

1) Treść zadania

Program do realizacji został wybrany z listy proponowanych zadań.

20.Restauracja

Napisać program do samodzielnego składania zamówienia w restauracji przez klientów.

Program korzysta z bazy potraw zapisanych w pliku w następującym formacie:

<id potrawy>; <Nazwa potrawy>; <kategoria potrawy>; <cena>; <opis>

baza.txt

```
1; Woda; Napoje; 5,00; Woda mineralna w szklance
2; Sok jabłkowy; Napoje; 8,00; Sok jabłkowy świeży w szklance
3; Jajecznica na boczku; Sniadania; 12; Jajecznica z 3 jajek na
  świeżym boczku
4; Tosty; Sniadanie; 8; Dwa tosty z serem i szynką
...
```

Program na podstawie tej Bazy wyświetla użytkownikowi menu w postaci wszystkich kategorii do wyboru:

```
1 Napoje
2 Śniadania
```

```
0 złoś zamówienie
```

```
R wyczyść zamówienie
```

Potem konkretną podkategorię, np.:

```
=====Napoje=====
```

```
1 Woda          5,00; Woda mineralna w szklance
2 Sok jabłkowy   8,00; Sok jabłkowy świeży w szklance
0 powrót
```

Program ma pozwolić użytkownikowi skompletować zamówienie poprzez wprowadzanie w odpowiednim menu odpowiednich liczb. Po wprowadzeniu liczby **0** w menu głównym powinien wygenerować się plik z zamówieniem klienta.:

S15_9_53.txt

```
Stolik nr 15
9:53
Sok jabłkowy      8,00
Tosty             8,00
Tosty             8,00
=====
24,00
```

Program uruchamiany jest z linii poleceń z wykorzystaniem następującego przełącznika:

-s numer stolika, dla którego składane jest zamówienie

2) Analiza rozwiązania

Głównym celem programu jest sporządzenie i zapisanie do pliku tekstowego zamówienia klienta restauracji. Plik z danymi wejściowymi, gdzie znajduje się lista produktów ma ustalony schemat, dlatego napiszę algorytm, dzięki któremu będę mógł zapisać każdy produkt i jego dane w strukturze, a struktury umieszczyć w dynamicznej tablicy.

Po odczytaniu danych z pliku i zamknięciu go będę mógł bezproblemowo korzystać i mieć dostęp do każdego elementu, dzięki czemu będę mógł stworzyć menu, gdzie użytkownik będzie mógł wybrać kategorię i poszczególny produkt.

Do nawigacji po menu skorzystam z metod w klasie, która będzie reprezentowała całą moją aplikację.

Użyję kolejnej struktury danych, która będzie zawierała nazwę i cenę produktu, który użytkownik będzie mógł dodać do swojego zamówienia. Takie struktury będę umieszczać w dynamicznej tablicy, z której przy opcji czyszczenia zamówienia lub kompletowania go i zapisywania do pliku.

Do zapisu zamówienia do pliku oprócz parametru przekazywanego do funkcji main() podczas uruchomienia programu, przechowującego numer stolika użyję także biblioteki <ctime>, która pozwoli mi zapisać godzinę do pliku, w której użytkownik złożył zamówienia.

3) Specyfikacja zewnętrzna

Program należy uruchomić z konsoli w następujący sposób z poziomu miejsca, gdzie znajduje się plik uruchamiający:

```
./order-app.exe -s [NR_STOLIKA]
```

Lub podać pełną ścieżkę do programu, jeśli nie uruchamiamy go z poziomu miejsca, gdzie się znajduje.

Po uruchomieniu program pokaże menu główne, gdzie zostaną wyświetlone kategorie pobrane z pliku dane.txt i opcje dodatkowe.

```
RESTAURACJA ABC
|-----| MENU |-----|

1 Napoje
2 Obiady
3 Sniadania

0 Złoz zamówienie
R Wyczyszc zamówienie

Wprowadz numer: |
```

Należy wpisać odpowiednią cyfrę/znak, aby móc korzystać z programu. Jeśli wpisana zostanie inna cyfra/znak, program wyświetli komunikat **[ERR] Nieprawidłowy znak. Spróbuj ponownie!** I będzie można ponownie wpisać cyfrę/znak.

Po wybraniu odpowiedniej kategorii wyświetlona zostanie lista produktów. Nawigacja wygląda tak samo. Możemy dodawać produkty do zamówienia wpisując odpowiednie cyfry. Jeśli chcemy wrócić do menu głównego wystarczy wpisać '0'.

```
|-----| Napoje |-----|

1 Woda                    5 Woda mineralna w szklance
2 Sok jablkowy            8 Sok jablkowy swiezy w szklance
3 Cola-Cola              8.5 Napoj gazowany Coca-Cola
4 Mirinda                8.5 Napoj gazowany o smaku pomaranczy
5 Herbata                6 Parzona herbata w filizance
0 Powrot

Wprowadz numer: 1
[+] Woda 5PLN
Wprowadz numer: 4
[+] Mirinda 8.5PLN
Wprowadz numer: 3
[+] Cola-Cola 8.5PLN
Wprowadz numer: 5
[+] Herbata 6PLN
Wprowadz numer: |
```

Z poziomu menu głównego istnieje także możliwość wyczyszczenia zamówienia, jeśli doszło do pomyłki.

Opcja „Złóż zamówienie” wyświetla krótką informację i tworzy nowy plik tekstowy, z skompletowanym zamówieniem w tym samym miejscu, gdzie uruchamiany jest program.

```
[#] Zamowienie 4 produktow do stolika nr 10 zostalo skompletowane.
```

4) Specyfikacja wewnętrzna

Do sterowania programem wykorzystywana jest klasa **OrderApp** z pomocą funkcji **main()**, która tworzy obiekt (aplikację) i przekazuje parametr startowy oraz pomocnicze struktury danych **Products** (zebranie zmiennych pojedynczego produktu) i **Order** (zebranie zmiennych pojedynczego zamówionego produktu)

```
class OrderApp
```

Korzysta z prywatnych danych dotyczących danych przeczytanych z pliku wejściowego z bazą produktów i realizowanego zamówienia.

Konstruktor domyślny przyjmujący parameter uruchamiający z funkcji main z numerem stolika:

```
OrderApp(std::string table) : table(table) {};
```

```
void init()
```

Ta metoda jako jedyna jest wywoływana w funkcji main(). Ma za zadanie rozpoczęcie działania poprzez uruchomienie metod load_categories i load_data.

```
void load_categories()
```

Otwiera plik tekstowy dane.txt i z wykorzystaniem **std::set**, by uniknąć duplikatów zapisuje kategorie produktów

```
void load_data()
```

Otwiera plik tekstowy dane.txt i z wykorzystaniem struktury **products** oraz wektora zapisuje wszystkie produkty. Dane są odczytywane z pomocą getline() i wyróżnianie za pomocą separatora, w przypadku domyślnym jest to „;”.

```
void showMenu()
```

Korzysta z std::set, by wyświetlić za pomocą instrukcji wyboru switch listę kategorii.

```
void openCategory(std::string c)
```

Za pomocą przekazanego stringa z metody **showMenu()** z nazwą kategorii wyświetla produkty przypisane do danej kategorii. Korzysta z pętli do..while, która prowadzi po użyciu odpowiedniego znaku z powrotem do metody **showMenu()**. Dla estetyki wyświetlenia produktów wykorzystywane są modyfikatory **std::cout.flags/width**

```
void completeOrder()
```

Metoda wywoływana jest bezpośrednio z menu głównego(**showMenu()**) -> switch. Pobiera czas i przekształca go na czas lokalny -> godziny i minuty za pomocą <ctime>. Korzysta z nr stolika podanego przy uruchamianiu programu i wektora wybranych produktów, by zapisać dane do pliku.

5) Testowanie

Test I:

Program został uruchomiony w następujący sposób: `./order-app.exe`

```
[ERR] Nie podano parametrow startowych.  
  
Uzyj  
-s [NR_STOLIKA], aby zlozyc zamowienie.
```

Test II:

Program został uruchomiony w następujący sposób: `./order-app.exe -s`

```
[ERR] Nie podano parametrow startowych.  
  
Uzyj  
-s [NR_STOLIKA], aby zlozyc zamowienie.
```

Test III:

Program został uruchomiony w następujący sposób: `./order-app.exe -s 1 1`

```
[ERR] Podano za duzo parametrow startowych.  
  
Uzyj  
-s [NR_STOLIKA], aby zlozyc zamowienie.
```

Test IV:

Wpisano inną cyfrę/znak niż wypisane w menu

```
Wprowadz numer: 5  
[ERR] Nieprawidlowy znak. Sprobuj ponownie!  
Wprowadz numer: |
```

Test V:

Użycie innego separatora niż „;” w pliku z danymi

```
Nie udalo sie odczytac danych z bazy
```

