App, mit der man programmiersprachen lernen kann wie Vokabeln

* Automatisiert informationen ziehen
  + Wie angeordnet?
  + Evtl. vorauswahl an wichtigsten funktionen – z.b. arrays haben viele unwichtige funktionen, aber einige, die man häufig benutzt
  + Vorgehen?
    - Supervised: „lösungen“ aus strukturierten websites holen
    - Unsupervised: (manuell)
    - Naive bayes oder multinomial gauss mit feaure vektoren
* Was sind vokabeln?
  + Konstanten (variablenwerte)
  + Befehlsausdrücke
  + Funktionen (funktionskopf ausfüllen mit Beschreibung), vergleichsoperatoren
  + Objekte
  + Felder
  + Komplexere beispiele, wie list comprehensions

User Needs

* Ich als programmieranfänger möchte die wichtigsten befehlsoperationen auswendig kennen.
* Ich als fortgeschrittener programmierer arbeite grade innerhalb eines packages, was ich auswendig lernen möchte.
* Ich als programmierer möchte nicht jedes vokabular auswendig lernen, nur die wichtigsten.
* Ich als programmierer brauche visualisierungen und eselsbrücken, um die zusammenhänge der objekte zu verstehen.
  + Farben für unterschiedliche objekte
* Ich möchte mir die funktionen als übersicht durchlesen können.
* Ich möchte karteikarten hinzufügen?

System Design

* Datenbank
  + Schema design regeln
    - Prefer embedding, needing to access data on its own is a compelling reason to not embed it
    - Dokument limit in größe von mongodb
    - Arrays should never grow without bound
    - Design the schema based on needs of application
* Packages runterladen? Oder remote ziehen
* Sicherheit
* Skalierbarkeit?

Vergleich: <https://link.springer.com/chapter/10.1007/978-3-030-49435-3_27#Sec3>

* Erstmal mögliche parameter und funktionen kennzeichnen?

Problemstellung BERT:

* Klasse oder funktionsbeschreibung als eingabe. Erkenne:
  + Erkenne funktionsnamen (konstruktordefinition)
  + Methoden der klasse
  + parameter
  + deren beschreibungen und deren beispiele
* input format:
  + müsste einrückungen berücksichtigen, model muss erinnerung haben

TODO

Primary features

* nlp of a package feature
  + scale correctly
  + parse correctly
  + add things (return type,…)
  + error analysis
* improve user experience
  + interviews, iterative

Secondary features

* Downloading new packages from the internet