

Programmentwurf GPX Rechner

Name: Philipp Reichert

Matrikelnummer:1758822

Abgabedatum: 28. Mai 2023

Kapitel 1:Einführung

Übersicht über die Applikation

Der GPX Rechner ist ein Programm zur Auswertung von GPX Dateien. Mit seiner Hilfe können Strecke, Dauer und Höhenprofil geplanter Touren vorhergesagt werden und bereits gegangene Touren ausgewertet werden auf Konsistenz der Geschwindigkeit und Geschwindigkeitsheuristiken für weitere Planungen.

Ein besonderes Feature bei geplanten Strecken ist die Aufteilung dieser anhand wichtiger Punkte die regelmäßig besucht werden müssen, wie etwa Unterkünfte oder Wasserquellen.

Die Benutzung der Applikation erfolgt indem man seine Dateien im Format `dateiname.gpx` in den Projektpfad kopiert. Mit starten der `Main()` Methode wird das Terminal gestartet und man wird aufgefordert einen Befehl einzugeben. Bei falscher Eingabe (bzw. "hilfe") werden alle verfügbaren Befehle aufgelistet.

Wie startet man die Applikation?

Wie testet man die Applikation?

Kapitel 2: Clean Architecture

Was ist Clean Architecture?

Clean Architecture ist der Aufbau von Anwendungen in verschiedenen Schichten, die nach innen hin immer beständiger werden. Äussere Schichten können dabei von inneren Schichten Abhängen, innere jedoch nicht von äusseren.

Analyse der Dependency Rule

Positiv-Beispiel: Dependency Rule

Das Positivbeispiel ist die Klasse Speedprofile. Die einzige Methode mit Abhängigkeiten ist

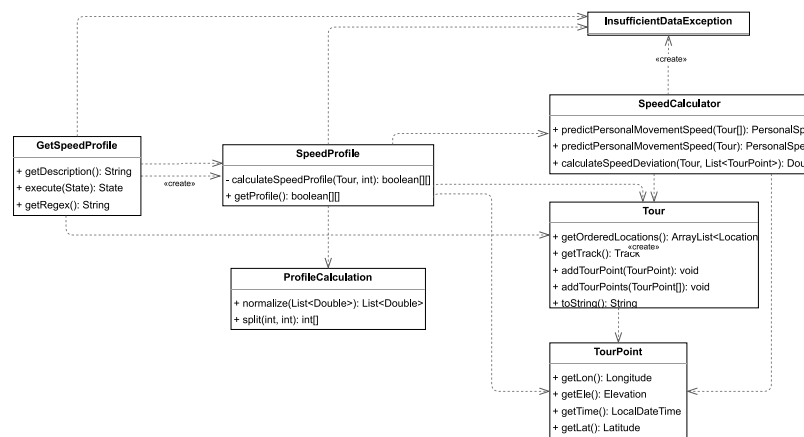


Figure 1: Abhängigkeiten der Klasse Speedprofile

Negativ-Beispiel: Dependency Rule

Schicht: Domain Code

Die Klasse `DistanceCalculator` ist dafür Zuständig verschiedene Distanzen zwischen Orten oder einer chronologischen Abfolgen von Orten im Sexalsystem zu berechnen. Die (zugegeben heuristische) Berechnung von Distanzen im Sexalsystem basieren auf grundlegenden Zusammenhängen und werden sich in absehbarer Zeit nicht ändern und sind Grundlegend für alle Auswertungen von Daten, welche im Sexalsystem abgespeichert sind.

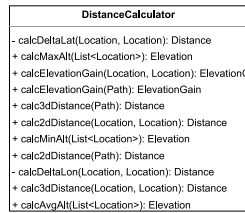


Figure 2: UML Diagramm der Klasse DistanceCalculator

Schicht: Plugins

Die Klasse ConsoleParsing ist dafür Zuständig bestimmte vorgegebene Werte von der Konsole zu lesen. Bei Eingabe unvorhergesehener Werte werden Vorschläge ausgegeben. Die Klasse stellt einen wesentlichen Bestandteil der Benutzerschnittstelle dar. Ein Austausch der Klasse durch beispielsweise ein tolles GUI mit Drop-Downs in Zukunft ist wahrscheinlich und lässt sich entsprechend einfach umsetzen.

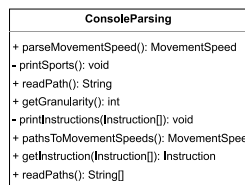


Figure 3: UML Diagramm der Klasse ConsoleParsing

Kapitel 3: SOLID

Analyse Single-Responsibility-Principle (SRP)

Positiv-Beispiel

Latitude
+ Latitude(double):
+ getValue(): double

Figure 4: Abhängigkeiten der Klasse Speedprofile

Negativ-Beispiel

SpeedCalculator
+ calculateSpeedDeviation(Tour, List<TourPoint>): Double
+ predictPersonalMovementSpeed(Tour): PersonalSpeed
+ predictPersonalMovementSpeed(Tour[]): PersonalSpeed

Figure 5: Abhängigkeiten der Klasse Speedprofile

Analyse Open-Closed-Principle (OCP)

Dependency-Inversion-Principle (DIP))