

# 人工智能应用

term project

# 提出について

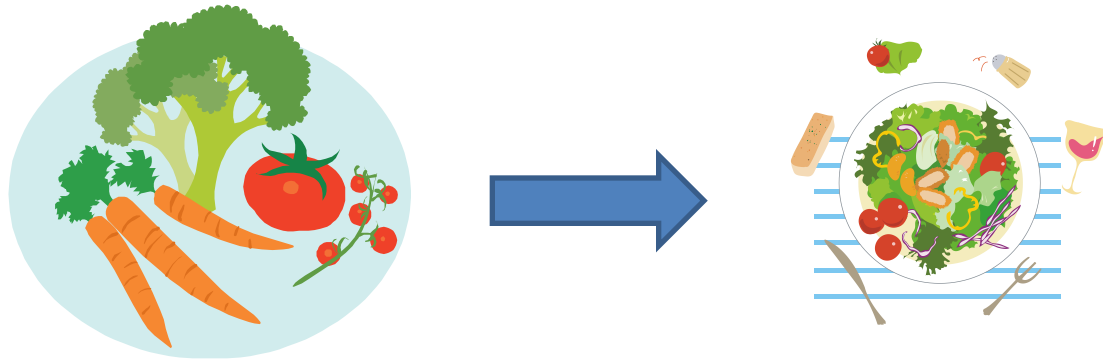
Submit a term-project report including the following items

1. Term project の説明
2. Where you put effort
3. Source program
4. Result screen shot

The deadline is **29th December 2016**

# Smart refrigerator

- 前向き推論を利用して、冷蔵庫内にある材料からおすすめの料理を提案する賢い冷蔵庫を実装してみましょう



# 実装の過程

- 食材(わかっている知識)は10個以上生成
- レシピ(ルール)を5個以上生成



- 好き嫌いによっておすすめメニューをランキング表示する

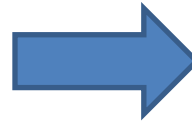


- カロリーや塩分、個人情報(健康状態)からおすすめメニューをランキング表示する

# 食材を10個以上生成する

- 車の例では、プログラムに直接知識を書いていましたが、外部ファイルから読み込むように変更しましょう

```
RuleBase() {  
    fileName = "CarShop.data";  
    wm = new WorkingMemory();  
    wm.addAssertion("my-car is inexpensive");  
    wm.addAssertion("my-car has a VTEC engine");  
    wm.addAssertion("my-car is stylish");  
    wm.addAssertion("my-car has several color models");  
    wm.addAssertion("my-car has several seats");  
    wm.addAssertion("my-car is a wagon");  
    rules = new ArrayList<Rule>();  
    loadRules(fileName);  
}
```



```
my-car is inexpensive  
my-car has a VTEC engine  
my-car is stylish  
my-car has several color models  
my-car has several seats  
my-car is a wagon
```

ヒント: ファイルの読み込みにはBufferedReader等を用いる  
<http://www.javadrive.jp/start/stream/>

# レシピを5個以上生成する

CarShop.dataを参考にしてた  
くさんのレシピを作ってみま  
しょう

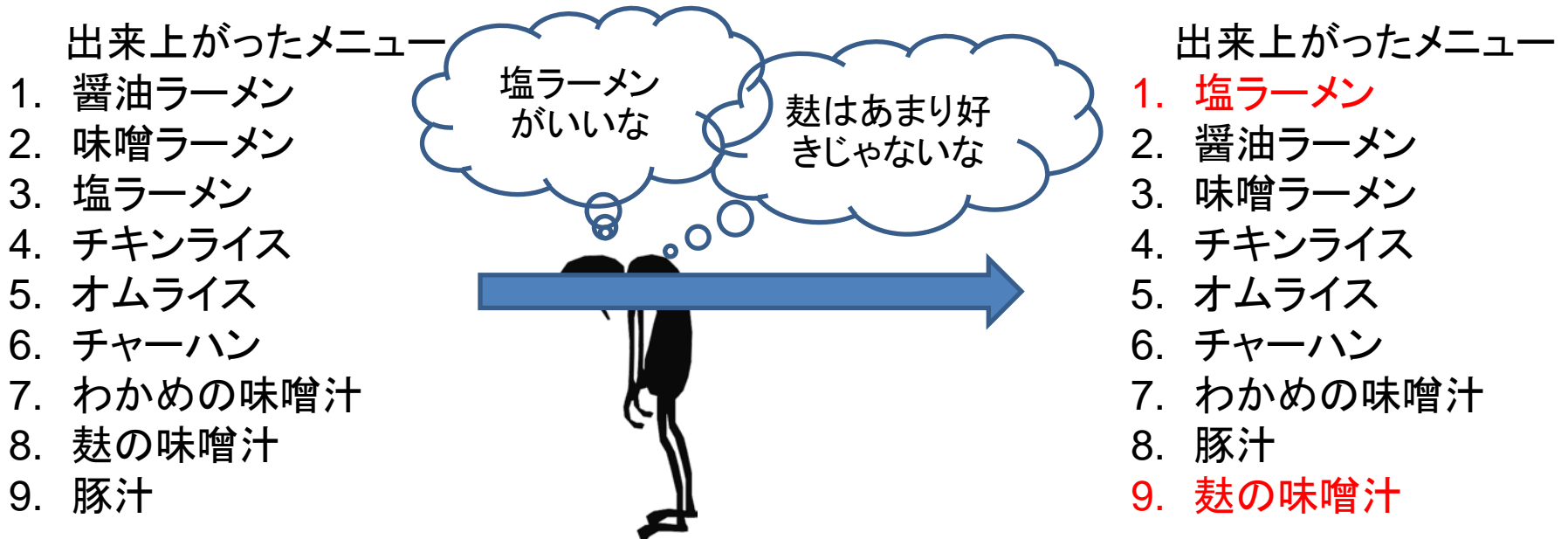
ルールの書き方は自由です

※レシピはクックパッド等を参考にして  
ください

```
rule    "CarRule1"  
if     "?x is inexpensive"  
then   "?x is made in Japan"  
  
rule    "CarRule2"  
if     "?x is small"  
then   "?x is made in Japan"  
  
rule    "CarRule3"  
if     "?x is expensive"  
then   "?x is a foreign car"  
  
rule    "CarRule4"  
if     "?x is big"  
       "?x needs a lot of gas"  
then   "?x is a foreign car"  
  
rule    "CarRule5"  
if     "?x is made in Japan"  
       "?x has Toyota's logo"  
then   "?x is a Toyota"  
  
rule    "CarRule6"  
if     "?x is made in Japan"  
       "?x is a popular car"  
then   "?x is a Toyota"
```

# 好き嫌いによるランキング

- 好きなもの・嫌いなもののリストを自由につくり、それに応じて出来上がったメニューをランキング形式で並び替える



# 料理情報の設定

- 新たにrecipeクラスを作り、料理ごとのカロリーや塩分などの情報を持たせる

```
Class Recipe{  
    String mealName;  
    double calorie,salt;  
    :
```



# ユーザー情報の設定

- 新たにUserクラスを作り、高血圧、ダイエット中などの個人情報を持たせる

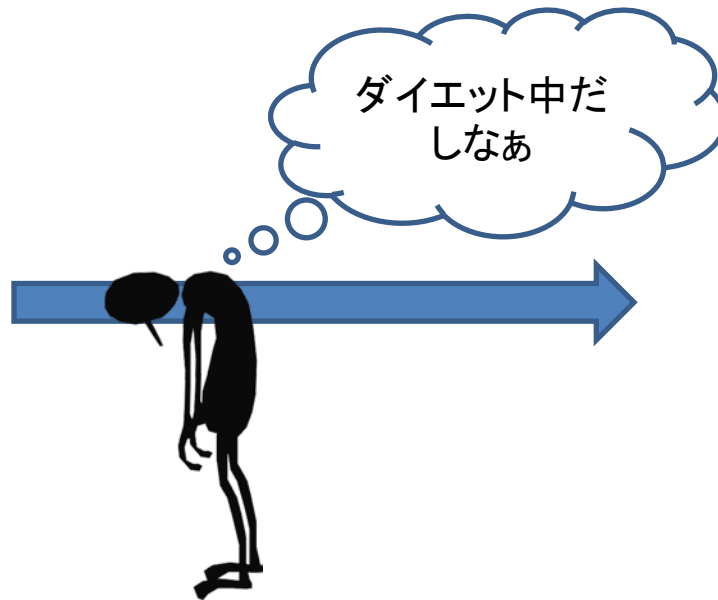
```
Class User{  
    boolean highBloodPressure;  
    boolean diet;  
    :
```

# 個人情報による並べ替え

- 個人情報とメニューの情報を元に、出来上がったメニューを並べ替えてみましょう

## 出来上がったメニュー

1. 醤油ラーメン
2. 味噌ラーメン
3. 塩ラーメン
4. チキンライス
5. オムライス
6. チャーハン
7. わかめの味噌汁
8. 麩の味噌汁
9. 豚汁



## 出来上がったメニュー

1. わかめの味噌汁
2. 麩の味噌汁
3. 豚汁
4. チャーハン
5. チキンライス
6. 塩ラーメン
7. 醤油ラーメン
8. 味噌ラーメン
9. オムライス

# さらに応用

- 余裕のある人は、メニューに“主菜”・“副菜”などの情報を持たせて定食のようなものを提案したり、バランスの取れたメニューを提案するように改良してみましよう