

Дополнительный материал к работе №2

Задание 1. Быстро и запятая

В нашем городе открывается новый ресторан быстрого питания «Быстро и запятая». В фиксированном меню ресторана: булочка, гамбургер, чизбургер, гранд, двойной гранд. Для приготовления каждого блюда требуется свой набор ингредиентов.

Напишите программу, позволяющую ввести список предпочтений клиента и выбрать заказ, который может быть выполнен. Если ни одно блюдо из текущих продуктов приготовить нельзя, то следует выдать сообщение «К сожалению, не можем предложить Вам бургер». Ресторан работает до последнего клиента (завершение программы только по согласию пользователя). При выдаче каждого очередного заказа следует корректировать список продуктов на кухне.

```
print('Введите через запятую сколько сегодня в наличии булочек, котлет, порций сыра,
порций овощей ')
ingredients = list(map(int, input().split(',')))
'''данная конструкция позволяет из строки, содержащей числа, разделенные запятой,
получить список целых чисел'''
menu = ('Булочка', 'Гамбургер', 'Чизбургер', 'Гранд', 'Двойной гранд')
#menu – это кортеж
sostav = {'Булочка': [1, 0, 0, 0], 'Гамбургер': [1, 1, 0, 0], 'Чизбургер': [1, 1, 1, 0], 'Гранд': [1, 1, 2, 1],
'Двойной гранд': [1, 2, 2, 1]}
#sostav – словарь рецептов
print('Добрый день! У нас в меню:')
print('1.Булочка', '\n', '2.Гамбургер', '\n', '3.Чизбургер', '\n', '4.Гранд', '\n', '5.Двойной гранд',
'\n')
repeat = 'да'
#repeat – переменная для организации повторений заказа
while repeat == 'да':
    print('Введите через пробел не более 5 различных чисел от 1 до 5 в порядке убывания
Ваших предпочтений к заказу')
    oder = list(map(int, input().split()))
    #oder – список заказов
    for i in oder:
        probs = sostav[menu[i - 1]]
        flag = True
        #flag – говорит о возможности приготовить бургер
        for j in range(4):
            #проверка, что для бургера хватает ингредиентов
            if ingredients[j] - probs[j] < 0:
                flag = False
                break
        if flag:
            for j in range(4):
                ingredients[j] = ingredients[j] - probs[j]
            #корректировка товаров на складе
```

```

        print('Ваш заказ ', menu[i - 1])
        break
    else:
        print('К сожалению, не можем предложить Вам бургер.')
    repeat = input('Еще заказ (да/нет)? ')
print('Спасибо за заказ!')
```

Задание 2. Быки и коровы

Напишите программу, реализующую логическую игру «Быки и коровы».

Разобьем задачу на подзадачи.

1) Генерация задуманного компьютером числа

```

import random
#импорт библиотеки для работы с генератором случайных чисел
secret = list(range(0, 10))
random.shuffle(secret)
#контроль ситуации нулевого значения первого символа
if secret[0] == 0:
    i = random.randint(1, 9)
    secret[i], secret[0] = secret[0], secret[i]
secret = (''.join(map(str, secret)))
leng = int(input('Введите длину числа не более 10 '))
secret = secret[:leng]
```

2) Ход игры

```

print('Число загадано', '\n', 'Попробуете отгадать (да/нет)?', end = ' ')
repeat = input ()
while repeat == 'да':
    bull, cow = 0, 0
    ans = input()
    if ans == secret:
        print('ПОЗДРАВЛЯЕМ! Число угадано!')
        break
    for i in range(leng):
        if ans[i] == secret[i]:
            bull += 1
        elif ans[i] in secret:
            cow += 1
    print ('Число не отгадано ')
    print('Быков %d Коров %d ' % (bull, cow))
    repeat = input ('Попробуете отгадать (да/нет)?')
print('Спасибо за игру!')
```

Задание 3. Снова расписание

Напишите второй вариант программы «Мое расписание». Продолжаем улучшать наш будущий бот «Расписание».

Предварительная подготовка.

- Создать словарь, который хранит расписание

Основы программирования на Python. Дополнительный материал к работе №2

```
shedule = {'пн': ['история', 'математика'], 'ср': ['право', 'ин.яз']}
```

Подзадачи, которые необходимо решить.

- Научиться добавлять в словарь занятие на заданный день.

#метод append() – добавляет элемент в конец указанного списка

```
repeat = 'да'
while repeat == 'да':
    day_for_add = input('Введите день ')
    subj = input('Введите занятие ')
    if day_for_add in shedule:
        print('Добавлено занятие в существующий день')
        shedule[day_for_add].append(subj)
    else:
        print('Добавлен новый день в расписание')
        shedule[day_for_add] = [subj]
    print(shedule)
    repeat = input('Добавить еще занятие (да/нет)? ')
```

- Научиться записывать расписание в файл и считывать его из файла

```
import pickle
ans = input('Сохранить файл (да/нет?) ')
if ans == 'да':
    with open("out.txt", "wb") as file:
        pickle.dump(shedule, file)
        print('Файл сохранен под именем out', '\n', 'Вывод содержимого файла')
        print('Вывод содержимого файла')
    with open("out.txt", "rb") as file:
        shedule = pickle.load(file)
        print(shedule)
```

- Предусмотреть возможность получения списка занятий на заданный день.

```
day_for_find = input('Выберите день ')
if day_for_find in shedule:
    for i in shedule[day_for_find]:
        print(i)
else:
    print('На этот день нет планов')
```

- Научиться определять, в какой день указанное занятие.

```
flag = 0
subj = input('Введите занятие ')
for i in shedule:
    for j in shedule[i]:
        if j == subj:
            print('День проведения занятия - ', i)
            flag = 1
            break
    if flag == 1:
        break
else:
    print('Занятия нет')
```

```
print('Такая задача не запланирована')
```