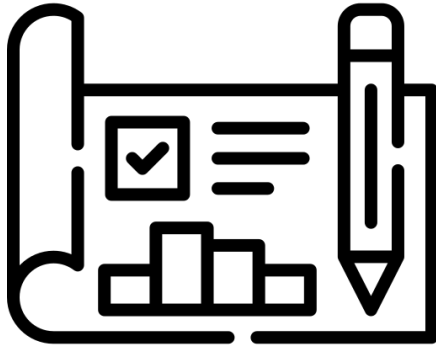




# **P**ersonal menu **R**ecomendation program

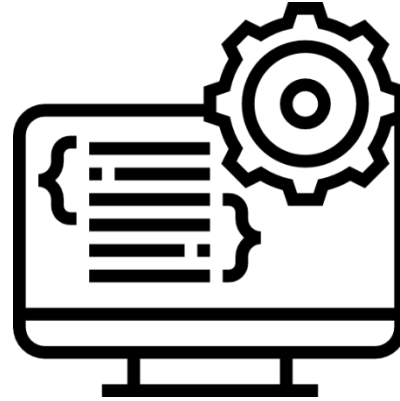
- CAKD3 임희진

## About



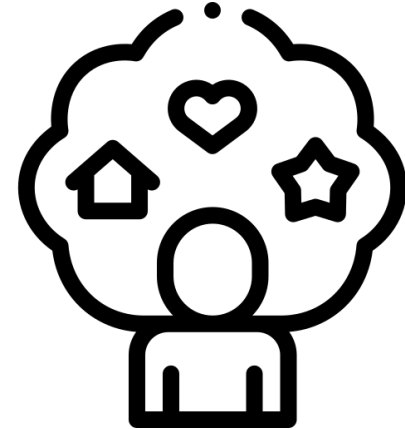
- Program 기획 의도
- Program 사용 설명

## Structure



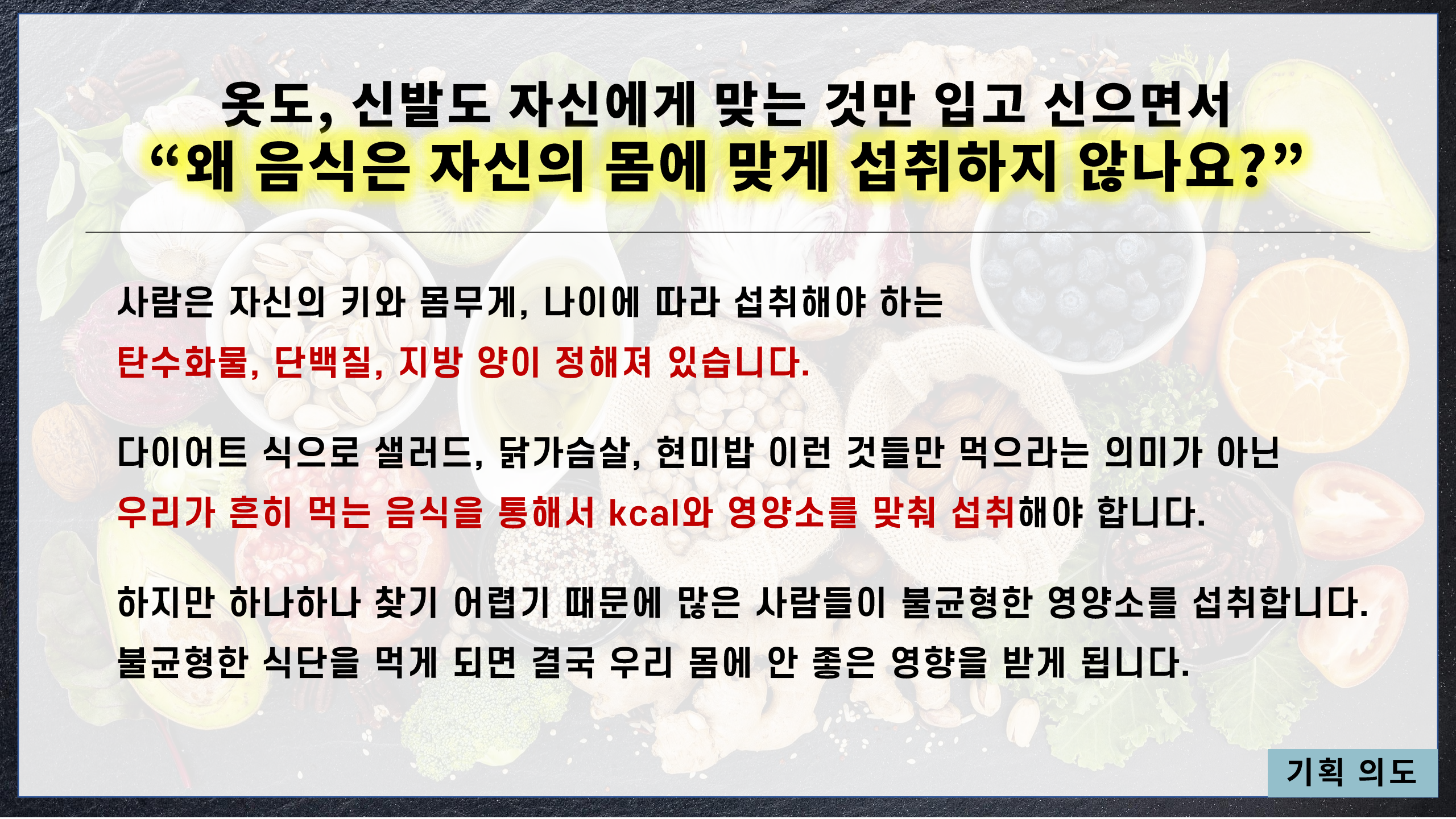
- Program 사용 함수
- Program 구조

## After



- Program 보완점
- Program 기대 효과





# 옷도, 신발도 자신에게 맞는 것만 입고 신으면서 “왜 음식은 자신의 몸에 맞게 섭취하지 않나요?”

사람은 자신의 키와 몸무게, 나이에 따라 섭취해야 하는  
탄수화물, 단백질, 지방 양이 정해져 있습니다.

다이어트 식으로 샐러드, 닭가슴살, 현미밥 이런 것들만 먹으라는 의미가 아닌  
우리가 흔히 먹는 음식을 통해서 kcal와 영양소를 맞춰 섭취해야 합니다.

하지만 하나하나 찾기 어렵기 때문에 많은 사람들이 불균형한 영양소를 섭취합니다.  
불균형한 식단을 먹게 되면 결국 우리 몸에 안 좋은 영향을 받게 됩니다.



따라서 우리 모두가 균형잡힌 건강한 식습관을 만들 수 있도록

**프로그램에 키, 몸무게, 성별, 나이만 입력하면**

**하루 섭취해야될 적정 kcal와 탄수화물, 단백질, 지방 kcal 양을 알려드릴려고 합니다.**

## 프로그램 사용 대상



균형잡힌 건강한 식사를  
하고 싶은 사람



다이어트로 체중을  
감량하고 싶은 사람



당뇨, 암등의 질병으로 인해  
균형잡힌 식사가 필요한 사람



# How to use?



## First.

키, 몸무게, 성별, 나이를 입력해 주세요.  
그리고 평소 활동량과 다이어트 중이신지 입력해주세요.

하루 섭취해야 될 칼로리를 확인해주세요.  
하루 몇 끼를 먹는지 입력해주시면 1끼 적정 섭취 칼로리를  
탄수화물, 단백질, 지방의 비율에 맞춰 알려드립니다.



## Second.



## Third.

취향과 섭취 칼로리에 맞춰 식단을 선택해주세요.  
섭취 칼로리가 부족한지, 많은지 판단해드립니다.

## Class 작성 및 상속

식단 Class와 사용자가 정보 입력을 할 수 있는 Class를 2개 만들어 상속

## Input

사용자가 자신의 정보를 입력 및 다른 선택 사항을 입력할 수 있도록 함.

## While ~ continue ~ break 반복문 및 Try, except

사용자가 입력한 정보가 맞지 않을 때에는 다시 반복 입력,  
정보가 맞을 때는 다음 while문으로 넘어갈 수 있도록 함.

## Time.sleep

프로그램이 제공하는 정보를 사용자가 시간을 두고 확인가능 할 수 있도록 함.

## With open ~

식단별 칼로리 데이터를 가지고 있는 파일을 불러와 사용

## round

칼로리 계산 시 소수점 단위를 반올림

```

class Menu():
    def korean(self):
        car1 = 254 ; pro1 = 128 ; fat1 = 167
        car2 = 258 ; pro2 = 214 ; fat2 = 113
        car3 = 276 ; pro3 = 97 ; fat3 = 92
        car4 = 314 ; pro4 = 260 ; fat4 = 154
        car5 = 164 ; pro5 = 462 ; fat5 = 290

        self.kor1 = f'''
> 1. 현미밥, 데친 두부, 김치국, 계란말이, 배추김치
-----
총 칼로리: [{car1+pro1+fat1}kcal] 탄수화물: [{car1}kcal] 단백질: [{pro1}kcal] 지방: [{fat1}kcal] \n\n'''

        self.kor2 = f'''
> 2. 쌀밥, 갈치구이, 콩조림, 동치미
-----
총 칼로리: [{car2+pro2+fat2}kcal] 탄수화물: [{car2}kcal] 단백질: [{pro2}kcal] 지방: [{fat2}kcal] \n\n'''
        self.kor3 = f'''
> 3. 검정콩밥, 돼지갈비찜, 된장찌개, 배추김치
-----
총 칼로리: [{car3+pro3+fat3}kcal] 탄수화물: [{car3}kcal] 단백질: [{pro3}kcal] 지방: [{fat3}kcal] \n\n'''
        self.kor4 = f'''
> 4. 현미밥, 김치국, 데친 두부, 갈치 구이, 콩조림, 배추김치
-----
총 칼로리: [{car4+pro4+fat4}kcal] 탄수화물: [{car4}kcal] 단백질: [{pro4}kcal] 지방: [{fat4}kcal] \n\n'''
        self.kor5 = f'''
> 5. 삼계탕, 배추김치
-----
총 칼로리: [{car5+pro5+fat5}kcal] 탄수화물: [{car5}kcal] 단백질: [{pro5}kcal] 지방: [{fat5}kcal] \n\n'''

        print(self.kor1)
        time.sleep(1)
        print(self.kor2)
        time.sleep(1)
        print(self.kor3)
        time.sleep(1)
        print(self.kor4)
        time.sleep(1)
        print(self.kor5)

```

## [메뉴 클래스]

식단 내용, 칼로리를 출력해주는 클래스

현재 예시는 한식 식단 메뉴 예시이며,  
양식, 일식 메뉴도 함수로 지정하여 작성함.

## [사용자 정보 입력 클래스] 프로그램이 직접 실행되는 메인 클래스

```
import time
class Recommendation(Menu):
    def __init__(self):
        print('사용하시는 분의 정보를 입력해주세요.')

    def person(self):
        while True:
            name = input('성함을 입력해주세요. ')

            #def 기초대사량(self):
            while True:
                try:
                    age = int(input('나이를 입력해주세요. '))
                    height = int(input('키를 입력해주세요. <cm 단위> '))
                    kg = int(input('몸무게를 입력해주세요. <소수점 제외> '))
                    sex = input('성별을 입력해주세요. <여자 or 남자> ')

                    basic_w = round((65.51 + (9.56*kg) + (1.85*height) - (4.68*age)),0)
                    basic_m = round((66.47 + (13.75*kg) + (5*height) - (6.76*age)),0)
                    print()
                    print('{}님께서 입력해주신 정보는 아래와 같습니다.'.format(name))
                    print()
                    time.sleep(1)
                    print('! 이름: {}님\n! 키: {}cm\n! 몸무게: {}kg\n! 성별: {} '.format(name,height,kg,sex))
                    time.sleep(1)
                    right = int(input('\n맞으면 1 입력, 다시 입력하시려면 2를 입력해주세요.\n입력한 정보가 맞'))
                except Exception:
                    print('다시 입력해주세요.')
                    continue

                if right ==1:
                    pass
                elif right ==2:
                    continue
                else:
                    print('1 or 2 다시 입력해주세요.')
                    continue
```

사용자 정보 입력

기초대사량 계산 식  
변수 지정

사용자 정보 입력 확인  
맞으면 다음으로 패스,  
틀리면 다시 입력



## [사용자 정보 입력 클래스] 프로그램이 직접 실행되는 메인 클래스

```
if sex == '여자':  
    time.sleep(1)  
    print('\n{}님의 기초 대사량은 {}kcal입니다.'.format(name,basic_w))  
    print()  
elif sex == '남자':  
    time.sleep(1)  
    print('\n{}님의 기초 대사량은 {}kcal입니다.'.format(name,basic_m))  
    print()  
else:  
    print('다시 입력해주세요.')  
    continue
```

성별에 따라  
기초대사량 계산

```
while True:  
    try:  
        time.sleep(1)  
        활동지수 = int(input('*활동량이 적으면: 1 입력\n*활동량이 보통이면: 2 입력\n*활동량이 많으면: 3 입력  '))  
        basic = (height - 100) * 0.9  
    except:  
        print('다시 입력해주세요.')  
        continue
```

활동지수 입력 및 활동지수 계산 변수 지정  
활동지수에 따라 총 섭취해야하는  
칼로리 계산할 수 있도록 함

## [사용자 정보 입력 클래스] 프로그램이 직접 실행되는 메인 클래스

```
if 활동지수 == 1:
    if kg > 60:
        a = int(basic*25)
        time.sleep(1)
        print('{}님은 하루 {}kcal를 섭취해야 합니다.'.format(name,a))
        print()
    else:
        a = int(basic*25-200)
        time.sleep(1)
        print('{}님은 하루 {}kcal를 섭취해야 합니다.'.format(name,a))
        print()
```

활동지수가 적을 때,  
몸무게가 60kg 이상, 60kg 이하일 경우  
계산을 다르게 함.

```
elif 활동지수 == 2:
    if kg > 60:
        a = int(basic*30)
        time.sleep(1)
        print('{}님은 하루 {}kcal를 섭취해야 합니다.'.format(name,a))
        print()
    else:
        a = int(basic*30-200)
        time.sleep(1)
        print('{}님은 하루 {}kcal를 섭취해야 합니다.'.format(name,a))
        print()
```

활동지수가 보통일 때,  
몸무게가 60kg 이상, 60kg 이하일 경우  
계산을 다르게 함.

```
elif 활동지수 == 3:
    if kg > 60:
        a = int(basic*40)
        time.sleep(1)
        print('{}님은 하루 {}kcal를 섭취해야 합니다.'.format(name,a))
        print()
    else:
        a = int(basic*40-200)
        time.sleep(1)
        print('{}님은 하루 {}kcal를 섭취해야 합니다.'.format(name,a))
        print()
```

활동지수가 많을 때,  
몸무게가 60kg 이상, 60kg 이하일 경우  
계산을 다르게 함.

```
else:
    print('다시 입력해주세요.')
    continue
```

## [사용자 정보 입력 클래스] 프로그램이 직접 실행되는 메인 클래스

```
while True:
    #def 탄단지(self,height):
    try:
        time.sleep(1)
        diet = int(input('다이어트 중이신가요? <y: 1 입력 or n: 2 입력> '))
        bcar = round(a*0.5);bpro = round(a*0.3);bfat = round(a*0.2)
        b_diet = '탄수화물: {}kcal, 단백질: {}kcal 지방: {}kcal'.format(bcar,bpro,bfat)
        n_diet = '탄수화물: {}kcal, 단백질: {}kcal 지방: {}kcal'.format(bcar,bpro,bfat)
    except:
        print('1(y) or 2(n)으로 다시 입력해 주십시오.')
        continue

    if diet == 1:
        time.sleep(1)
        print('다이어트 중이시면 {}를 섭취해야 합니다.'.format(b_diet))
        print()
    elif diet == 2:
        time.sleep(1)
        print('다이어트 중이 아니시면 {}를 섭취해야 합니다.'.format(n_diet))
        print()
```

다이어트 여부 확인  
탄수화물, 단백질, 지방 섭취  
비율 계산 변수 지정

다이어트 여부에 따른 멘트 출력



# [사용자 정보 입력 클래스] 프로그램이 직접 실행되는 메인 클래스

```
while True:
    #def 식단추천(self,name):
    try:
        time.sleep(1)
        print('메뉴를 추천해드리겠습니다.')
        끼니 = int(input('평소 몇 끼니를 드시나요? <2끼: 1 입력, 3끼: 2 입력>'))
    except:
        print('1(2끼) or 2(3끼)를 입력해 주십시오.')
        continue
    if 끼니 == 1:
        time.sleep(1)
        print(f'{name}님의 1끼 식사별 적정 kcal을 안내 해드립니다. 총 kcal:{a/2}kcal 탄수화물: {a/2*0.5}kcal 단백질: {a/2*0.5}kcal')
        for i in range(2):
            while True:
                try:
                    time.sleep(1)
                    menu = input('원하시는 종류를 선택해주세요. <한식: 1 입력, 양식: 2 입력, 일식: 3 입력>')
                    time.sleep(1)
                except:
                    print('1(한식) or 2(양식) or 3(일식) 으로 다시 입력해 주십시오.')
                    return
                if menu == '1':
                    self.korean()
                    num = int(input('원하시는 메뉴의 번호를 입력해주세요. '))
                    if num == 1:
                        print(self.kor1+'를 선택하셨습니다.')
                    elif num == 2:
                        print(self.kor2+'를 선택하셨습니다.')
                    elif num == 3:
                        print(self.kor3+'를 선택하셨습니다.')
                    elif num == 4:
                        print(self.kor4+'를 선택하셨습니다.')
                    elif num == 5:
                        print(self.kor5+'를 선택하셨습니다.')
                    else:
                        print('다시 입력해주세요.')
                        continue
                    break
                elif menu == '2':
                    self.western()
                    num = int(input('원하시는 메뉴의 번호를 입력해주세요. '))
                    if num == 6:
                        print(self.wes1+'를 선택하셨습니다.')
```

평소 먹는 끼니 수 입력  
1끼별 섭취 칼로리 계산

한식, 양식, 일식 메뉴 선택

상속받은  
한식, 양식, 일식  
메뉴 선택 Class

## [사용자 정보 입력 클래스] 프로그램이 직접 실행되는 메인 클래스

매 끼당 식단 선택 후 섭취 해야 될 칼로리 잔여량 계산 코드

2끼, 3끼 계산 코드 有

```
if i == 0:
    filename = '칼로리표.csv'
    with open(filename, 'r') as f:
        kcal_list = []
        while True:
            data = f.readline()
            if not data:
                break
            div = data.split(',')
            if div[0] == '{}'.format(num):
                kcal_list.append(div)
            time.sleep(1)
    i1 = int(a - int(kcal_list[0][1])); i2 = int(bcar - int(kcal_list[0][2])); i3 = int(bpro - int(kcal_list[0][3])); i4 = int(bfat - int(kcal_list[0][4]))
    print(f'\n|총 kcal: [{i1}kcal] |탄수화물: [{i2}kcal] |단백질: [{i3}kcal] |지방: [{i4}kcal] |\n를 더 섭취
```

첫번째 끼니 선택 후  
섭취 칼로리 잔여량 계산

```
elif i == 1:
    with open(filename, 'r') as f:
        kcal_list = []
        while True:
            data = f.readline()
            if not data:
                break
            div = data.split(',')
            if div[0] == '{}'.format(num):
                kcal_list.append(div)
            time.sleep(1)
    i5 = i1 - int(kcal_list[0][1]); i6 = i2 - int(kcal_list[0][2]); i7 = i3 - int(kcal_list[0][3]); i8 = i4 - int(kcal_list[0][4])
    print(f'\n|총 kcal: [{i5}kcal] |탄수화물: [{i6}kcal] |단백질: [{i7}kcal] |지방: [{i8}kcal] |\n를 더 섭취
```

두번째 끼니 선택 후  
섭취 칼로리 잔여량 계산

# [사용자 정보 입력 클래스] 프로그램이 직접 실행되는 메인 클래스

최종 칼로리 잔여량에 따른 문구 출력

```
if (i5 > 0):  
    print()  
    time.sleep(1)  
    print(''
```

하루 권장 kcal보다 적은 kcal의 식단을 선택하셨습니다.  
식사사이사이 먹을 수 있는 간식을 추천드립니다.  
간식을 챙겨 드시기 바랍니다.

[추천 간식 목록]

|아몬드 10알 \*총 kcal: [18kcal] 탄수화물: [9kcal] 단백질: [9kcal]  
|바나나 1개 \*총 kcal: [185kcal] 탄수화물: [160kcal] 단백질: [8kcal] 지방: [17kcal]  
|삶은 달걀 2개 \*총 kcal: [15kcal] 탄수화물: [8kcal] 단백질: [55kcal] 지방: [69kcal]  
''')

```
elif (i5 < 0):  
    print()  
    time.sleep(1)  
    print(''
```

하루 권장 kcal보다 많은 kcal의 식단을 선택하셨습니다.  
식사량을 조절하면서 드시는 것을 추천드립니다.'''')  
time.sleep(1)  
print('맛있는 식사를 하세요.^-^')  
break

최종 칼로리가 섭취 칼로리  
보다 적을 경우  
더 섭취할 간식 추천

최종 칼로리가 섭취 칼로리  
보다 많을 경우  
식사량 조절할 것을 추천

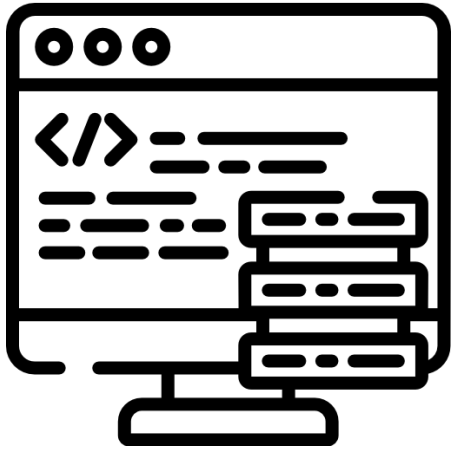
[프로그램 종료]

구조



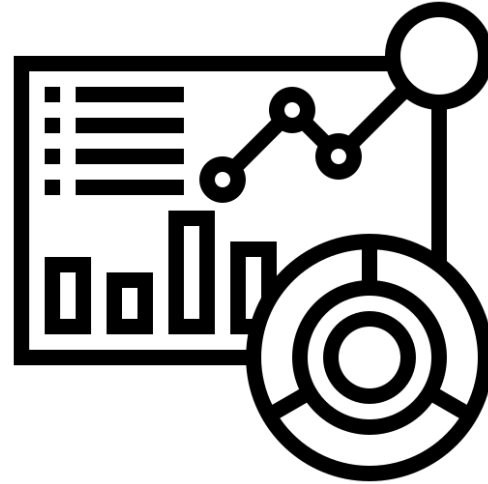
# Program 보완점

1.



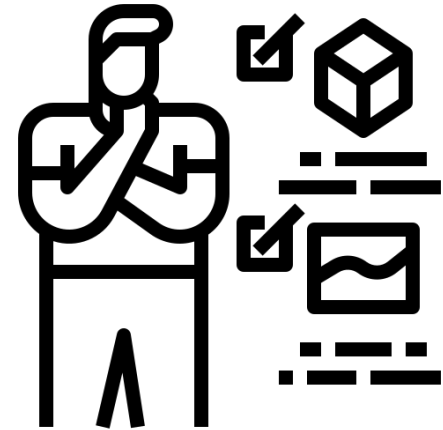
코드 간소화

2.



데이터 추가

3.



사용자가 직접  
메뉴 구성하도록 수정

# Program 기대 효과

1. 각 질병에 따라 섭취해야 되는 식단 입력 후 환자들이 정보를 얻을 수 있도록 함.
  - 질병 카테고리를 따로 지정해 식단 추천을 할 수 있도록 함.
2. 재료, 메뉴 별 탄수화물, 단백질, 지방 뿐 아니라 나트륨, 비타민, 당질 등 데이터를 추가하면 더 많은 케이스의 사람들이 이용할 수 있음.
3. 자신의 식단을 입력하여 사람들과 공유할 수 있도록 함.
4. 알레르기 또는 못 먹는 음식 입력 시, 대체 식품 추천할 수 있도록 함.

# Program 기대 효과

처음에는 일반 사람들 대상으로 식단을 추천하려고 했지만,  
식단 조절을 해야하는 대상은 일반인이 아닌 질병을 가진 사람들이라고 판단함.

따라서 각 질병 및 알레르기 식단을 추천하고 정보를 알려주는 프로그램으로  
업데이트하면 병원 및 일상생활에서 유용하게 활용 가능하다고 생각함.



**End of document.**

**Q & A Time**