Konzept “GPS-Tracker”

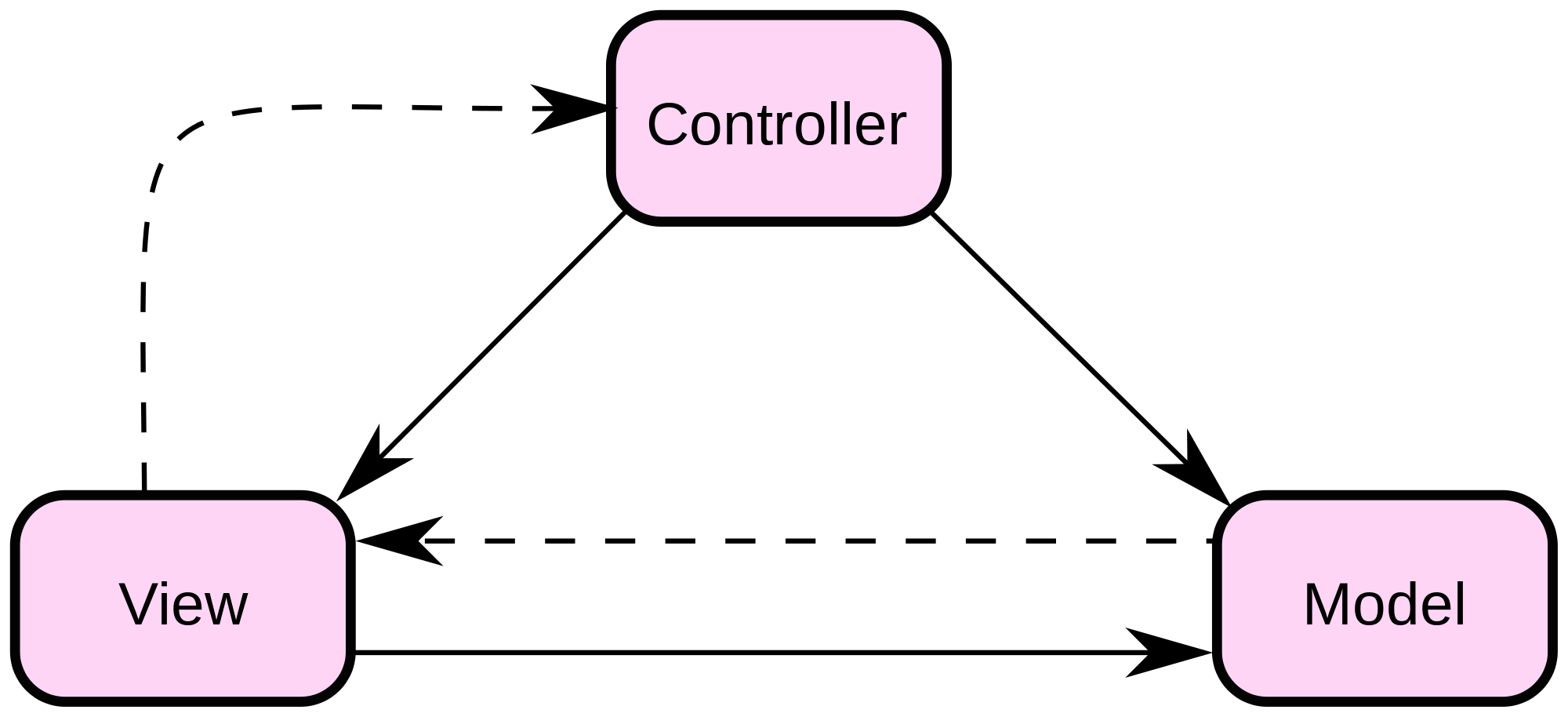
Im Rahmen des Praktikum Software Engineering soll eine Applikation zur Verwaltung von „GPS-Tracks“ erstellt werden. Mittels dieser Applikation ist es Nutzern möglich Ihre gesammelten GPS-Tracks innerhalb einer Applikation zu organisieren, anzusehen und auszuwerten.

# Funktionale Anforderungen

1. Die Applikation soll auf „java-fähigen“ (mindestens Version 15) Computern ausführbar sein.
2. Wenn der Nutzer das Einlesen von Dateien auswählt, soll die Applikation ein Menü zur Auswahl einer Datei öffnen.
   1. Falls der Nutzer die Auswahl abbricht, soll die Applikation das Einlesen abbrechen.
   2. Wenn der Nutzer eine Datei gewählt hat, soll diese Datei von der Applikation geparst werden.
      1. Falls eine Datei gewählt wird, welche sich schon in der Applikation befindet, dann soll die Applikation das Einlesen abbrechen und den Nutzer mittels Fehlermeldung informieren. Die Bestimmung erfolgt durch das Vorhandensein einer namentlich identen Datei im Verzeichnis der Applikation.
   3. Wenn die Datei geparst wurde, soll die Applikation folgende Informationen über den darin enthaltenen “GPS-Track” in der Applikation speichern: Name des “GPS-Tracks”, Aktivität, Datum der Datenaufzeichnung, Startzeitpunkt der Datenaufzeichnung und die enthaltenen GPS-Datenpunkte.
      1. Der Name des “GPS-Tracks” ist ident dem Namen der dazugehörigen Datei.
      2. Falls die Aktivität in der Datei nicht angegeben ist, dann soll der Nutzer diese auswählen können.
      3. Wenn die GPS-Datenpunkte geparst werden, soll die Applikation ebenso die damit assoziierte zurückgelegte Distanz und benötigte Zeit in der Applikation speichern.
   4. Falls ein Fehler auftritt, soll die Applikation das Einlesen abbrechen und mittels Fehlermeldung den Nutzer informieren.
   5. Annahme: Es werden nur Dateien im Format .tcx (Schemaversion 2) und .gpx (Schemaversion 1.1) (in Folge als GPS-Dateien gekennzeichnet) akzeptiert.
   6. Annahme: Es werden nur Dateien, welche eine Distanzangabe und Zeitangabe pro GPS-Datenpunkt angeben, akzeptiert.
3. Wenn das Parsen erfolgreich abgeschlossen wurde, soll die Anwendung die dazugehörige Datei in das Verzeichnis der Applikation kopieren.
   1. Falls der Kopiervorgang nicht erfolgreich abgeschlossen wurde, soll die Applikation den daraus eingelesenen “GPS-Track” aus der Applikation entfernen und mittels einer Fehlermeldung den Nutzer informieren.
4. Wenn die Applikation gestartet wird, sollen alle im Verzeichnis der Applikation befindlichen GPS-Dateien eingelesen werden.
   1. Falls das Einlesen nicht erfolgreich abgeschlossen wurde, soll die Applikation den Nutzer mittels Fehlermeldung informieren und die Option ermöglichen das Einlesen erneut zu probieren.
5. Die Applikation soll eine Listenansicht von den einzelnen eingelesenen „GPS-Tracks“ unterstützen.
   1. Während die Listenansicht der einzelnen eingelesenen „GPS-Tracks“ aktiv ist, soll die Applikation pro “GPS-Track” den Namen, das Datum der Datenaufzeichnung, den Startzeitpunkt der Datenaufzeichnung, die zurückgelegte Distanz, die benötigte Zeitdauer, die durchschnittliche Pace und die durchschnittliche Geschwindigkeit anzeigen.
6. Die Applikation ermöglicht das Filtern der “GPS-Tracks” nach deren Aktivität.
   1. Aktivitäten sind Biking, Driving, Flying, Hiking, Running und Skiing.
   2. Wenn der Nutzer eine Kategorie ausgewählt, soll die Applikation applikationsweit nur jene “GPS-Tracks” präsentieren, welche der Kategorie entsprechen.
7. Die Applikation soll eine zweite Listenansicht unterstützen.
   1. Wenn der Nutzer einen einzelnen “GPS-Tracks” in der Listenansicht auswählt, sollen die Applikation die in diesem “GPS-Track” enthaltenen einzelnen aufgezeichneten GPS-Datenpunkte laut Segmentierungsargument (bezogen auf die Distanz) dargestellt werden.
      1. Die Segmentierung kann dabei mittels 1 Meter, 10 Meter, 100 Meter, 400 Meter, 500 Meter, 1000 Meter, 5000 Meter, 10000 Meter, ein Viertel eines Marathons, ein halber Marathon und pro gespeichertem GPS-Datenpunkt erfolgen. Bei Auswahl einer Distanz sollen die GPS-Datenpunkte hinsichtlich ihrer Distanz addiert werden bis das gewünschte Segmentierungsargument überschritten wird. Diese GPS-Datenpunkte bilden dann ein Segment. Dies wird fortgesetzt bis alle GPS-Datenpunkte einem Segment zugewiesen sind. Das letzte Segment muss das Segmentierungsargument nicht erreichen. Während der Applikationsdurchführung ist ein Segmentierungsargument immer aktiv. Das voreingestellte Argument bei Applikationsstart ist GPS-Datenpunkt.
         1. Annahme: Die möglichen Segmentierungsargumente sind fixiert und können nicht erweitert werden.
      2. Wenn der Nutzer einen einzelnen “GPS-Tracks” in der Listenansicht auswählt, sollen folgende Informationen präsentiert werden: Nummer des Segmentes, kombinierte Distanz, kombinierte benötigte Zeit, die durchschnittliche Pace und die durchschnittliche Geschwindigkeit.
      3. Falls ein Fehler auftritt, soll die Applikation die zweite Listenansicht als leer anzeigen.
   2. Während innerhalb der Listenansicht kein Eintrag gewählt worden ist, soll die Applikation die zweite Listenansicht als leer anzeigen.
8. Wenn der Nutzer in dem Tabellenkopf der Listenansicht eine Spalte auswählt, soll die Applikation die darin enthaltene Liste aufsteigend beziehungsweise absteigend sortieren.
9. Die Applikation soll das Entfernen von geladenen „GPS-Tracks“ ermöglichen.
   1. Wenn der Nutzer einen “GPS-Track” zur Entfernung auswählt, soll zuerst die Applikation diesen “GPS-Track” aus der Applikation entfernen und anschließend soll die Persistierung der dazugehörigen Datei ebenso rückgängig gemacht werden.
   2. Falls es ein Fehler auftritt, soll die Applikation mittels einer Fehlermeldung den Nutzer darüber informieren.
10. Die Applikation soll das Gruppieren von „GPS-Tracks“ nach Zeiträumen ermöglichen.
    1. Diese Gruppierung kann dabei täglich (innerhalb von Jahr), monatlich (innerhalb von Jahr), wöchentlich (innerhalb von Jahr) oder jährlich geschehen.
       1. Annahme: Die Gruppierungsargumente sind fixiert und können nicht erweitert werden.
    2. Wenn der Nutzer die Gruppierung auswählt, soll die Applikation die “GPS-Tracks” entsprechend dem Gruppierungsargument gruppieren und in der Listenansicht als Gruppen (siehe Punkt 5) präsentieren.
    3. Wenn der Nutzer die Gruppierung auswählt, soll die Applikation folgende Informationen pro Gruppe in der Listenansicht anzeigen: Zeitraum der Gruppierung, Anzahl der Elemente in der Gruppe, kombinierte zurückgelegte Distanz, kombinierte benötigte Zeit, durchschnittliche Pace und durchschnittliche Geschwindigkeit.
    4. Wenn der Nutzer eine Gruppe in der Listenansicht auswählt, soll die Applikation alle darin enthaltenen “GPS-Tracks” dieser Gruppe in der zweiten Listenansicht anzeigen (siehe Punkt 7). Die anzuzeigenden Informationen pro “GPS-Track” sind dieselben wie bei Punkt 5.1.
    5. Falls ein Fehler auftritt, soll die Applikation die Listenansicht beziehungsweise zweite Listenansicht als leer anzeigen und mittels Fehlermeldung den Nutzer informieren.
11. Die Applikation soll ein Balkendiagramm anzeigen.
    1. Die Einheiten auf der y-Achse können zurückgelegte Distanz, benötigte Zeit, durchschnittliche Pace und durchschnittliche Geschwindigkeit sein.
       1. Wenn der Nutzer eine Einheit für die y-Achse auswählt, soll die Applikation das Balkendiagramm entsprechend dem gewählten Argument anpassen. Während der Applikationsdurchführung ist ein y-Achse-Argument immer aktiv. Das voreingestellt Argument ist zurückgelegte Distanz.
    2. Während die Applikation ausgeführt wird, soll ein Balkendiagramm angezeigt werden, welches sich der Auswahl des Nutzers zur Laufzeit anpasst.
       1. Während sich die Applikation in der Listenansicht ohne Gruppierung befindet, sollen alle “GPS-Tracks”, die sich in der Listenansicht befinden, laut Argument der y-Achse dargestellt werden. Die Namen der “GPS-Tracks” bilden die Inhalte der x-Achse.
       2. Während sich die Applikation in der Listenansicht ohne Gruppierung aber mit ausgewähltem “GPS-Track” befindet, sollen alle Segmente (siehe Punkt 7.1) laut Argument der y-Achse dargestellt werden. Die Nummer der Segmente bilden die Inhalte der x-Achse.
       3. Während sich die Applikation in der Listenansicht mit Gruppierung befindet, sollen alle Gruppierungen laut Argument der y-Achse dargestellt werden. Der Zeitraum der Gruppierungen bilden die Inhalte der x-Achse.
       4. Während sich die Applikation in der Listenansicht mit Gruppierung und ausgewählter Gruppe befindet, sollen die in der Gruppe befindlichen “GPS-Tracks” laut Argument der y-Achse dargestellt werden. Die Namen der “GPS-Tracks” in der Gruppe bilden die Inhalte der x-Achse.
    3. Falls ein Fehler auftritt, soll die Applikation das Balkendiagramm leer anzeigen und mittels Fehlermeldung den Nutzer informieren.
12. Die Applikation soll das Schließen der Applikation ermöglichen.
    1. Wenn der Nutzer die Applikation schließt, soll die Applikation alle offenen Fenster schließen.

# Architektur

Zur Umsetzung dieser Applikation wird Java Version 15 verwendet. Zur Realisierung der graphischen Benutzeroberfläche wird das Framework “JavaFX” in der Version 18 verwendet. Die Applikation wird mittels einem “MVC-Pattern” realisiert, da “JavaFX” fundamentall dafür strukturiert und es für die Aufgabenstellung geeignet ist. Das “MVC-Pattern” besteht dafür aus einem Modell, einem oder mehreren Views und der Anzahl der Views entsprechende Controllers.

Das Modell enthält die für die Applikation notwendigen Daten und die entsprechende Geschäftslogik. Das sind zum Beispiel Methoden um Elemente hinzuzufügen oder sie auch wieder zu entfernen. In unserem Projekt verwenden wir das Modell um alle “GPS-Tracks” und weitere notwendige Informationen zu verwalten. Zum Interagieren mit dem Modell und dem Liefern von Informationen an die Views wird ein Controller verwendet. Dieser definiert welche Aktion beim Drücken von zum Beispiel einem Knopf auf der View passieren soll. Ebenso bestimmt der Controller, welche Daten aus dem Modell auf der View zu sehen sind. Unser Projekt besteht aus einem Controllern. Dieser Controller übernimmt die Steuerung der Listenansicht, des Diagramms und falls benötigt der zweiten Listenansicht. Die View übernimmt die Darstellung der Interaktionsmöglichkeiten und der Daten. Sie kommuniziert dazu mit dem Controller und interagiert ebenso mit dem Modell. In unserem Projekt interagiert die View mit dem zuvor angesprochenen Modell um die Daten für die Darstellung der Listenansichten und dem Balkendiagramm zu beschaffen.

Durch die Auswahl von “MVC-Pattern” sind die Komponenten der Applikation eine Main-Klasse zum Start der Applikation, ein Modell, ein Controller und eine View.

Die graphische Benutzeroberfläche wird, wie schon angesprochen, mittels “JavaFX” und FXML umgesetzt. Der Vorteil durch den Einsatz von “FXML” ist die Trennung zwischen der graphischen Komponente (View) und der Logik, welche mit der Grafik interagiert (Controller). Dies ermöglicht es die graphische Benutzeroberfläche anzupassen ohne Einflüsse in die Logik vornehmen zu müssen.

# Kernfunktionen (Must-haves)

Folgende Funktionen sind basierend auf den Anforderungen als “must-haves” kategorisiert worden. Die Applikation soll daher folgende Funktionen spätestens mit dem finalen Release umgesetzt haben.

## Start

Die Applikation ist auf Computern, welche Java Version 15 unterstützen, ausführbar. Beim Start der Applikation sollen alle zuvor geladenen „GPS-Tracks“ einzeln in einer Listenansicht wiederhergestellt werden. Dazu muss die Applikation beim Start die zuvor schon geladenen “GPS-Tracks” einlesen. Wenn keine „GPS-Tracks“ vorhanden sind, ist die Liste leer. Das Diagramm in der Applikation soll nach dem Start die zurückgelegte Distanz pro Monat anzeigen. Die detaillierte Listenansicht ist leer nach dem Start der Applikation.

## Einlesen und Persistieren

Die Applikation unterstützt das Einlesen und Persistieren von neuen „GPS-Tracks“, welche im Format „.tcx“ (Schema Version 2) und „.gpx“ (Schema Version 1.1) vorliegen. Diese Dateien sollen dabei vom Programm ausgelesen und die relevanten Informationen innerhalb der Applikation gespeichert werden. Relevante Informationen sind dabei der Name, das Datum der Datenaufzeichnung, der Startzeitpunkt, die Kategorie und die einzelnen aufgezeichneten GPS-Datenpunkte des “GPS-Tracks”. Im Rahmen des Einlesens kann der Nutzer die Kategorie, das Datum und den Startzeitpunkt anpassen. Der Name entspricht dem zugehörigen Dateinamen. Die einzelnen GPS-Datenpunkte werden jeweils mit der zurückgelegten Distanz und der benötigten Zeit relativ von der Startzeit oder falls vorhanden dem vorangegangen Datenpunkt abgespeichert. Falls das Einlesen und/oder Abspeichern der „GPS-Tracks“ nicht möglich ist, soll eine Fehlermeldung, die den User über den Fehler informiert, präsentiert werden.

Das Einlesen der “GPS-Tracks” kann dabei entweder einzeln oder per Ordner geschehen.

Löschen

Die Applikation soll es Nutzern ermöglichen “GPS-Tracks” zu entfernen. Diese entfernten “GPS-Tracks” dürfen bei einem erneuten Start der Applikation nicht erneut gezeigt werden.

## Listenansicht

Die “GPS-Tracks” sollen jeweils einzeln mittels einer Liste angezeigt werden. Anzuzeigende Informationen sind der Name, das Datum, die Startzeit, die zurückgelegte Distanz, die benötigte Zeit, die Pace und die Geschwindigkeit.

## Detaillierte Listenansicht

Durch die Auswahl eines einzelnen Tracks soll eine detaillierte Listenansicht dieses Tracks angezeigt werden. Die detaillierte Listenansicht zeigt dabei folgende Informationen pro aufgezeichneten GPS-Datenpunkt dieses “GPS-Tracks” an: Reihenfolgenummer, die zurückgelegte Distanz, die benötigte Zeit, die Pace und die Geschwindigkeit.

## Gruppieren

Der Benutzer kann die “GPS-Tracks” nach Zeiträumen gruppieren. Diese Gruppierungen sind jeweils auf Basis von Wochen, Monaten oder Jahren möglich. Bei der Gruppierung sollen in der Listenansicht die Anzahl der “GPS-Tracks” als Spalte angezeigt werden, die Kombination der Distanz und Zeit aller “GPS-Tracks” in der Gruppierung als separate Spalten, die durchschnittliche Pace und die durchschnittlichen Geschwindigkeit der Gruppe. Ebenso sollen die Akkumulation der Höhenunterschiede angezeigt werden.

## Sortieren

Im Rahmen der Listenansicht und detaillierten Listenansicht sollen Sortierungen nach den Werten pro Spalte absteigend oder aufsteigend möglich sein.

Diagrammansicht

Neben der Listenansicht soll ebenso eine Diagrammansicht angezeigt werden. Diese Diagrammansicht ist dabei ein Balkendiagramm, welches je nach der aktuellen Selektion und dem gewählten Vergleichswert (zurückgelegte Distanz, benötigte Zeit, Pace oder Geschwindigkeit) an die gegebenen Werte anpasst. Die Selektion kann dabei keine Selektion sein, wobei das Diagramm in diesem Fall die entsprechenden Werte der Listenansicht widerspiegelt. Bei Selektion eines einzelnen “GPS-Tracks” werden die Werte laut Segmentierung gespiegelt. Bei Selektion einer Gruppierung sollen die Werte der “GPS-Tracks” in dieser Gruppierung verwendet werden.

Filtern

Die Applikation ermöglicht es die Tracks nach ihrem Aktivitätstypen zu filtern. Die möglichen Aktivitäten sind dabei Biking, Driving, Flying, Hiking, Running und Skiing. Diese Filterung gilt dabei sowohl übergreifend über die Listenansicht als auch der Diagrammansicht.

Schließen

Die Applikation ermöglicht es die die Applikation zu schließen.

Zusatzfunktion (nice-to-haves)

Vergleich

Ein Vergleich der “GPS-Tracks” wird durch die Applikation ebenfalls unterstützt. Der Vergleichen von “GPS-Tracks” ist dabei über Jahre möglich. Die Anzahl der zu vergleichenden Jahre ist dabei frei wählbar.

Trackpoint:

* Datum (fix bei tcx & optional bei GPX)
* Zeitpunkte (fix bei tcx & optional bei GPX)
* zurückgelegte Distanz (optional bei tcx & optional bei GPX)
* benötigte Zeit (fix bei tcx (über Zeitpunkte) & optional bei GPX)
* pace (optional bei tcx & optional bei GPX)
* speed (optional bei tcx & optional bei GPX)
* bpm (optional bei tcx & bei GPX leer)
* max bpm (optional bei tcx & bei GPX leer)
* elevation (optional bei tcx & optional bei GPX)

Track:

* Trackname (fix bei tcx & optional bei GPX)
* Datum (fix bei tcx (aus Lap) & optional bei GPX)
* zurückgelegte Distanz (fix bei tcx (über lap) & optional bei GPX)
* benötigte Zeit (fix bei tcx (über lap) & optional bei GPX)
* pace (fix bei tcx (über oben) & optional bei GPX)
* speed (fix bei tcx (über oben) & optional bei GPX)
* bpm (optional bei tcx & bei GPX leer)
* max bpm (optional bei tcx & bei GPX leer)
* elevation (optional bei tcx & optional bei GPX)

GPS-Tracks

Diese Tracks bestehen aus

* Name
* Datum
* Startzeit
* Entfernung
* Dauer
* Geschwindigkeit
* Herzschlag pro Minute
* Maximaler Herzschlag
* Höhe

Grafik

Die Applikation soll die Möglichkeit bieten Tracks mithilfe von Grafiken miteinander zu vergleichen.

Zum Vergleich kann ausgewählt werden:

* Distanz
* Dauer
* Erhebungen
* Geschwindigkeit
* Herzschlag

# Komponente

Working with inconsistent xml files

<https://gist.github.com/bennadel/9759130#file-code-3-cfm-L62>