

### Allgemein:

- wollen eine Datenbank mit Orten aus Marburg erstellen
- jeder Ort soll die Attribute Koordinaten, Name, Typ und Zeiteinheit besitzen
- diese Datenbank soll mittels OSM erstellt werden
- basierend auf dieser Datenbank soll ein Tagesablauf für die einzelnen Orte generiert werden
  - das heißt, es soll am Ende möglich sein, eine Liste (oder ähnliches) von Orten auszugeben, an denen sich der Knoten an einem Tag aufgehalten hat

### Gruppe Dustin und Sven:

- haben Pseudo-Datenbanken von Orten in Marburg erstellt
  - diese haben die Attribute Typ, Name, Koordinate, Zeiteinheit, Adresse
    - Zeiteinheit ist abhängig von vom Typ des Ortes
  - diese Datenbanken liegen in verschiedenen Formen vor (Liste von Listen, csv) und können ineinander umgewandelt werden (Funktionen als Input)
- haben basierend auf diesen Datenbanken einen vereinfachten Tagesablauf implementiert
  - zufällige Ausgabe von Orten, an denen sich eine Person am Tag (24 Stunden) aufhält
  - Erster Ort ist immer „Zuhause“
  - Typ „Arbeit“ taucht nur einmal am Tag auf (wie gesagt: vereinfachter Tagesablauf)
  - counter, der anhand der Zeiteinheit der Orte prüft, ob der Ort noch in den Tag passt
  - nach counter==24 ist der Tag vorbei und Programm gibt alle Orte/Koordinaten/Namen/Zeiteinheiten aus
    - dies kann dann noch angepasst werden, je nachdem welche Ausgabe am Ende benötigt wird (Liste, csv, etc.)
- zukünftige Aufgaben: siehe Dokument 2017\_12\_18

### Gruppe Thomas und Ehab:

- haben sich mit OSM auseinandergesetzt und einen Parser für die Daten entwickelt/gefunden
  - dieser liefert eine XML-Datei mit den OSM Daten für Marburg
    - diese ist wiederum sehr groß, sodass die XML Datei „gekürzt“ wird
  - aus der gekürzten XML-Datei wird dann eine csv erstellt
    - mittels dieser csv ist es möglich unsere gewünschten Listen von Orten in Marburg (z.B. Restaurants, Friseure,...) mit den einzelnen oben genannten Attributen (Typ/Name/Koordinaten/Zeiteinheiten) zu erzeugen
      - jeder Ortsknoten in OSM hat eine ID
      - jede dieser IDs besitzt key-value-Einträge, also (key,value)
        - diese werden in OSM auch „tag“ genannt
      - anhand der key-value Werte kann entschieden werden, welchen Typ dieser Ort bekommen soll (z.B. key=shop könnte darauf schließen, dass es sich um einen Friseur handelt; wenn der value dieses keys dann zum Beispiel Friseur ist (also falls (key=shop, value=Friseur) gilt), so kann diese KnotenID als Friseur in unserer Liste gespeichert werden
      - anhand dieser KnotenIDs können dann auch Infos über Name und Koordinaten (oder Öffnungszeiten) abgerufen und zu der Liste hinzugefügt werden

- aus der XML und csv wurde schon ein Dictionary für shops erstellt (shop, name, Koordinate)
  - shop ist dann sowas wie Friseur, Supermarkt, ...
- **Das Programm für den Tagesablauf sollte also mit den so erstellten Datenbanken kompatibel sein**
- zukünftige Aufgaben: siehe Dokument 2017\_12\_18