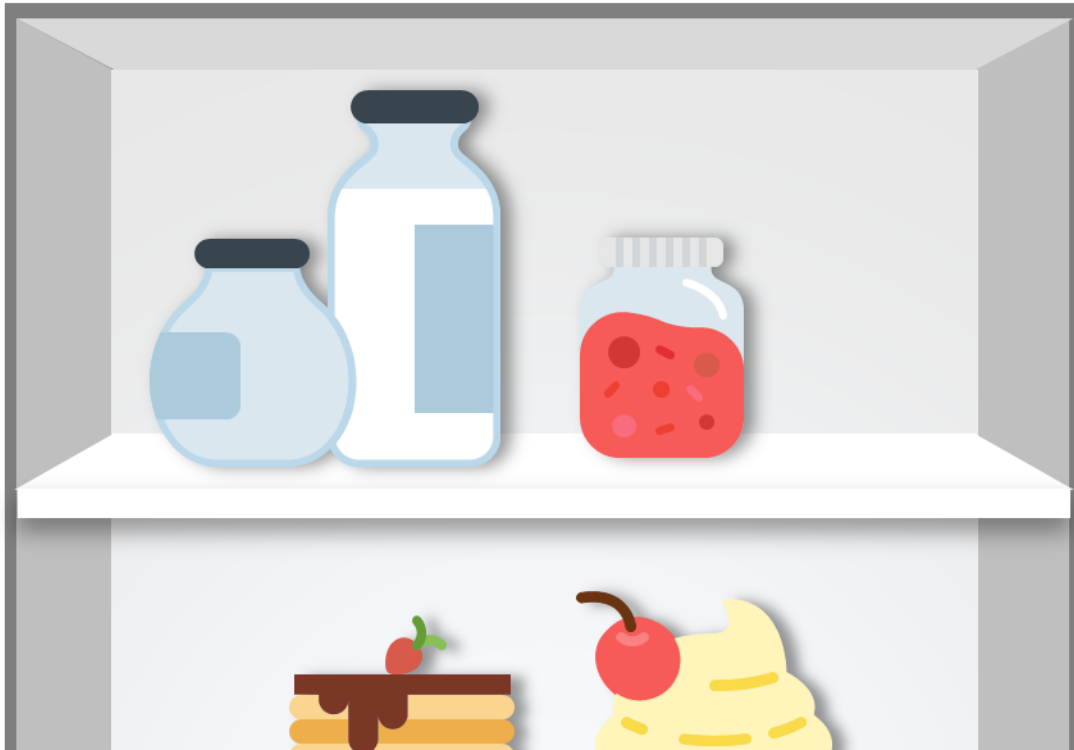


박준석

전기전자공학과 2017313822

김승윤

문헌정보학과 2017310301



What's in My Refrigerator

2020-2 데이터사이언스와 컴퓨팅2

안용학 교수님

데이터 관리 프로그램 개발 팀 프로젝트



What's in My Refrigerator

데이터 관리 프로그램 개발 팀 프로젝트

1. Data Structure

냉장고 속 식재료 데이터를 관리하기 위해
사용한 자료구조 : 힙 Heap에 대한 설명

2. Classes

냉장고 식재료 관리 프로그램 구현을 위해 사용
한 클래스에 대한 설명

3. Functions

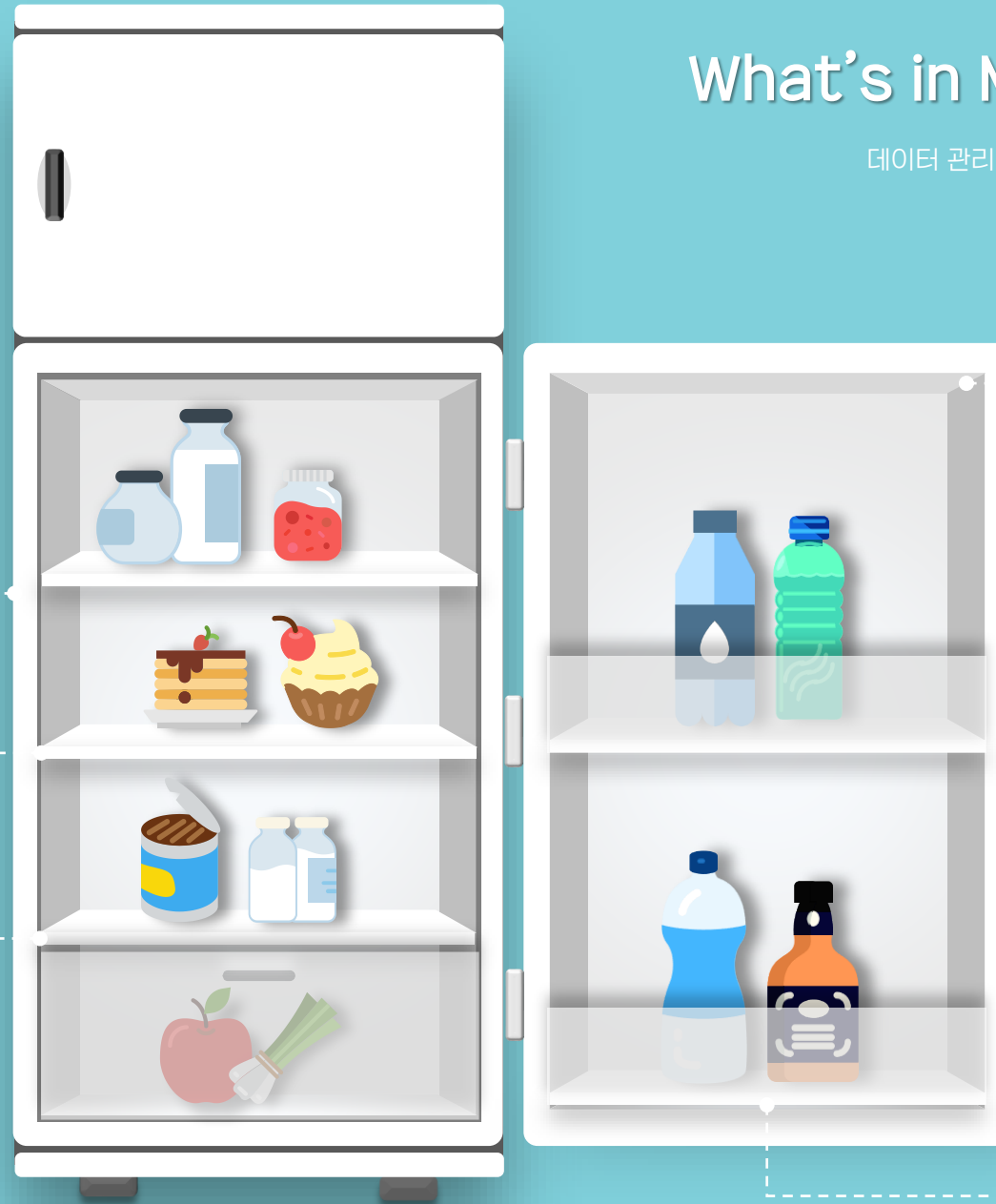
냉장고 식재료 관리 프로그램 구현을 위해 작성
한 함수에 대한 설명

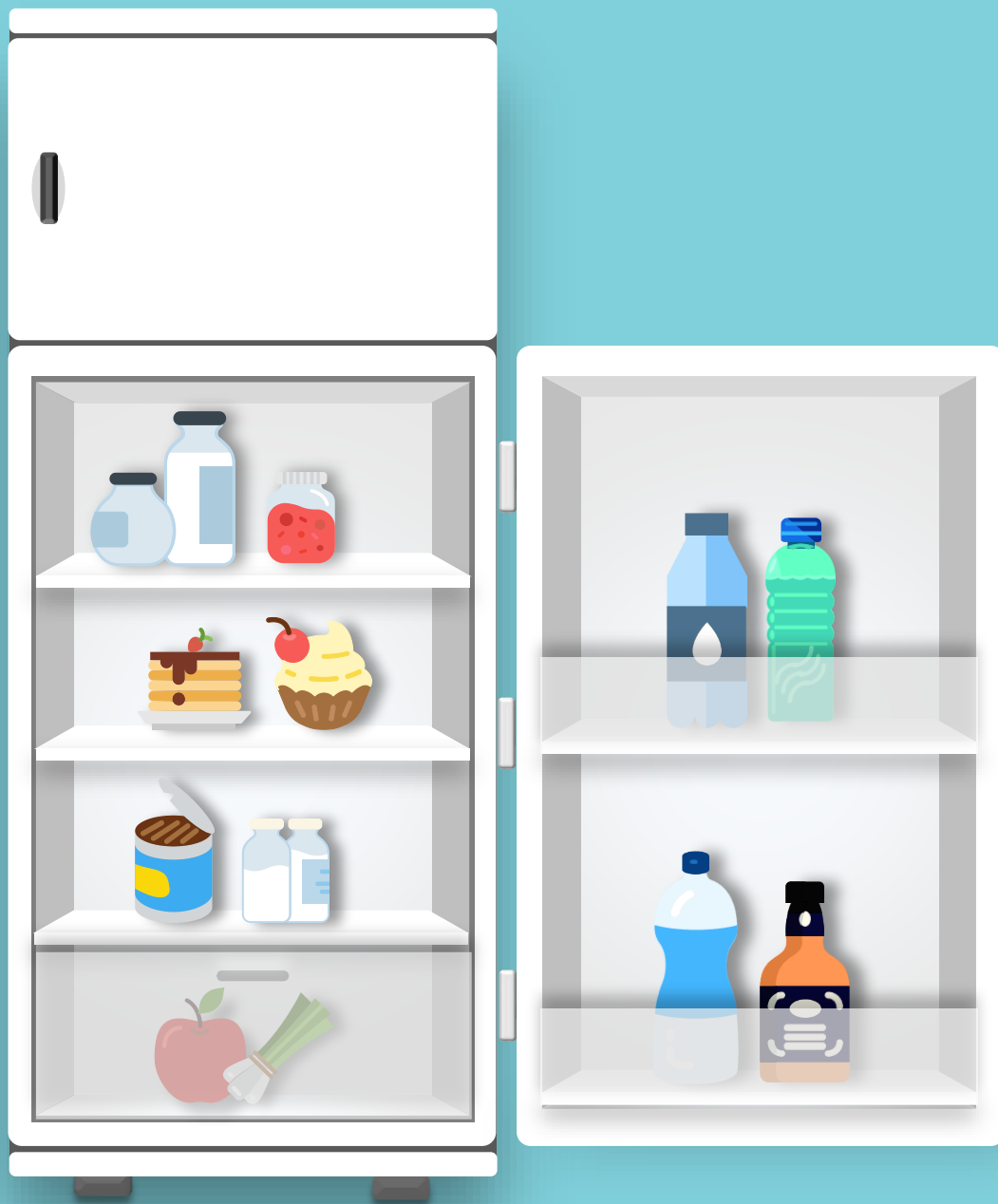
4. Interface

컨텐츠에 대한 내용을 적어요
냉장고 식재료 관리 프로그램 사용 인터페이스에
대한 설명

5. Execution

냉장고 관리 프로그램 시연





What's in My Refrigerator

데이터 관리 프로그램 개발 팀 프로젝트

- 냉장고 속 식재료 데이터 관리 프로그램
- Python 기반
- 냉장고 속 식재료를 유통기한 순으로 정렬하여 관리한다.
- 유통기한이 임박하면 알려준다.
- 냉장고에 있는 식재료를 이용하여 만들 수 있는 요리를 알려준다.



1. Data Structure

What's in My Refrigerator

ADVANTAGE

- 처리 속도가 빠르다
 $O(n \log n)$
- 제자리 정렬이 가능하다

HEAP

DISADVANTAGE

- 안정성이 떨어진다

Data Structure

Classes

Functions

Interface

Execution



Data Structure

Classes

Functions

Interface

Execution



1. Data Structure

What's in My Refrigerator

✓ What we choose HEAP

1. 많은 자료를 불러올 때 ex) 파일 시작 시 텍스트 파일 읽어올 때 상향식 힙 생성으로 $O(n)$ 시간만이 소요된다.
2. item을 추가할 때 $O(\log n)$ 시간만이 소요된다.
3. 정렬할 때 힙 정렬 2기만 수행하면 된다.



Data Structure

Classes

Functions

Interface

Execution



1. Data Structure

What's in My Refrigerator

✓ Updating Item

Update한 자료를 기준으로 upheap 또는 downheap을 수행

✓ Removing Item

item remove은 힙 속성을 유지하기 위해 Root를 제거할 때와 유사한 방식으로 수행된다.

(1) 수정할 자료와 마지막 자료의 위치 변경

(2) 마지막에 위치한 자료를 삭제한 것으로 간주

(3) 수정할 자료가 있던 자리에서 수정된 값에 따라 upheap 또는 downheap 수행



Data Structure

Classes

Functions

Interface

Execution

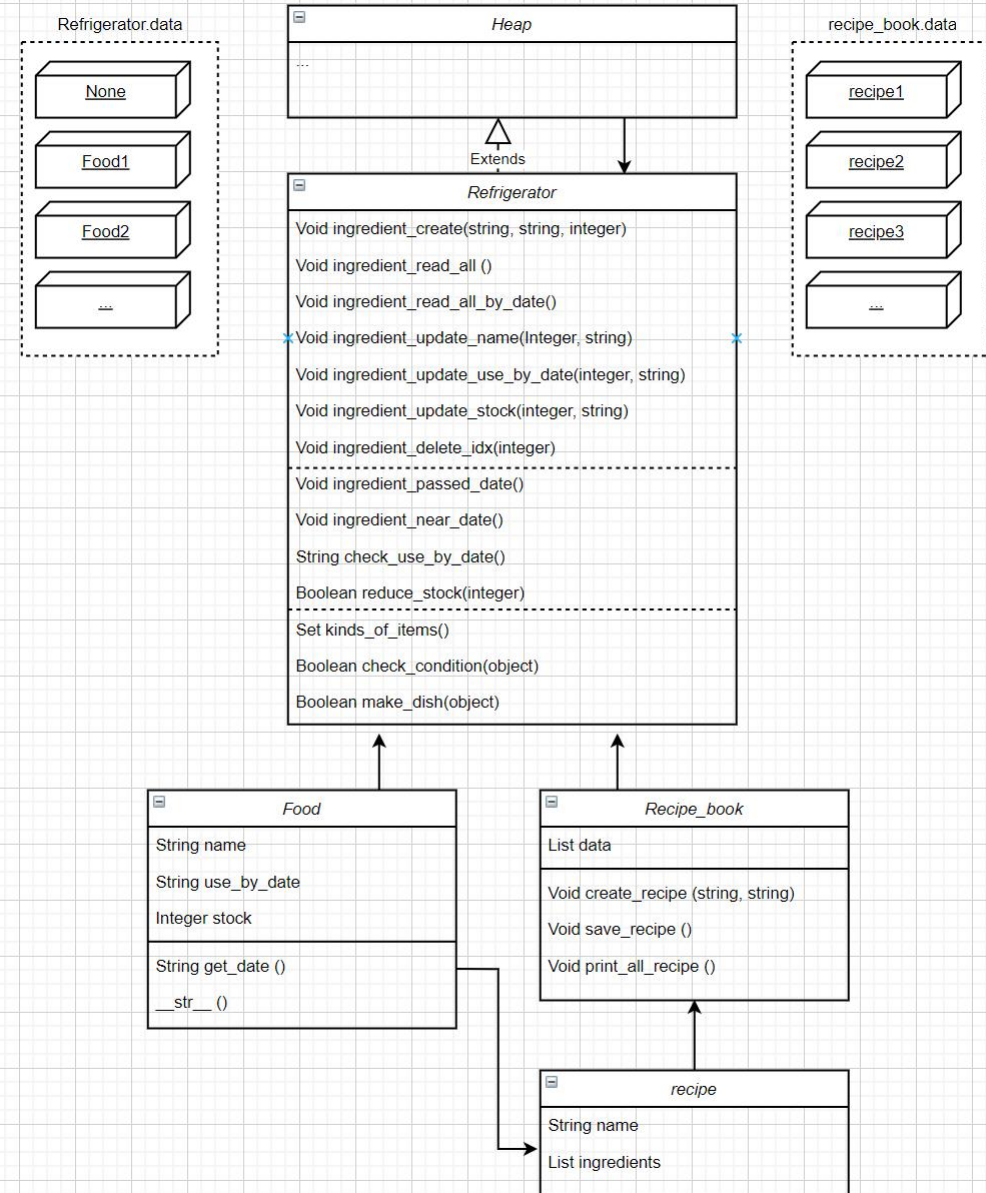


1. Data Structure

What's in My Refrigerator

✓ Class Diagram

- Refrigerator 클래스에서 CRUD, 추가 기능을 제공하는 함수에 접근
- Refrigerator를 상속한 Heap
- Refrigerator 클래스에서 사용된 객체: Food, Recipe_book
- Recipe_book의 사용된 객체: recipe
- Recipe의 객체로 사용된 Food





2. Classes

What's in My Refrigerator

Refrigerator

식재료 데이터를 저장하고, CRUD 기능을 제공하는 클래스

Food

관리 대상 데이터로 이름, 유통기한, 재고수량을 클래스 변수로 가지는 클래스

Ingredient & Dish

Food 객체의 type을 지정하는 클래스



2. Classes

What's in My Refrigerator

recipe

식재료를 이용해 만들 수 있는 요리 레시피를 나타내는 클래스로 레시피 이름과 필요한 재료를 클래스 변수로 가짐

Recipe_book

recipe 클래스를 객체로 저장하고 출력하는 클래스

What's
in My
Refrig.



2. Classes

What's in My Refrigerator

Heap

힙 정렬을 통해 데이터를 관리하는 클래스

Data Structure

Classes

Functions

Interface

Execution



3. Functions

What's in My Refrigerator

Refrigerator.py

```
def ingredient_create(self, name, use_by_date, stock):  
    # 이름, 유효기간, 재고수를 입력하여 새로운 식재료 생성  
def ingredient_read_all(self):  
    # refrigerator에 들어간 모든 식재료 데이터 출력  
def ingredient_read_all_by_date(self):  
    # refrigerator에 들어간 모든 식재료 데이터를 유효기간 순으로 출력
```



3. Functions

What's in My Refrigerator

Refrigerator.py

```
def ingredient_update_name(self, idx, new_name):  
    # 식재료 이름 수정  
def ingredient_update_use_by_date(self, idx, new_use_by_date):  
    # 식재료 유효기간 수정  
def ingredient_update_stock(self, idx, new_stock):  
    # 식재료 재고 수량 수정  
def ingredient_delete_idx(self, idx):  
    # 특정 식재료 제거
```



3. Functions

What's in My Refrigerator

Refrigerator.py

```
def ingredient_update_name(self, idx, new_name):  
    # 식재료 이름 수정  
def ingredient_update_use_by_date(self, idx, new_use_by_date):  
    # 식재료 유효기간 수정  
def ingredient_update_stock(self, idx, new_stock):  
    # 식재료 재고 수량 수정  
def ingredient_delete_idx(self, idx):  
    # 특정 식재료 제거
```



3. Functions

What's in My Refrigerator

Refrigerator.py

```
def ingredient_passed_date(self):  
    # 유효기간 지난 식재료 반환  
def ingredient_near_date(self):  
    # 유효기간 임박한 식재료 반환  
def check_use_by_date(self):  
    # 유효기간 지난, 임박한 식재료 반환  
def reduce_stock(self, idx):  
    # 사용한 식재료 재고 수량 감소, 재고가 다 떨어진 경우 삭제
```



3. Functions

What's in My Refrigerator

Refrigerator.py

```
def kinds_of_items(self):  
    # 식재료의 타입 (ingredient, dish) 반환  
def check_condition(self):  
    #  
def make_dish(self, recipe):  
    # 저장된 식재료로 요리를 만들 수 있는지 여부 반환
```



Data Structure

Classes

Functions

Interface

Execution



4. Interface

What's in My Refrigerator

What's in My Refrigerator

[Refrigerator CRUD]

[1] CREATE

[2] READ

[3] UPDATE

[4] DELETE

[5] Additional Features

[6] Exit

Enter the number:

|

What's
in My
Refrig.



Data Structure

Classes

Functions

Interface

Execution



4. Interface

What's in My Refrigerator

What's in My Refrigerator

```
Returning to the main menu    #retrun message  
error : <return_msg>        #error message
```

|



4. Interface

What's in My Refrigerator

What's in My Refrigerator

```
# create - insert ingredient to the refrigerator
```

```
name :  
expiration data YYYY-MM-DD :  
number of stock:
```

|

What's
in My
Refrig.



Data Structure

Classes

Functions

Interface

Execution



4. Interface

What's in My Refrigerator

What's in My Refrigerator

```
# show all ingredients in the refrigerator  
# ex) 1. ingredient, 양파, 2021-03-18, 3  
#      2. ingredient, 감자, 2021-03-18, 3
```

|



4. Interface

What's in My Refrigerator

What's in My Refrigerator

```
# update information of ingredients
```

```
*****
```

```
[1] UPDATE name
```

```
[2] UPDATE use by date
```

```
[3] UPDATE stock
```

```
*****
```

```
Enter the number:
```

```
|
```



4. Interface

What's in My Refrigerator

What's in My Refrigerator

```
# show all ingredients in the refrigerator  
# ex) 1. ingredient, 양파, 2021-03-18, 3  
#      2. ingredient, 감자, 2021-03-18, 3
```

Enter the number to modify:

```
# name update
```

new name :

```
# use by date update
```

new use by date (yyyy-mm-dd) :

```
# stock update
```

new stock :

|

What's
in My
Refrig.



Data Structure

Classes

Functions

Interface

Execution



4. Interface

What's in My Refrigerator

What's in My Refrigerator

```
# delete a specific ingredient  
# show all items in refrigerator
```

Enter the number to delete:

```
>> <removed item> deleted
```

|

What's
in My
Refrig.



Data Structure

Classes

Functions

Interface

Execution



4. Interface

What's in My Refrigerator

What's in My Refrigerator

- [1] Check Use by Date
- [2] Food Consumption
- [3] Cookable Dish
- [4] Cook Dish
- [5] Check Recipe
- [6] New Recipe
- [7] Return to main menu

Enter the number:

|

What's
in My
Refrig.



4. Interface

What's in My Refrigerator

What's in My Refrigerator

exit the program...

Data Structure

Classes

Functions

Interface

Execution

What's
in My
Refrig.



5. Execution

What's in My Refrigerator

Data Structure

Classes

Functions

Interface

Execution

Thank You

What's in My Refrigerator

데이터 관리 프로그램 개발 팀 프로젝트



removed

What's
in My
Refrig.



Data Structure

Classes

Functions

Interface

Execution



3. Functions

What's in My Refrigerator

Data_structure.py

- Heap 구현 함수

```
class Heap:

    def __init__(self):
        self.data = [None]
        self.last = 0
```



3. Functions

What's in My Refrigerator

Data_structure.py

```
def left_child(self, i):  
    # 인자로 전달된 노드의 왼쪽 자식 노드 인덱스 반환  
def right_child(self, i):  
    # 인자로 전달된 노드의 오른쪽 자식 노드 인덱스 반환  
def parent(self, i):  
    # 인자로 전달된 노드의 부모 노드 인덱스 반환  
def swap_element(self, i, j):  
    # 인자로 전달된 노드 2개를 맞바꿈
```



3. Functions

What's in My Refrigerator

Data_structure.py

```
def downHeap(self, i):  
    # 힙 속성 복구를 위한 downHeap  
def downHeap_by_date(self, i):  
    # 식재료 데이터의 유효기간을 기준으로 힙 정렬을 하기 위한 downHeap  
def downHeap_by_name(self, i):  
    # 식재료 데이터의 이름을 기준으로 힙 정렬을 하기 위한 downHeap
```



3. Functions

What's in My Refrigerator

Data_structure.py

```
def upHeap(self, i):  
    # 힙 속성 복구를 위한 upHeap  
def upHeap_by_date(self, i):  
    # 식재료 데이터의 유효기간을 기준으로 힙 정렬을 하기 위한 upHeap  
def upHeap_by_name(self, i):  
    # 식재료 데이터의 이름을 기준으로 힙 정렬을 하기 위한 upHeap
```



3. Functions

What's in My Refrigerator

Data_structure.py

```
def insert_item(self, item):  
    # Heap에 식재료 데이터 삽입 후 힙 속성 복구를 위해 upheap 호출  
def insert_item_by_date(self, item):  
    # Heap에 데이터 삽입 후 유효기간을 기준으로 힙 속성 복구를 위해 upheap 호출
```

Data Structure

Classes

Functions

Interface

Execution



3. Functions

What's in My Refrigerator

Data_structure.py

```
def remove_max(self):  
    # Heap에서 최대값 제거  
def remove_max_by_date(self):  
    # Heap에서 식재료 데이터의 유효기간이 가장 큰 값 제거  
def remove_idx(self, i):  
    # Heap에서 특정 인덱스 값 제거  
    # Heap 속성 복구를 위해 upheap, downHeap 모두 수행  
def remove_idx_by_date(self, i):  
    # Heap에서 특정 유효기간 데이터를 갖는 값 제거  
    # Heap 속성 복구를 위해 upheap, downHeap 모두 수행
```

What's
in My
Refrig.



Data Structure

Classes

Functions

Interface

Execution



3. Functions

What's in My Refrigerator

Data_structure.py

```
def get_sorted_by_date(self):  
    #  
def get_sorted_by_name(self):  
    #  
def print_heap(self):  
    #  
def print_heap_items(self):  
    #  
def build_heap(self, data_list):  
    # 비재귀적 상향식 힙생성  
def build_heap_by_date(self, data_list):  
    # 비재귀적 상향식 힙생성
```