

냉장고 현황판 프로그램

진척 보고서

제출일자: 2024.11.03

과목: C++ 프로그래밍및실습

학과: 소프트웨어공학과

학번: 214270

이름: 현경서

목차

1. 서론
 - 1) 배경 및 필요성
 - 2) 프로젝트 목표
 - 3) 차별점
2. 기능 계획
 - 1) 음식 목록 추가/제거
 - 2) 유통기한 임박 음식 출력
 - 3) 추천 메뉴 출력
3. 진척사항
 - 1) 기본 환경 구현
 - 2) 기능 구현
 - 3) 테스트 결과
4. 계획 대비 변경 사항
 - 1) 변경사항 없음
5. 프로젝트 일정

1. 서론

1) 배경 및 필요성

냉장고에 있는 음식을 제때 처리하지 못하거나 어떤 음식이 있는지 잊어버리는 경우가 있어서 냉장고 현황판이 있어야겠다는 생각을 했다. 이번 프로젝트 통해서 이걸 프로그램으로 만들어보려 한다.

2) 프로젝트 목표

냉장고 안 음식 리스트를 csv 파일에 저장하고 이를 불러와서 유통기한이 임박한 음식과 추천 메뉴를 알려준다.

3) 차별점

냉장고 현황판을 프로그램으로 만든다면 아날로그로 사용할 때 보다 더 많은 기능을 사용할 수 있다. 구현하려는 기능 중 추천 메뉴 출력 기능은 아날로그 현황판에서는 사용할 수 없는 기능이다. 아날로그 현황판에서는 현황판의 음식을 보고 직접 검색해서 메뉴를 찾아내야 하지만 프로그램의 경우 알아서 메뉴를 알려준다. 냉장고에 남아있는 음식과 식사 메뉴 고민을 한번에 해결해 줄 수 있다.

또한 유통기한이 임박한 음식을 확인할 때 아날로그 현황판의 경우 직접 확인해야 하지만 프로그램의 경우 분류해서 알려주기 때문에 더 편하다고 할 수 있다.

2. 기능 계획

1) 음식 목록 추가/제거

음식 목록은 csv 파일에 저장해서 관리하며 프로그램은 csv 파일을 불러와서 목록을 변경한다. 새로운 음식을 추가하거나 이미 소비한 음식을 제거할 수 있다.

2) 유통기한 임박 음식 출력

csv 파일에는 음식의 이름, 개수, 유통기한을 저장한다. 유통기한이 얼마 남지 않은 식품들을 확인할 수 있다.

3) 메뉴 출력

csv 파일에 있는 음식들을 기반으로 추천 메뉴를 출력한다. openai api를 이용한다.

3. 진척사항

1) 기본 환경 구현

① 음식 정보 구조체

```
24 // 음식 정보 구조체
25 struct FoodInfo{
26     string name;
27     int count;
28     int date; // YYYYMMDD
29 };
```

② 음식 목록 파일 경로 전역 상수로 선언

```
31 const string file_dir = "food_list.csv";
```

③ 음식 목록 불러오기

```

45 int main(){
46     // 음식 리스트 불러오기
47     if (!isFile()){
48         cout << "프로그램을 종료합니다" << endl;
49         return 0;
50     }
51
52     vector<FoodInfo> food_list; // 음식 리스트를 저장하는 벡터
53     FoodInfo food;
54     string elem;
55     ifstream file(file_dir);
56
57     int i = 0; // while내 인덱스를 위한 변수
58     while (file >> elem){
59         // ',' 제거
60         size_t pos = elem.find(',');
61         if (pos != string::npos)
62             elem.replace(pos, 1, "");
63         // 요소 저장
64         switch (i % 3){
65             case 0:
66                 food.name = elem;
67                 break;
68             case 1:
69                 food.count = stoi(elem);
70                 break;
71             case 2:
72                 food.date = stoi(elem);
73                 food_list.push_back(food);
74                 break;
75         }
76         i += 1;
77     }
78     file.close();

```

- isFile() 함수로 파일이 존재하는지 확인한다. (isFile()함수는 기능 구현 부분에서 설명)
- csv파일을 읽어서 구조체 벡터에 저장한다.
- 파일의 내용이 바뀔 때마다 파일을 다시 불러오는 대신, 벡터를 생성하여 파일과 벡터를 동시에 변경하는 방식으로 구현한다.

④ 구조체 원소에 대한 사용자 입력 함수

```

138 string cinName(){
139     string name;
140     cout << "음식 이름: ";
141     cin >> name;
142     return name;
143 }

```

```

146 int cinCount(){
147     int count;
148
149     while(1){
150         cout << "개수: ";
151         cin >> count;
152         if (cin.fail() || cin.peek() != '\n'){
153             cout << "정수를 입력하십시오" << endl;
154             cin.clear();
155             cin.ignore(numeric_limits<streamsize>::max(), '\n');
156             continue;
157         }
158         break;
159     }
160
161     return count;
162 }
175 int cinDate(){
176     int date;
177     int current_date = getCurrentDate();
178
179     while(1){
180         cout << "유통기한: ";
181         cin >> date;
182         if (cin.fail() || cin.peek() != '\n'){
183             cout << "날짜 포맷에 맞춰 입력하십시오(YYYYMMDD)" << endl;
184             cin.clear();
185             cin.ignore(numeric_limits<streamsize>::max(), '\n');
186             continue;
187         }
188         if (countDigits(date) != 8){
189             cout << "날짜 포맷에 맞춰 입력하십시오(YYYYMMDD)" << endl;
190             continue;
191         }
192         if (date < current_date){
193             cout << "유통기한이 이미 지났습니다" << endl;
194             continue;
195         }
196         break;
197     }
198
199     return date;
200 }

```

- I. 입력
- II. 결과
 - 사용자 입력 반환
- III. 설명
 - 정수 입력의 경우 정수가 아닌 값이 입력되었는지 확인이 필요하다. 정수가 아닌 값이 입력되면 다음 입력에서 오류가 발생하므로, 이를 방지하기 위해 입력이 정수가 아니면 입력 버퍼를 완전히 비우고 다시 입력 받는다.

- 날짜 입력의 경우 날짜 포맷을 맞춰서 입력 받아야 하며 유통기한이 이미 지났는지 확인해야 한다.
- 사용자의 입력을 받을 때 확인해야 할 부분이 많으므로 이를 함수로 구현하여 입력이 필요한 부분에서 코드를 더 간결하게 작성할 수 있게 하였다.
- 이름 입력의 경우 확인해야 할 부분은 없지만 다른 두 원소의 입력이 함수로 구현되어 있으므로 이름 입력도 함수로 구현하였다.

⑤ 자릿수 반환 함수

```

314 int countDigits(int number){
315     int count = 0;
316     while (number != 0) {
317         number /= 10;
318         count++;
319     }
320     return count;
321 }

```

- I. 입력
 - number: 정수
- II. 결과
 - number의 자릿수 반환
- III. 설명
 - 정수의 자릿수가 필요한 경우가 있어서 구현하였다.

⑥ 현재 날짜 반환 함수

```

353 int getCurrentDate(){
354     int current_date;
355
356     time_t raw_time;
357     struct tm* time_info;
358     raw_time = time(NULL);
359     time_info = localtime(&raw_time);
360
361     int current_y = time_info->tm_year + 1900;
362     int current_m = time_info->tm_mon + 1;
363     int current_d = time_info->tm_mday;
364
365     current_date = current_y*10000 + current_m*100 + current_d;
366     return current_date;
367 }

```

- I. 입력
- II. 결과
 - 현재 날짜 반환
- III. 설명

- 유통기한 임박 음식을 확인할 때 현재 날짜가 필요해서 구현하였다.

2) 기능 구현

① 파일 존재 확인

```

124 ~ bool isFile(){
125     fstream file(file_dir);
126 ~ if (!file.is_open()){
127         cout << "음식 목록 파일을 찾을 수 없습니다" << endl;
128         file.close();
129         return false;
130     }
131 ~ else{
132         file.close();
133         return true;
134     }
135 }

```

I. 입력

II. 결과

- 파일 유무에 대한 true/false 출력

III. 설명

- 전역 상수로 선언한 경로에 파일이 존재하지 않을 경우 오류 메시지를 출력하고 false를 반환한다.
- 파일이 존재할 경우 true를 반환한다.

② 기능 출력

```

188 void printFunction(){
189     cout << "1. 현재 음식 보기" << endl;
190     cout << "2. 음식 추가" << endl;
191     cout << "3. 음식 제거" << endl;
192     cout << "4. 유통기한 임박 음식 보기" << endl;
193     cout << "5. 추천 메뉴 보기" << endl;
194     cout << "6. 프로그램 종료" << endl;
195 }

```

I. 입력

II. 결과

- 사용할 수 있는 기능 출력

III. 설명

- 기능이 너무 많아서 사용자가 원할 때만 기능을 볼 수 있게 하기 위해 함수로 구현하였다.

③ 기능 선택

```
90     printFunction();
91     while (1){
92         // 입력
93         int choice; // 사용자 입력을 저장하는 변수
94         cout << "사용할 기능을 선택하십시오(기능 보기: 0): ";
95
96         cin >> choice;
97         if (cin.fail() || cin.peek() != '\n'){
98             cout << "정수를 입력하십시오" << endl;
99             cin.clear();
100             cin.ignore(numeric_limits<streamsize>::max(), '\n');
101             continue;
102         }
103         if (choice == 0){
104             printFunction();
105         }
106         else if (choice == 1){
107             printFood(food_list);
108         }
109         else if (choice == 2){
110             addFood(food_list);
111         }
112         else if (choice == 3){
113             deleteFood(food_list);
114         }
115         else if (choice == 4){
116             printExpiringFood(food_list);
117         }
118         else if (choice == 5){
119
120         }
121         else if (choice == 6){
122             break;
123         }
124         else{
125             cout << "잘못된 입력입니다" << endl;
126         }
127     }
128
129     cout << "프로그램을 종료합니다" << endl;
130     return 0;
131 }
```

I. 입력

- 사용자 입력

II. 결과

- 사용자 입력에 따른 함수 호출

III. 설명

- 맨 처음에는 기능을 출력하고 그 다음 입력부터는 사용자가 요구하지 않는 한 기능을 출력하지 않는다.
- 사용자 입력이 정수이므로 정수가 아닌 입력에 대한 검사와 처리를 한다.
- 사용자의 입력이 0인 경우 기능을 출력한다.
- 사용자의 입력이 1인 경우 printFood() 함수를 호출하여 음식 목록

을 출력한다.

- 사용자의 입력이 2인 경우 addFood() 함수를 호출하여 음식을 추가한다.
- 사용자의 입력이 3인 경우 deleteFood() 함수를 호출하여 음식을 제거한다.
- 사용자의 입력이 6인 경우 프로그램을 종료한다.
- 그 외의 입력의 경우 오류 메시지를 출력하고 다시 입력 받는다.

④ 음식 목록 출력

```
198 void printFood(vector<FoodInfo> list){
199     int max_len = 0; // 이름 최대 길이
200     int max_digits = 0; // 개수 최대 자릿수
201     int date_size = 8; // 날짜 길이 (8로 고정)
202
203     for (auto food : list){
204         // 이름 길이
205         int len = food.name.length();
206         if (len > max_len)
207             max_len = len;
208         // 개수 자릿수
209         int digits = countDigits(food.count);
210         if (digits > max_digits)
211             max_digits = digits;
212     }
213
214     cout << "|-";
215     for (int i = 0; i < max_len; i++) cout << '-';
216     cout << "-|-";
217     for (int i = 0; i < max_digits; i++) cout << '-';
218     cout << "-|-";
219     for (int i = 0; i < date_size; i++) cout << '-';
220     cout << "-|" << endl;
221
222     for (auto food : list){
223         cout << "| " << food.name;
224         for (int i = 0; i < max_len - food.name.length(); i++) cout << ' ';
225         cout << " | " << food.count;
226         for (int i = 0; i < max_digits - countDigits(food.count); i++) cout << ' ';
227         cout << " | " << food.date << " |" << endl;
228
229         cout << "|-";
230         for (int i = 0; i < max_len; i++) cout << '-';
231         cout << "-|-";
232         for (int i = 0; i < max_digits; i++) cout << '-';
233         cout << "-|-";
234         for (int i = 0; i < date_size; i++) cout << '-';
235         cout << "-|" << endl;
236     }
237 }
```

I. 입력

- list: 음식 목록 벡터

II. 결과

- 음식 목록 출력

III. 설명

- 이름의 길이와 개수의 자릿수가 음식마다 모두 다르기 때문에 길이와 자릿수에 대한 최댓값을 구한다.
- 최댓값을 기준으로 구분선을 출력하고 음식 정보도 같이 출력해서 전체 표를 출력한다.

⑤ 음식 추가

```

240 void addFood(vector<FoodInfo>& list){
241     string name;
242     int count;
243     int date;
244
245     // 사용자 입력 받기
246     name = cinName();
247     count = cinCount();
248     date = cinDate();
249
250     // 파일에 입력
251     ofstream file(file_dir, ios::app);
252     file << '\n' << name << ", " << count << ", " << date;
253     file.close();
254
255     // 벡터에 추가
256     FoodInfo line;
257     line.name = name;
258     line.count = count;
259     line.date = date;
260     list.push_back(line);
261
262     cout << name << " " << count << "개가 추가되었습니다." << endl;
263 }

```

I. 입력

- list: 음식 목록 벡터
- 사용자 입력

II. 결과

- 벡터와 파일에 음식 추가

III. 설명

- 사용자의 입력을 받아서 벡터와 파일에 새로운 음식을 추가하고 메시지를 출력한다.

⑥ 음식 제거

```

266 void deleteFood(vector<FoodInfo>& list){
267     string name;
268     int count;
269
270     // 사용자 입력 받기
271     name = cinName();
272     count = cinCount();
273
274     // 벡터에서 제거
275     vector<FoodInfo>::iterator iter = list.begin();
276     vector<FoodInfo>::iterator list_size = list.end();
277     for (auto& food : list){
278         if (name.compare(food.name) == 0){
279             // 입력한 수량이 현재 수량보다 많으면 오류 출력
280             if (count > food.count){
281                 cout << "입력한 수량이 현재 수량을 초과합니다" << endl;
282                 return;
283             }
284             else if (count < food.count){
285                 food.count -= count;
286                 break;
287             }
288             else{ // count == food.count
289                 list.erase(iter);
290                 break;
291             }
292         }
293         iter += 1;
294     }
295
296     // 입력한 음식이 벡터 내에 없으면 오류 출력
297     if (iter == list_size){
298         cout << "해당 음식이 없습니다" << endl;
299         return;
300     }
301
302     // 파일에서 제거
303     // 제거한 벡터를 파일에 덮어쓰기
304     ofstream file(file_dir);
305     for (auto food : list){
306         file << food.name << ", " << food.count << ", " << food.date << '\n';
307     }
308     file.close();
309
310     cout << name << " " << count << "개가 제거되었습니다." << endl;
311 }

```

I. 입력

- list: 음식 목록 벡터
- 사용자 입력

II. 결과

- 벡터와 파일에서 음식 제거

III. 설명

- 사용자로부터 이름과 개수를 입력 받는다.
- 입력한 개수가 현재 수량보다 많으면 오류를 출력한다.
- 입력한 개수가 현재 수량과 같으면 입력한 음식을 벡터에서 제거한

다.

- 입력한 개수가 현재 수량보다 적으면 입력한 음식의 수량을 입력한 개수만큼 감소시킨다.
- 만약 입력한 이름과 같은 이름의 음식이 벡터에 없으면 오류를 출력한다.
- 변경된 벡터를 파일에 덮어써서 변경 사항을 저장한다.

⑦ 유통기한 임박 음식 출력

```
329 void printExpiringFood(vector<FoodInfo>& list){
330     int current_date = getCurrentDate();
331     int threshold = 3; // 유통기한 임박 기준
332
333     // 유통기한 임박 음식 벡터를 생성하여 printFood() 함수로 보내준다
334     vector<FoodInfo> expiring_food_list;
335     for (auto food : list){
336         if (food.date - threshold <= current_date)
337             expiring_food_list.push_back(food);
338     }
339     printFood(expiring_food_list);
340 }
```

I. 입력

- list: 음식 목록 벡터

II. 결과

- 유통기한 임박 음식 출력

III. 설명

- 음식 목록에서 유통기한이 얼마 남지 않은 음식들을 출력한다.
- getCurrentDate() 함수로 얻은 현재 날짜를 기준으로 유통기한이 3일 남은 음식들을 새로운 벡터에 저장하고 그 벡터를 printFood() 함수로 보내서 유통기한 임박 음식 목록을 출력한다.

3) 테스트 결과

① 파일 존재 확인

```
PS C:\code\CPP2409-P> & 'c:\Users\user\Documents\Microsoft-MIEngine-In-ux5zcmjc.y2f' 'Microsoft-MIEngine-Pid-u1oo5jh3.xmd' '--dbgExe=
음식 목록 파일을 찾을 수 없습니다
프로그램을 종료합니다
PS C:\code\CPP2409-P>
```

② 기능 출력

```
PS C:\code\CPP2409-P> & 'c:\Users\user\.vscode\Microsoft-MIEngine-In-tl1f2eeh.5sq' '--stdout=MIEngine-Pid-pbvkrersh.mdd' '--dbgExe=C:\msys64\
```

1. 현재 음식 보기
2. 음식 추가
3. 음식 제거
4. 유통기한 임박 음식 보기
5. 추천 메뉴 보기
6. 프로그램 종료

③ 기능 선택

```
사용할 기능을 선택하십시오(기능 보기: 0):
```

④ 음식 목록 출력

```
사용할 기능을 선택하십시오(기능 보기: 0): 1
```

egg	10	20241111
lettuce	10	20241105
sausage	6	20241111
meat	2	20241113

```
사용할 기능을 선택하십시오(기능 보기: 0):
```

⑤ 음식 추가

```
사용할 기능을 선택하십시오(기능 보기: 0): 2
음식 이름: fish
개수: 4
유통기한: 20241115
fish 4개가 추가되었습니다.
사용할 기능을 선택하십시오(기능 보기: 0):
```

사용할 기능을 선택하십시오(기능 보기 : 0): 1

egg	10	20241111
lettuce	10	20241105
sausage	6	20241111
meat	2	20241113
fish	4	20241115

사용할 기능을 선택하십시오(기능 보기 : 0):

```

food_list.csv
1 egg, 10, 20241111
2 lettuce, 10, 20241105
3 sausage, 6, 20241111
4 meat, 2, 20241113
5
6 fish, 4, 20241115

```

- 음식 제거 기능에서 벡터를 파일에 덮어쓸 때 반복문으로 "{이름}, {개수}, {날짜}wn"을 파일에 쓰기 때문에 마지막에 빈 줄이 하나 나온다. 이 상태에서 음식 추가 기능으로 새로운 음식을 추가하면 위의 이미지와 같이 한 줄이 비어진 채로 음식이 추가되지만 이는 이후에 파일을 읽거나 쓸 때 아무 문제도 일으키지 않는다.

⑥ 음식 제거

```
사용할 기능을 선택하십시오(기능 보기: 0): 3
음식 이름: fish
개수: 1
fish 1개가 제거되었습니다.
사용할 기능을 선택하십시오(기능 보기: 0):
```

```
사용할 기능을 선택하십시오(기능 보기: 0): 1
|-----|----|-----|
| egg      | 10  | 20241111 |
|-----|----|-----|
| lettuce  | 10  | 20241105 |
|-----|----|-----|
| sausage  | 6   | 20241111 |
|-----|----|-----|
| meat     | 2   | 20241113 |
|-----|----|-----|
| fish     | 3   | 20241115 |
|-----|----|-----|
사용할 기능을 선택하십시오(기능 보기: 0):
```

```
food_list.csv
```

```
1 egg, 10, 20241111
2 lettuce, 10, 20241105
3 sausage, 6, 20241111
4 meat, 2, 20241113
5 fish, 3, 20241115
6
```



```
사용할 기능을 선택하십시오(기능 보기: 0): 3
음식 이름: fish
개수: 3
fish 3개가 제거되었습니다.
사용할 기능을 선택하십시오(기능 보기: 0):
```

```
사용할 기능을 선택하십시오(기능 보기: 0): 1
|-----|----|-----|
| egg    | 10  | 20241111 |
|-----|----|-----|
| lettuce| 10  | 20241105 |
|-----|----|-----|
| sausage| 6   | 20241111 |
|-----|----|-----|
| meat   | 2   | 20241113 |
|-----|----|-----|
사용할 기능을 선택하십시오(기능 보기: 0):
```

```
food_list.csv
1  egg, 10, 20241111
2  lettuce, 10, 20241105
3  sausage, 6, 20241111
4  meat, 2, 20241113
5
```

⑦ 유통기한 임박 음식 출력

사용할 기능을 선택하십시오(기능 보기: 0): 1

-----	----	-----
egg	10	20241124
-----	----	-----
lettuce	10	20241116
-----	----	-----
sausage	6	20241121
-----	----	-----
meat	2	20241117
-----	----	-----
potato	10	20241120
-----	----	-----
fish	4	20241118
-----	----	-----

사용할 기능을 선택하십시오(기능 보기: 0): 4

-----	----	-----
lettuce	10	20241116
-----	----	-----
meat	2	20241117
-----	----	-----
fish	4	20241118
-----	----	-----

사용할 기능을 선택하십시오(기능 보기: 0): 6
프로그램을 종료합니다

PS Z:\CPP2409-P> date

2024년 11월 15일 금요일 오전 1:03:18

PS Z:\CPP2409-P>

4. 계획 대비 변경 사항

1) 변경사항 없음

5. 프로젝트 일정

