



E&L Fast Cooker



Ce jeu consiste à préparer des commandes dans un fast-food. Il faut servir le plus vite possible afin d'avoir le plus d'argent. (des pourboires supplémentaires. Pour plus de variété on aura trois niveaux: matin, midi, soir. Pour chacun des menus, le client peut commander ou non une boisson, poulet/poisson/Oeuf Bacon, un accompagnement.

Les trois menu en fonction du moment de la journée sont:

Pour le matin:

Café/OeufBacon/Pain

Pour le midi:

Soda/Pouletfrie/Frites

Et pour le soir:

Vin/Poisson/Salade

Classe:

- (Affichage)
- Jeu
- Demandes
- protéine
- boisson (lvl)
- accompagnement

Méthodes:

machine à boisson

poêle

friteuse

emplacement

cuisson(crue, cuit, brûlé)

poubelle

café

soda

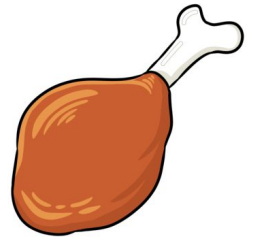
vin

poisson

poulet

bacon

oeuf

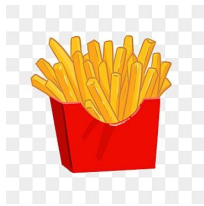


frites
salade
pain

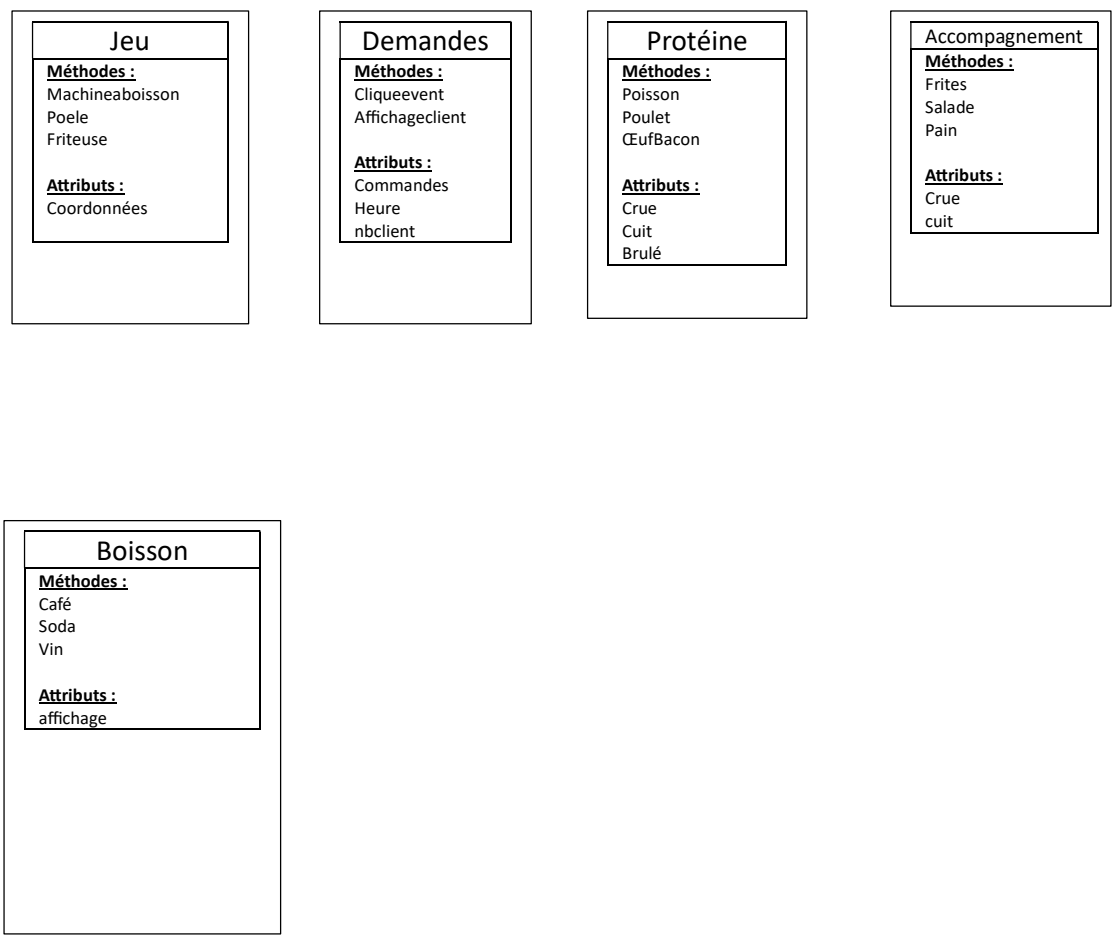
poubelle
cuisson
client
cliques
méthode pour chaque plats

attributs:

cuit
crue
brûler
heure
nbclient
commandes
coordonnées
Idée d' Affichage:



Interfaces



construteur:

- l'attribut nbclient est égale à 0
- l'attribut heure est égale à "matin"
- l'attribut commandes est égale à une table vide

arrivée du client (boisson,proteine,accompagnement):

- affichage client
- tupple est égale (boisson,proteine,accompagnement)
- entrée de sa commande dans la table (file circulaire) sous forme du

tuples (boisson, protéine, accompagnement)

- affichage de la commande

préparation:

- repas est Faux
- boisson est égale à une table composée de None et "boisson"
- proteine est égale à une table composée de None et "proteine"
- accompagnement est égale à une table composée de None et

"accompagnement"

- appeler arrivée du

client(boisson[aleatoire(0,1)],proteine[aleatoire(0,1)],accompagnement[aleatoire(0,1)])

- si heure est égale à matin:

- si boisson n'est pas égale à None:
 - créer une instance de la classe boisson
 - appeler la méthode café
- si protéine n'est pas égale à None:
 - créer une instance de la classe protéine
 - appeler la méthode oeuf bacon
- si accompagnement n'est pas égale à None:
 - créer une instance dans la classe
 - appeler la méthode pain

repas est vrai

- si heure est égale à midi:

- si boisson n'est pas égale à None:
 - créer une instance de la classe boisson
 - appeler la méthode soda
- si protéine n'est pas égale à None:
 - créer une instance de la classe protéine
 - appeler la méthode poulet
- si accompagnement n'est pas égale à None:
 - créer une instance dans la classe
 - appeler la méthode frite

repas est vrai

- si heure est égale à soir:

- si boisson n'est pas égale à None:
 - créer une instance de la classe boisson
 - appeler la méthode vin
 - True[0] est True
- si protéine n'est pas égale à None:

```
        créer une instance de la classe protéine
        appeler la méthode poisson
        repas[1] est True
    si accompagnement n'est pas égale à None
        créer une instance dans la classe
        appeler la méthode salade
        repas[2] est True
si repas = [True,True,True]
    appeler fincommande
si nbclient est > 15:
    appeler fin serviceheure
```

fincommande:

```
    si repas est égale à vrai:
        servir le repas
        depart client
    ajouter 1 à nbclient
    si client
```

finserviceheure:

```
    si heure est égale à soir:
        appeler finjeu
    si heure est égale à midi:
        heure est égale à "soir"
        appeler class Demandes
    si heure est égale à matin:
        heure est égale à "midi"
        appeler class Demandes
```

finjeu:

```
    affichage écran fin
```

