

1 Klassenstruktur

1.1 Model

1.1.1 Fassade

Die Fassade kapselt das Komplette Model von dem Viewmodel ab. Sie enthält keine Geschäftslogik, aber verwaltet alle Services. Methoden:

- `+startRunning(*)`
- `+startPushUp(*)`
- `+startSitUp(*)`
- `+stopRunning(*)`
- `+stopPushUp(*)`
- `+stopSitUp(*)`
- `+getSamplingrate()`
- `+setSamplingrate()`
- `+getAccelerator()`
- `+setAccelerator()`
- `+getRecentData(): List<DBEntry>`
- `+getLastData(): DBEntry`
- `+deleteData()`
- `+importData(inputpath)`
- `+exportData(outputpath)`

1.1.2 Dangling

- irgendwas muss sich um die Settings kümmern - irgendjemand muss Schritte zählen
 - Ablauf des Trainingsablaufplans
 - Fassade, die Zugriff auf Services und auf Bibliothek Zugriff hat
 - Dependency Services kann von außen aufgerufen werden ("global", von Xamarin), verwaltet Singletons
- Folgende Services gibt es:

- Bibliothek
- Erweiterungsmodul
- Datenmodul
- TTS (Text-To-Speech)

1.1.3 Bibliothek

Bis jetzt unstrukturiert, schnittstellen noch unklar... Bestandteile der Bibliothek:

- get/set Parameter (konstanten)
 - Samplingrate
 - Accelerator/Gyroscope(range/filter)
- Daten aufzeichnen
- Start/Stop Samplingrate
- Daten verpacken in Format per push (Event) (Beobachter) Wann?
 - bei ca. 100 Rohdaten
 - configs

1.1.4 Erweiterungsmodul

Methoden nach außen (EMSchnittstelle):

- **async registerEvent()**
- **async unregisterEvent()**

innen: TaskHandler und Taskklassen (SitUp, PushUp,...)

mit Beobachter: man kann sich Registrieren für Schritterkennung, Liegest. und Situpsevents

Verarbeitung, wenn mindestens ein Listener registriert ist.

interface ActivityEvent:

- StepEvent: wenn ein Schritt gemacht wurde, enthält Frequenz
- WalkingEvent: wenn man anfängt / aufhört zu laufen, enthält ob laufen/stehen

- PushUpEvent: nach PushUp
- SitUpEvent: nach SitUp

1.1.5 Datenmodul

Methoden nach außen: **GetEntries**, **SaveDBEntry** Kl DatabaseConnection Kl DatabaseEntry

1.2 Viewmodel

1.3 View

2 Schnittstellen zwischen Model, Viewmodel und View

Verbindung zwischen Bibliothek und Erweiterungsmodul: