP_Schnittstelle_Auswahl.py enthält die auswählbaren Ziele der Hauptmenüs. Der übernommene Parameter ist der Anforderer (Anwender oder Entwickler), die übergebene Variable eine Liste der Auswahlmöglichkeiten.

AOP_Schnittstelle_Button_Funktion.py übernimmt die Auswahl des Anwenders und baut die entsprechende GUI auf.

AOP_Schnittstelle_Schrift macht Vorgaben zur Schriftart. Eigentlich gehört diese Datei zu den Layoutvorgaben, aber die Zeit zur Abgabe ist jetzt zu knapp, um das alles noch umzumodeln.

AOP_Schnittstelle_Titel stellt Titel und Überschriften zur Verfügung.

Das bereits im Vorfeld programmierte Spiel TicTacToe könnte in verschiedenen Mikroservices zerlegt werden und so auch als Grundlage für andere Spiele dienen.