改良の方法、方向性は自由！

・今回紹介したプログラムの機能拡張をする

・GUIで操作できるようにする

・DBに対応する

・データの取得方法を工夫する（Webスクレイピング）

・Web APIに対応する

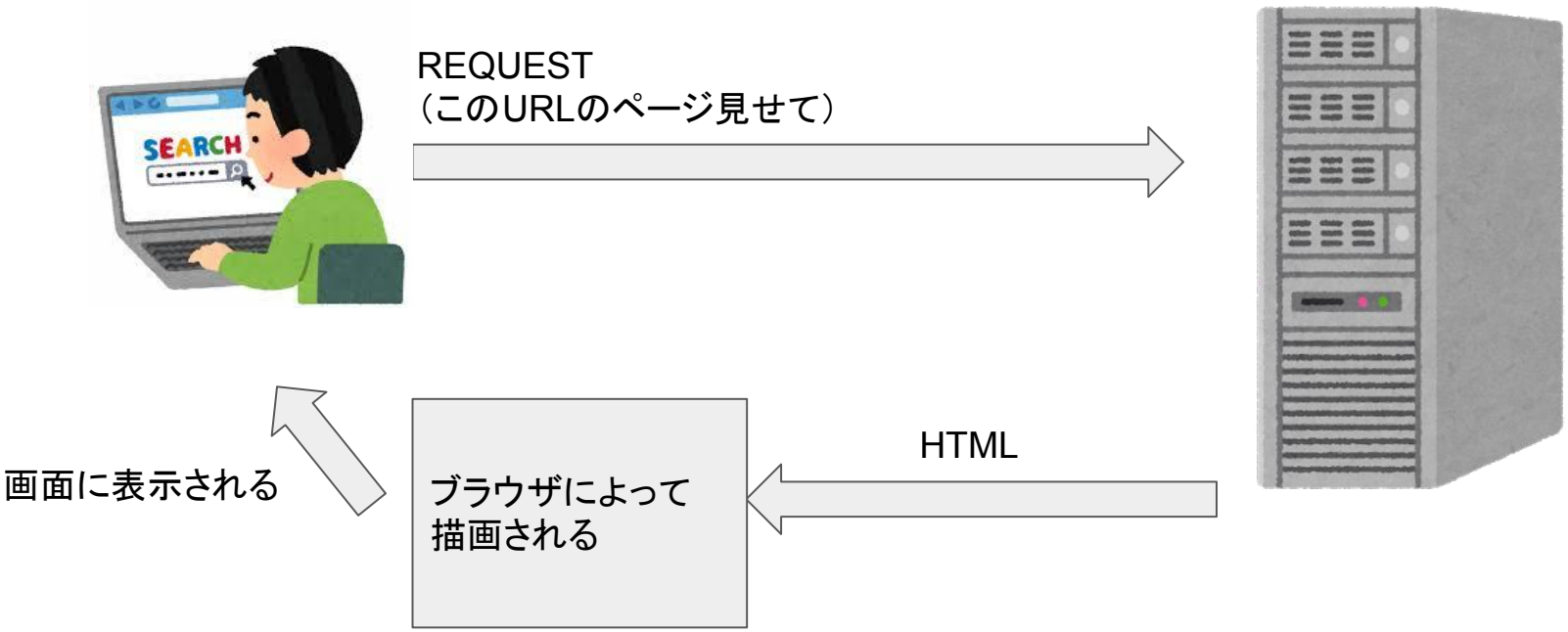
・形態素解析を用いた自然言語処理

・word2vecの学習モデルの精度や対応する単語を増やす

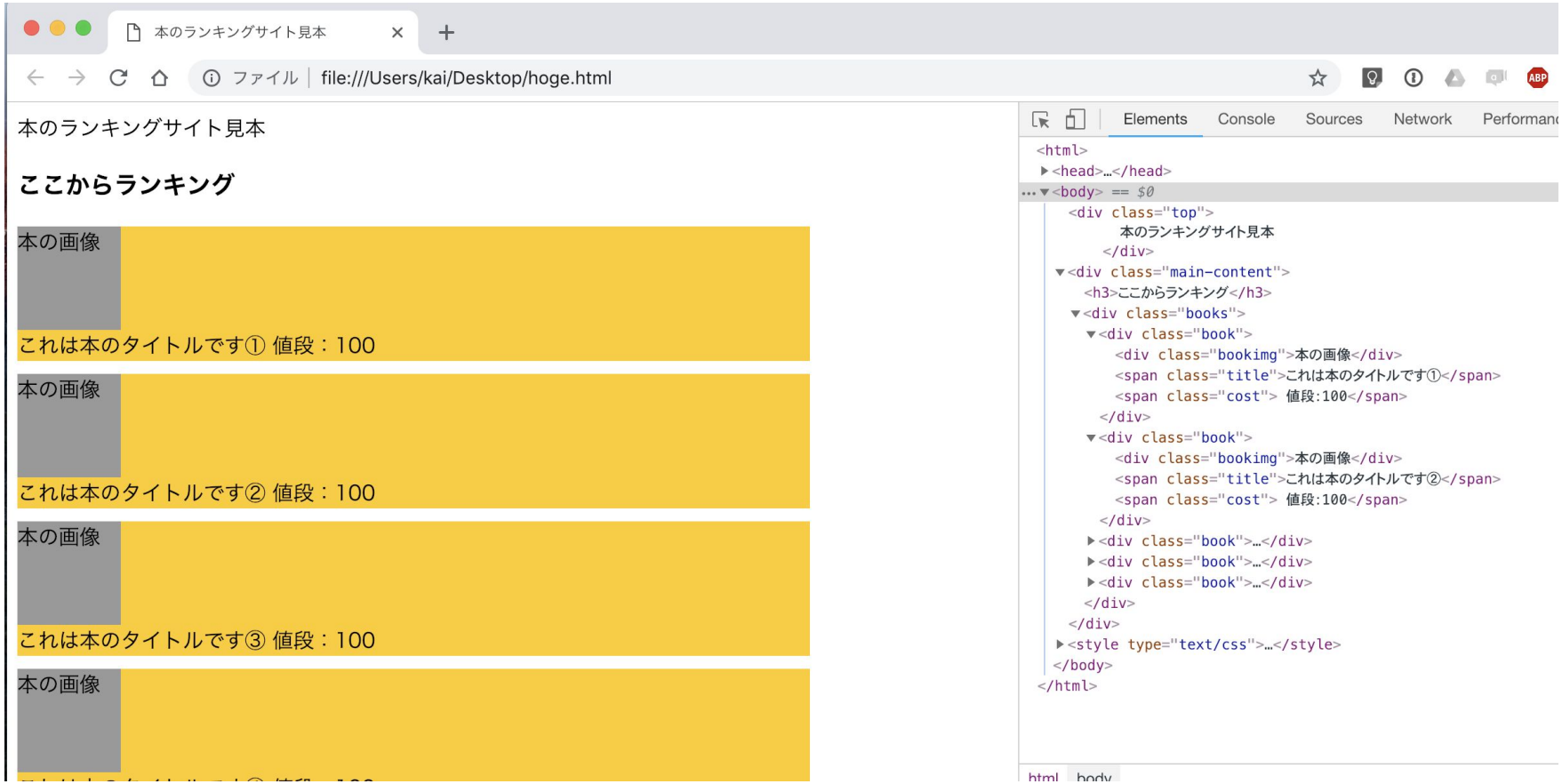
などなど

データの収集自動化（静的サイトのWebスクレイピング）

ブラウザでサイトが開かれるまでの簡略図



HTMLの描画例


本のランキングサイト見本
ファイル | file:///Users/kai/Desktop/hoge.html
本のランキングサイト見本
ここからランキング
本の画像
これは本のタイトルです① 値段：100
本の画像
これは本のタイトルです② 値段：100
本の画像
これは本のタイトルです③ 値段：100
本の画像
Elements
Console
Sources
Network
<html>
<head>…</head>
<body> == $0

本のランキングサイト見本

<h3>ここからランキング</h3>

本の画像
これは本のタイトルです①
 値段:100

本の画像
これは本のタイトルです②
 値段:100

…
…
…

<style type="text/css">…</style>
</body>
</html>
html
body

pythonでWebスクレイピング（静的なサイト）

```
[ ]  from bs4 import BeautifulSoup
     import requests

     url = "https://www.e-hon.ne.jp/bec/SE/Genre?ccode=99&dcode=06"
     response = requests.get(url)
     response.encoding = response.apparent_encoding
     soup = BeautifulSoup(response.text,"html.parser")

     # 本のタイトルを取得
     books = [a.text for a in soup.select("div.rankInner a")]

     #空白文字列を削除する
     books = [''.join(b.split()) for b in books]
     books = filter(lambda x:x != '', books)

     for b in books:
       print(b)
```

```
片づけ・収納・掃除・洗濯の教科書
医者が考案した「長生きみそ汁」
英単語の語源図鑑見るだけで語彙が増える
チコっと冒険チコちゃんに叱られる!ビジュアルファンブックFirst
わけあって絶滅しました。世界一おもしろい絶滅したいきもの図鑑
国家と教養
未だ行ならず下
未だ行ならず上
白魔のクリスマス長編新伝奇小説
IDOLMAKEBIBLE@アカ1
日本国紀
日英語表現辞典
本と鍵の季節
冥界からの電話
童の神
おしりたんていプップッゆきやまのしろいかいぶつ!?
5秒ひざ裏のばしですべて解決壁ドン!壁ピタ!ストレッチ
ブラタモリ15
熱帯
公式TOEICListening&Reading問題集4
```

データの収集自動化（動的サイトのWebスクレイピング）

ブラウザでサイトが開かれるまでの簡略図

REQUEST
（このURLのページ見せて）

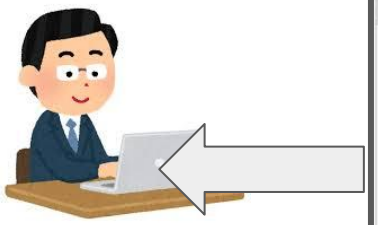
(JavaScriptが埋め込まれた)HTML

・ブラウザによってJavaScriptが実行され、HTMLが書き換えられる

・ブラウザによって描画される

画面に表示される

先程の本のランキングサイトが動的サイトの場合



ブラウザによって
描画される

```
<html>
▶<head>…</head>
▼<body>
    <div class="top">
          本のランキングサイト見本
        </div>
... ▼<div class="main-content"> == $0
      <h3>ここからランキング</h3>
    ▼<div class="books">
      ▼<div class="book">
          <div class="bookimg">本の画像</div>
          <span class="title">これは本のタイトルです①</span>
          <span class="cost"> 値段:100</span>
        </div>
      ▶<div class="book">…</div>
      ▶<div class="book">…</div>
      ▶<div class="book">…</div>
      ▶<div class="book">…</div>
      </div>
    </div>
  ▶<style type="text/css">…</style>
    <script>
          var hoge=0
          ...
        </script>
  </body>
</html>
```

```
<html>
▶<head>…</head>
▼<body>
    <div class="top">
          本のランキングサイト見本
        </div>
... ▼<div class="main-content"> == $0
      <h3>ここからランキング</h3>
      <div class="books">
          </div>
    </div>
  ▶<style type="text/css">…</style>
    <script>
          var hoge=0
          ...
        </script>
  </body>
</html>
```

ブラウザによってJavaScriptが実行され、HTMLが書き換えられる

PythonでWebスクレイピング（動的なサイト）

```
from selenium import webdriver
from selenium.webdriver.firefox.options import Options
url = "https://www.amazon.co.jp/gp/bestsellers/books/ref=zg_bs_books_pg_1?ie=UTF8&pg=1"
browser = webdriver.PhantomJS()
browser.implicitly_wait(3)
browser.get(url)
html_source = browser.page_source
bs_obj = BeautifulSoup(html_source)
books = [a.text for a in bs_obj.select("div.a-fixed-left-grid-col div.p13n-sc-truncated")]
for b in books:
  print(b)
```

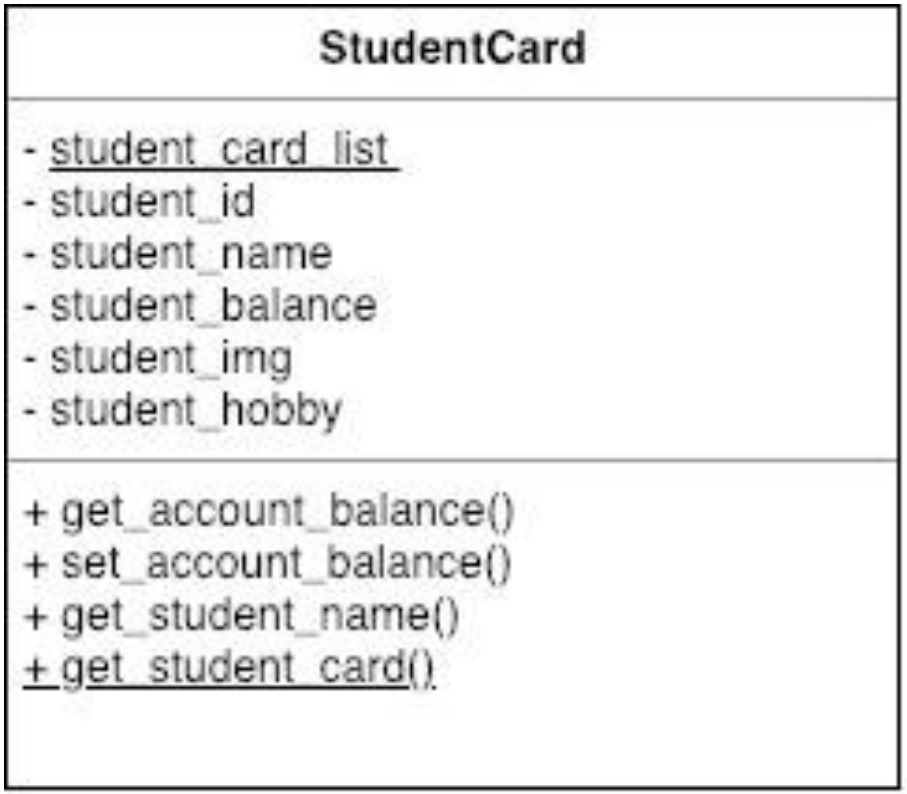
```
/usr/local/lib/python3.6/dist-packages/selenium/webdriver/phantomjs/webdriver.py:49: UserWarning: Sele
  warnings.warn('Selenium support for PhantomJS has been deprecated, please use headless '
KINGDOM HEARTS PERFECT BOOK (バラエティ)
メモの魔力 The Magic of Memos (NewsPicks Book)
一切なりゆき 樹木希林のことば (文春新書 1194)
神様と仏様から聞いた 人生が楽になるコツ
生田絵梨花写真集『タイトル未定』
一人の力で日経平均を動かせる男の投資哲学
Fate/Grand Order カルデアエース VOL.2
日本国紀
「日本国紀」の副読本 学校が教えない日本史
思考の整理学 (ちくま文庫)
医者が考案した「長生きみそ汁」
上馬キリスト教会の世界一ゆるい聖書入門
英単語の語源図鑑
学校の「当たり前」をやめた。― 生徒も教師も変わる！ 公立名門中学校長の改革 ―
わけあって絶滅しました。世界一おもしろい絶滅したいきもの図鑑
ファンタシースターオンライン2 ファッションカタログ2017-2018 LEGACY OF OMEGA
チコっと冒険 First: Eternal Five CHICO チコちゃんに叱られる! ビジュアルファンブック
spoon.2Di vol.45
【Amazon.co.jp限定】鴻池剛と猫のぽんた ニャアアアン! 3 (特典:ぽんた待ち受け画像配信)
キングダム ハーツIII アルティマニア (SE-MOOK)
```

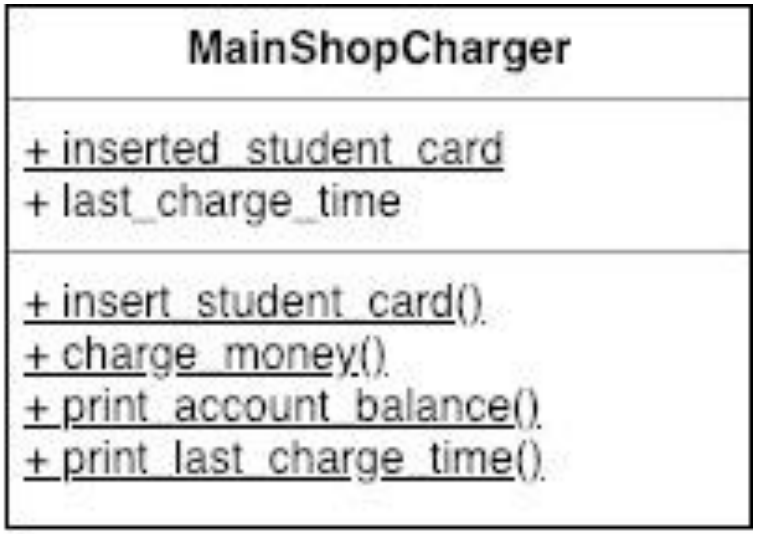
mecabについての説明

```
import MeCab
mecab = MeCab.Tagger("-Ochasen")
mecab.parse('ハリーポッターと賢者の石').split('\n')
```

```
['ハリーポッター\tハリーポッター\tハリーポッター\t名詞-一般\t\t',
'と\tト\tと\t助詞-並立助詞\t\t',
'賢者\tケンジャ\t賢者\t名詞-一般\t\t',
'の\tノ\tの\t助詞-連体化\t\t',
'石\tイシ\t石\t名詞-一般\t\t',
'EOS',
'']
```

現状のクラス

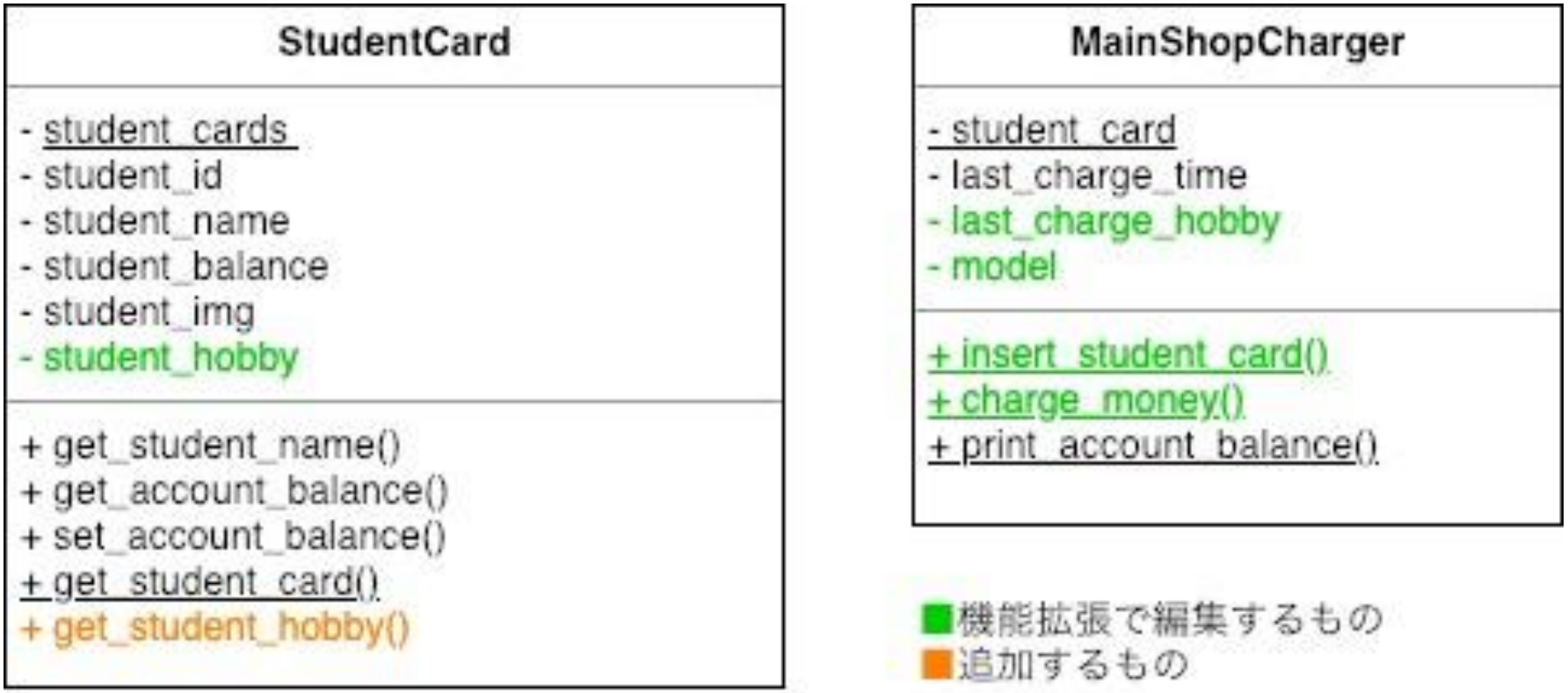

StudentCard
- student_card_list
- student_id
- student_name
- student_balance
- student_img
- student_hobby
+ get_account_balance()
+ set_account_balance()
+ get_student_name()
+ get_student_card()


MainShopCharger
+ inserted_student_card
+ last_charge_time
+ insert_student_card()
+ charge_money()
+ print_account_balance()
+ print_last_charge_time()

改良１：学生証をチャージしたときにおみくじ機能をつける

おみくじ機能の発行条件

直前にチャージした人の趣味と、チャージした人の趣味が近ければ良い結果のおみくじを、遠ければ悪い結果のおみくじを発行する



メソッドの機能

StudentCard

・get_student_hobby()
　→学生のhobbyを返す関数

MainShopCharger

・insert_student_card()
　→StudentCardをチャージしたときに、趣味も保存するようにする

・charge_money()
　→学生証にお金をチャージしたときに、
　　直前にチャージした人と今チャージした人の趣味の類似度を計算し、
　　類似度が高ければおみくじでよい結果を悪ければおみくじの結果を悪くする

実装結果（趣味の類似度算出部分）

```
 @staticmethod
 def charge_money(money):
  kujis = ['凶','末吉','吉','小吉','中吉','大吉']
  omikuji_datas = [{'kuji':k, 'max_score':(2/len(kujis)*i-1)} for i,k in enumerate(kujis, start=1)]

  if MainShopCharger.__inserted_student_card != None:
   MainShopCharger.__inserted_student_card.set_account_balance(MainShopCharger.__inserted_student_card.get_account_balance() + money)
   if not MainShopCharger.__last_charge_hobby:
    print('大吉：今日はじめての利用者です')
   else:
    sim_score = MainShopCharger.__model.wv.similarity(MainShopCharger.__inserted_student_card.get_student_hobby(), MainShopCharger.__last_charge_hobby)
    kuji_kekka = ''
    for d in omikuji_datas:
     if sim_score < d['max_score']:
      kuji_kekka = d['kuji']
      break
    print(kuji_kekka)
  else:
   print('学生証が挿入されていません')
```

実装結果

```
  @staticmethod
  def main():
    student_card1 = StudentCard(0,'tut','サッカー')
    student_card1.set_account_balance(1000)
    student_card2 = StudentCard(1,'tenpaku','ボクシング')
    student_card2.set_account_balance(2000)

    MainShopCharger.insert_student_card(0)
    MainShopCharger.charge_money(1000)
    print('tutくんの状態')
    MainShopCharger.print_account_balance()
    print('')

    MainShopCharger.insert_student_card(1)
    MainShopCharger.charge_money(500)
    print('tutくんの状態')
    MainShopCharger.print_account_balance()

MainShopCharger.main()
```

```
大吉:今日はじめての利用者です
tutくんの状態
残高を表示します
学生名tut
残高2000

中吉
tutくんの状態
残高を表示します
学生名tenpaku
残高2500
```

改良2:学生におすすめの本を、表示する

おすすめの条件
学生の趣味と、販売されている全ての本(今回は100件)のタイトルの類似度を算出し、類似度が一番高い本TOP3を表示する

メソッドの機能

・get_student_hobby()
　→学生のhobbyを返す関数

・insert_student_card()
　→学生証をShopRegisterに差し込む（MainShopChargerと同じ機能）

・recommended()
　→販売されている本の中から学生の趣味に近いものをTOP3を抜粋し、表示

趣味に近い本の算出方法

本のタイトル

⇩ 形態素解析（mecabを使用）

本のタイトル（単語区切りされた本のタイトル）

⇩ 名詞のみ取り出す（mecabを使用）

名詞（本のタイトルに含まれる名詞）

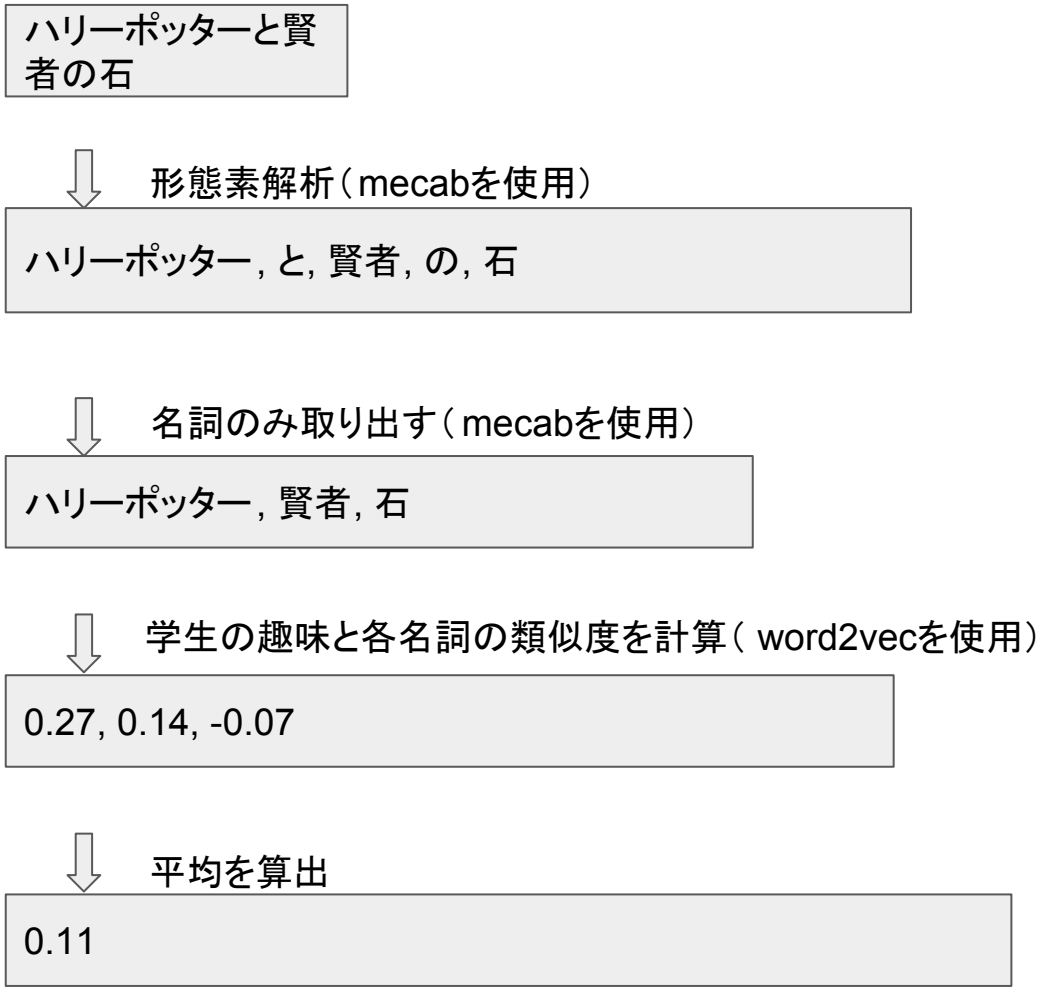
⇩ 学生の趣味と各名詞の類似度を計算（ word2vecを使用）

名詞と学生の趣味の類似度（ word2vecを使用）

⇩ 平均を算出

学生の趣味と本の距離（ =各名詞における類似度の平均）

趣味に近い本の算出方法（例：趣味が”漫画”の時）



実装結果

```
  @staticmethod
  def main():
    shop_reg1 = ShopRegister()
    student_card1 = StudentCard(0,'tut','歴史小説')
    student_card1.set_account_balance(1000)
    ShopRegister.insert_student_card(0)
    shop_reg1.recommended(3)

ShopRegister.main()
```

```
{'title': '東大教授がおしえる やばい日本史', 'cost': 1080, 'score': 0.3446366861462593}
{'title': '漫画 君たちはどう生きるか', 'cost': 1404, 'score': 0.2991243700186412}
{'title': '「日本国紀」の副読本 学校が教えない日本史', 'cost': 950, 'score': 0.2682011895600957}
```