

Gliwice, 21.05.2017r.

Laboratorium programowania komputerów PK4

Sprawozdanie z projektu

Temat projektu: Program kucharski

Autor: Arkadiusz Pilarski
Kierunek: Informatyka
Semestr 4
Grupa 1, sekcja 1
Prowadzący: dr inż. Grzegorz
Kwiatkowski

1. Temat

Tematem zadanego projektu było napisanie programu, który pełniłby rolę pomocnika kucharskiego, przechowującego bazę przepisów i zwracającego te, które użytkownik może przyrządzić w oparciu o posiadane przez siebie składniki. Program przechowywałby listę dziesięciu ostatnio wybranych dań, z której możliwy byłby bezpośredni dostęp do odpowiadających im przepisów.

2. Analiza, projektowanie

2.1 Algorytmy i struktury danych

W programie zostały użyte kontenery danych vector oraz lista. Kontenery vector służą do przechowywania informacji o posiadanych przez użytkownika składnikach oraz gromadzeniu propozycji dań możliwych do przyrządzenia w oparciu o te składniki. Zastosowanie listy służy tworzeniu listy dziesięciu ostatnio wybranych dań. Algorytmy znajdujące się w programie są algorytmami przeszukującymi (własnymi), napisanymi w taki sposób, aby przeszukiwanie bazy danych przebiegało bezbłędnie i płynnie. Wszelkie wprowadzania danych przez użytkownika zostały opatrzone funkcją kontrolującą poprawność tych danych, co zapobiega nieprzewidzianemu zachowaniu programu.

2.2 Analiza obiektowa

W programie zostały utworzone własne klasy, których dokładny opis znajduje się w punkcie im poświęconym. W dokumencie znajduje się również diagram klas, przedstawiający zastosowaną technikę dziedziczenia oraz zawierania.

3. Specyfikacja zewnętrzna

Program działa w oparciu o utworzoną z plików tekstowych bazę danych. Znajdują się w niej następujące pliki:

- Baza składników – wykorzystywany do weryfikacji wprowadzanych przez użytkownika składników. Przechowuje on nazwy wszystkich składników występujących w przepisach.
- Instrukcja – zawierający instrukcję użytkowania programu, którą można wyświetlić z poziomu menu głównego wybierając odpowiednią opcję.
- Ostatnie – wykorzystywany do tworzenia listy ostatnich dziesięciu wybranych dań. Dania te są tutaj zapisywane i stąd odczytywane.
- Spis kategorii – przechowujący nazwy kategorii dań dostępnych w bazie. Jego zawartość wyświetlana jest w momencie wybierania kategorii w trakcie działania programu.
- Tabela – tutaj przechowywany jest przelicznik gramatur, pomocny podczas wprowadzania gramatury posiadanych przez siebie składników.
- Pliki o nazwach tożsamyh z nazwami kategorii dań – pliki te zawierają odpowiednio sformatowane informacje o daniach, które się w danej kategorii znajdują. Są one wykorzystywane w czasie wyszukiwania propozycji – dopasowywania składników posiadanych przez użytkownika do wymaganych w przepisie.
- Pliki o nazwach tożsamyh z nazwami dań – każdy przepis znajduje się w osobnym pliku tekstowym.

Celem niniejszego programu kucharskiego jest pomoc użytkownikowi w znalezieniu dania, które będzie on w stanie przyrządzić w oparciu o posiadane przez siebie składniki. Aby skorzystać z wyszukiwarki dań zawartej w tym programie, należy w pierwszej kolejności utworzyć listę posiadanych przez siebie składników. W procesie tworzenia tej listy pomaga użytkownikowi Kreator listy składników. Dostęp do niego uzyskuje się wybierając w menu głównym opcję "Wyszukaj dania", a następnie podając kategorię, w której chcemy szukać. Menu Kreatora zawiera wskazówki i polecenia, które pozwolą w łatwy sposób sporządzić listę swoich składników. Należy wprowadzić wszystkie składniki, które mają być użyte do przygotowania dania, wraz z ich gramaturami. Dodawane składniki są automatycznie weryfikowane, tzn.:

- Program sprawdza, czy dodawany składnik występuje w bazie;
- Program sprawdza, czy dodawany składnik nie widnieje już na liście;
- Program sprawdza, czy gramatura dodawanego składnika została podana w gramach.

Użytkownik jest na bieżąco informowany o poprawności wprowadzanych przez siebie danych poprzez odpowiednie komunikaty wyświetlane na ekranie. Po utworzeniu listy opcja "wyszukaj propozycji dań" rozpocznie proces przeszukiwania bazy danych programu celem znalezienia wszystkich dań z wybranej kategorii, do których przygotowania należy użyć wszystkich podanych przez użytkownika składników. Następnie wyświetlone zostanie okno ze znalezionymi daniami w postaci propozycji. W tym miejscu użytkownik może dokonać wyboru i wyświetlić przepis interesującej go propozycji poprzez podanie nazwy dania. Wyświetlenie przepisu otwiera przed użytkownikiem następujące możliwości:

- Może on wybrać dany przepis, co spowoduje dopisanie dania do listy ostatnio wybranych dań;
- Może on powrócić do widoku wyszukanych propozycji;
- Możliwy jest również powrót do menu głównego bez dokonywania wyboru.

Z pozycji menu głównego możliwe jest również:

- Wyświetlenie listy ostatnio wybranych dań oraz jej modyfikacja. Dodatkowo, z poziomu tej listy, można wyświetlić przepis interesującego nas dania znajdującego się na tej liście, lub spoza niej (jeśli tylko występuje w bazie);
- Wyświetlenie przelicznika gramatur, pomocnego przy wprowadzaniu gramatur składników. Jest on również dostępny z poziomu Kreatora listy składników;
- Wyświetlenie instrukcji użytkowania programu;
- Wyjście z programu.

4. Specyfikacja wewnętrzna

4.1 Struktury i zmienne

W programie stworzona została struktura danych o nazwie `Zmienne`, która przechowuje zmienne wykorzystywane przez funkcję `Znajdz_Propozycje`.

```
struct Zmienne
{
    string bufor;
    int liczba_skladnikow;
    int czas_przygotowania;
    int trafione_skladniki;
    int liczba_porcji = 0;
    int ostateczna_liczba_porcji = 0;
    int gram;
    bool znaleziono = false;
    char znak;
    bool pierwsze_trafienie;
    fstream plik;
};
```

Opis tych zmiennych:

- bufor – bufor pobierający dane z pliku.
- liczba_skladnikow - określa liczbę składników potrzebnych do przygotowania dania.
- czas_przygotowania – określa czas przygotowania dania.
- trafione_skladniki – określa ile składników, z tych podanych przez użytkownika, znajduje się w danym przepisie.
- liczba_porcji – tymczasowa liczba porcji na podstawie podanego składnika.
- ostateczna_liczba_porcji – liczba możliwych porcji na podstawie podanych składników.
- gram – przechowuje gramaturę pobraną z pliku.
- znaleziono – zmienna pomocnicza przy czytaniu pliku.
- znak – bufor dla znaków.
- pierwsze_trafienie – zmienna pomocnicza : gramatura pierwszego trafionego składnika (wtedy wartość true) określa końcową liczbę porcji, kolejne trafienia (zmiana wartości na false) mogą ją zmodyfikować.
- plik – umożliwia wykonywanie operacji na pliku.

Zmienne wykorzystywane w funkcji main:

```
int wybor_menu = 0;
bool powrot_do_menu = false;
string kategoria;
Lista_Skladnikow lista_skladnikow;
Propozycja propozycja;
Lista_Propozycji lista_propozycji;
```

Opis tych zmiennych:

- wybor_menu – zmienna nawigacyjna (przechowuje wybór opcji z menu głównego).
- powrot_do_menu – zmienna zawierająca informacje o woli powrotu do menu.
- kategoria – zmienna przechowująca nazwę wybranej kategorii
- lista_skladnikow – zmienna do przechowywania składników wprowadzanych przez użytkownika.
- propozycja – zmienna obrazująca propozycje dania wyszukanego przez program.
- lista_propozycji - Zmienna do przechowywania znalezionych propozycji.

Pozostałe zmienne zostały opisane przy okazji opisywania funkcji, w których występują.

4.2 Funkcje

4.2.1 Funkcje główne programu

➤ Wczytywanie_skladnikow

```
Lista_Skladnikow Wczytywanie_skladnikow(Lista_Skladnikow & lista, bool & powrot_do_menu);
```

Funkcja ta odpowiada za utworzenie listy posiadanych przez użytkownika składników. Wprowadza następujące zmienne:

```
Skladnik skladnik;  
string nazwa_skladnika;  
int gramatura_skladnika;  
bool znaleziono;  
int decyzja = 2;
```

Opis tych zmiennych:

- skladnik – obiekt reprezentujący pojedynczy składnik w programie.
- nazwa_skladnika – zmienna przechowująca nazwę składnika pobieraną od użytkownika.
- gramatura_skladnika – zmienna przechowująca gramaturę składnika podaną przez użytkownika.
- znaleziono – zmienna przechowująca informację o tym, czy podany składnik występuje w bazie.
- decyzja – zmienna nawigacyjna, określająca wybraną przez użytkownika opcję z dostępnych w menu kreatora.

Funkcja wykorzystuje także funkcje sprawdzające poprawność wprowadzanych przez użytkownika informacji. Opis tych funkcji znajduje się dalej w dokumencie. Są to:

- Kontrola_decyzji
- Kontrola_skladnika
- Kontrola_gramatury

➤ Wyczysc_listy

```
void Wyczysc_listy(Lista_Propozycji & lista_propozycji, Lista_Skladnikow & lista_skladnikow);
```

Funkcja ta odpowiada za usunięcie przechowywanych danych w postaci listy propozycji oraz listy składników.

➤ Znajdz_Propozycje

```
void Znajdz_Propozycje(string & kategoria, Lista_Skladnikow & lista_skladnikow,
Propozycja & propozycja, Lista_Propozycji & lista_propozycji);
```

Funkcja ta odpowiada za przeszukanie bazy danych i znalezienie dań, których przepisy zawierają składniki podane przez użytkownika. Zmienna wprowadzana przez tę funkcję to obiekt struktury Zmienne.

Wykorzystywane w niej funkcje, których opis znajduje się dalej w dokumencie, to:

- Otworz_kategorie
- Pobierz_zmienne

4.2.2 Funkcje wypisujące dane

➤ Wyniki

```
void Wyniki(Lista_Propozycji & lista_propozycji);
```

Funkcja ta wyświetla wyniki wyszukiwania oraz odpowiada za realizowanie decyzji podejmowanych przez użytkownika w oparciu o zmienną nawigacyjną. Wprowadza następujące zmienne:

```
Lista_Ostatnich lista_ostatnich;
string danie_do_otwarcia;
fstream plik;
int wybor = 1;
```

Opis tych zmiennych:

- lista_ostatnich – obiekt reprezentujący listę ostatnio wybranych dań, umożliwiający wykonywanie na niej operacji.
- danie_do_otwarcia – zmienna przechowująca nazwę dania podaną przez użytkownika.
- plik – zmienna do pracy z plikiem.
- wybor – zmienna nawigacyjna, przechowująca decyzję użytkownika.

Wykorzystywane w niej funkcje, których opis znajduje się dalej w dokumencie, to:

- Wyswietl_propozycje
- Wypisz_przepis
- Decyzja_propozycje

➤ Wyświetl_propozycje

```
void Wyświetl_Propozycje(Lista_Propozycji & lista_propozycji, fstream & plik, string & danie_do_otwarcia);
```

Zadaniem tej funkcji jest wypisanie utworzonej listy z propozycjami dań, które zostały wyszukane na podstawie wprowadzonych do programu składników. Umożliwia również wybranie dania, którego przepis zostanie wyświetlony.

➤ Wypisz_przepis

```
void Wypisz_przepis(fstream & plik, string danie_do_otwarcia);
```

Funkcja wypisująca przepis z pliku.

➤ Wyświetl

```
void Wyświetl(int & name);
```

Funkcja wypisująca z pliku dziesięć ostatnio wybranych dań lub przelicznik gramatur, zależnie od wartości zmiennej name.

4.2.3 Funkcje kontroli poprawności wprowadzanych danych

➤ Kontrola_skladnika

```
void Kontrola_skladnika(Lista_Skladnikow & lista, string & nazwa_skladnika, bool & znaleziono, int & decyzja);
```

Funkcja sprawdzająca poprawność podanej przez użytkownika nazwy składnika. Wprowadza następujące zmienne:

```
bool pomocnicza = false;  
string bufor;  
ifstream plik1;
```

Opis tych zmiennych:

- pomocnicza – zmienna pomocnicza przy wyświetlaniu menu kreatora.
- bufor – bufor przechowujący nazwy składników pobierane z pliku.
- plik1 – zmienna do pracy z plikiem.

Funkcja ta korzysta również z funkcji Dodawanie_skladnika_menu, która wypisuje menu dodawania składnika.

➤ Kontrola_gramatury

```
void Kontrola_gramatury(Lista_Skladnikow & lista, string & nazwa_skladnika, int & gramatura_skladnika);
```

Funkcja sprawdzająca poprawność podanej przez użytkownika gramatury składnika. W razie podania gramatury w niepoprawnym formacie wyświetlany jest odpowiedni komunikat, po czym następuje kolejna próba podania gramatury. Funkcja ta wprowadza następujące zmienne:

```
string gram;  
regex wzor_gramatury("^[1-9]([0-9]*g$)");
```

Opis tych zmiennych:

- gram – zmienna przechowująca wprowadzoną przez użytkownika gramaturę.
- wzor_gramatury – przechowuje wzór poprawnego formatu gramatury.

➤ Kontrola_decyzji

```
void Kontrola_decyzji(int & x, Lista_Skladnikow & lista_skladnikow);
```

Funkcja ta za również za pomocą wyrażenia regularnego kontroluje poprawność wprowadzanych danych. W przypadku podania opcji nieprzewidzianej przez program, wyświetlany jest odpowiedni komunikat, a następnie proces pobierania danych powtarza się.

4.2.4 Funkcje pobierania zmiennych

➤ Pobierz_zmienne

```
void Pobierz_zmienne(struct Zmienne & zmienna);
```

Funkcja pobiera odpowiednie wartości zmiennych z pliku.

➤ Wybierz_kategorie

```
void Wybierz_kategorie(string & kategoria);
```

Funkcja odpowiadająca za pobranie od użytkownika nazwy wybranej przez niego kategorii oraz sprawdzenie jej poprawności. Jeśli podana zostanie niepoprawna nazwa – program wyświetli odpowiedni komunikat i użytkownik zostanie poproszony o ponowne podanie nazwy.

➤ **Pobierz_zmienna_nawigacyjna**

```
void Pobierz_zmienna_nawigacyjna(int & x);
```

Funkcja pobiera od użytkownika i sprawdza poprawność zmiennej nawigacyjnej wybor_menu. W przypadku podania nieznanej opcji, wyświetlany jest odpowiedni komunikat.

➤ **Decyzja_propozycje**

```
void Decyzja_propozycje(int & wybor);
```

Funkcja pobiera zmienną przechowującą decyzję użytkownika co do wyświetlonego przepisu.

4.2.4 Funkcje pomocnicze

➤ **Kreator_pomocnicza**

```
void Kreator_pomocnicza(Lista_Skladnikow & lista);
```

Funkcja podtrzymująca widok kreatora w razie wprowadzenia niepoprawnych danych.

➤ **Ostatnie_dania_menu**

```
void Ostatnie_dania_menu(int & wybor_menu);
```

Funkcja wyświetla listę ostatnich dziesięciu wybranych dań i dostępne na niej operacje.

➤ **Ostatnie_dania**

```
void Ostatnie_dania(int & wybor_menu);
```

Za pomocą funkcji Ostatnie_dania_menu wyświetla listę, a następnie odpowiada za pobranie polecenia od użytkownika i wykonywanie operacji na liście.

➤ **Dodawanie_skladnika_menu**

```
void Dodawanie_skladnika_menu();
```

Wyświetla menu dodawania składnika.

➤ **Brak_wynikow**

```
void Brak_wynikow();
```

Wyświetla informację o braku dań zawierających daną kombinację składników.

➤ **menu_glowne**

```
void menu_glowne();
```

Wyświetla menu główne.

➤ **Animacja**

```
void Animacja();
```

Wyświetla animację ładowania danych.

➤ **Decyzja**

```
void Decyzja();
```

Wyświetlenie operacji dostępnych w kreatorze listy składników.

4.2.5 Funkcje otwierające pliki

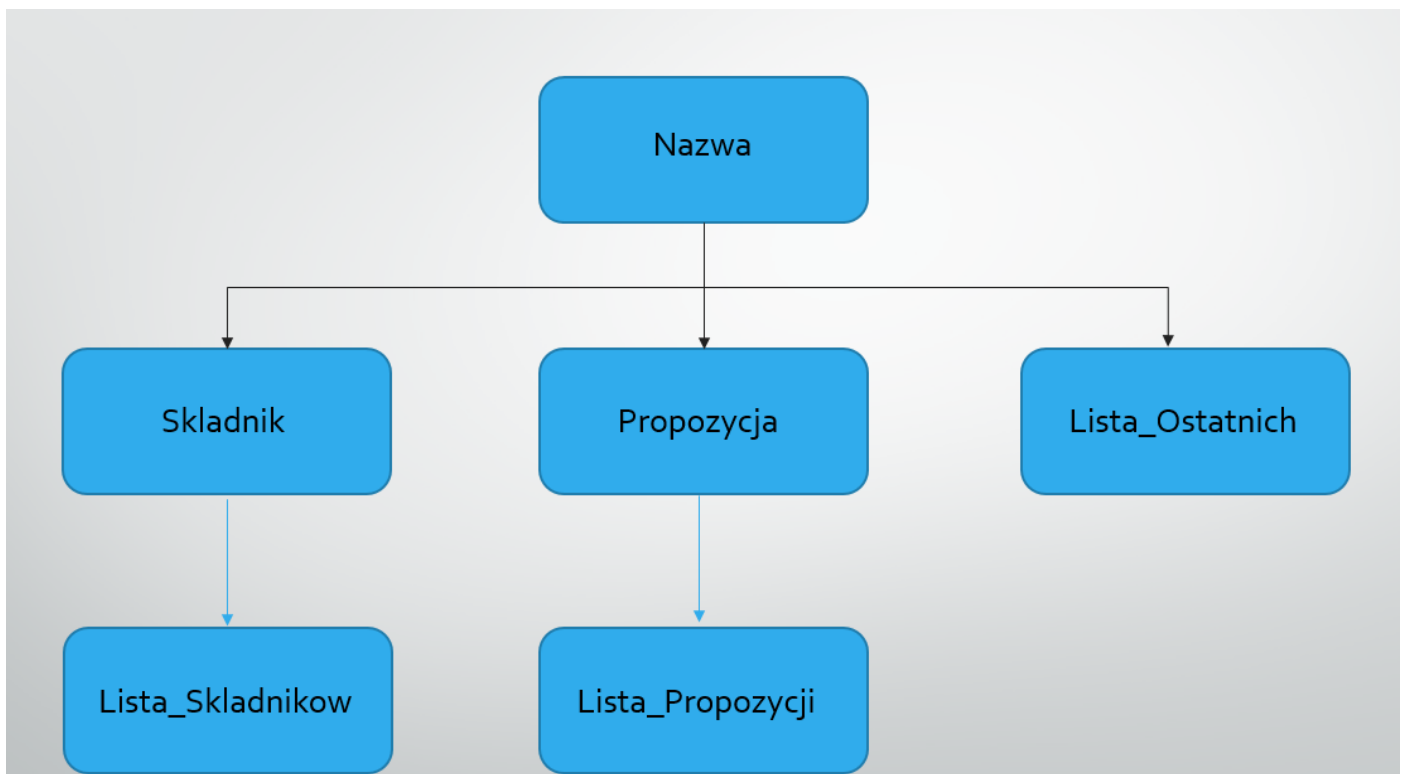
```
void Otworz_kategorie(string & kategoria, fstream & plik);  
void Otworz_przepis(fstream & plik1, string & nazwa);  
void Otworz_spis_kategorii(fstream & plik);  
void Otworz_instrukcje();
```

4.3 Klasy

- **Nazwa** – bazowa klasa przechowująca zmienną string, umożliwiającą nadawanie nazwy obiektom.
- **Składnik** – klasa dziedzicząca publicznie po klasie Nazwa, zawierająca dodatkowo zmienną int symbolizującą gramaturę składnika. Klasa ta reprezentuje jeden składnik w bazie – jego nazwę oraz gramaturę. Dostarcza do programu następujące operacje:
 - Wypisz_składnik – wypisanie danych o składniku.
 - Definiuj_składnik – Tworzenie składnika o podanych parametrach.
 - Pobierz_gramature – funkcja zwracająca gramaturę składnika.
- **Lista_Składnikow** – klasa zawierająca kontener vector Składników, dostarczająca następujące operacje:
 - Dodaj_składnik – dodanie składnika do listy.
 - Usun_składnik – usunięcie składnika z listy.
 - Wypisz_składniki – wypisanie wszystkich składników znajdujących się w kontenerze.
 - Rozmiar – zwraca liczbę składników w kontenerze.
 - Czysc_liste – usuwa wszystkie składniki z kontenera.
 - Czy_unikatowy – funkcja sprawdza, czy dodawany składnik nie znajduje się już w kontenerze.
- **Propozycja** – klasa dziedzicząca publicznie po klasie Nazwa, reprezentująca pojedynczą propozycję dania, wyszukaną w bazie na podstawie podanych składników. Zawiera dwie zmienne int, odpowiednio czas i liczba_porcji, które przechowują informację na temat czasu przewidywanego na wykonanie dania oraz liczbę porcji, którą będzie można przyrządzić z posiadanych składników. Klasa dostarcza następujące operacje:
 - Czas_Przygotowania – zwraca czas potrzebny do przygotowania dania.
 - Liczba_Porcji – zwraca liczbę porcji możliwych do przygotowania.
 - Wypisz_Propozycje – wypisuje propozycję.
 - Definiuj_propozycje – tworzy propozycję na podstawie przekazanych danych.
- **Lista_Propozycji** – klasa zawierająca kontener Propozycji, dostarczająca następujące operacje:
 - Dodaj_propozycje – dodaje propozycję do kontenera.
 - Wypisz_propozycje – wypisuje w odpowiednim formacie wszystkie propozycje znajdujące się w kontenerze.
 - Czysc_liste – usuwa wszystkie elementy w kontenerze.
 - Rozmiar – zwraca rozmiar kontenera, czyli liczbę zgromadzonych w nim propozycji.

- **Lista Ostatnich** – klasa dziedzicząca publicznie po klasie Nazwa, służąca obsłudze listy ostatnio wybranych dań. Zawiera listę tych dań oraz zmienną string do przechowywania tytułu tej listy. Dostarcza następujące operacje:
- Wczytaj_liste – wczytuje listę z pliku.
 - Zapisz – zapisuje zaktualizowaną listę do pliku.
 - Dodaj – dodaje danie do listy.
 - Czyszczenie_listy – usuwa wszystkie elementy listy.

4.4 Diagram klas



5. Testowanie

Program został przetestowany pod względem :

- Działania podczas wprowadzania poprawnych danych w różnych kombinacjach.
- Poprawnego zachowania w sytuacji, kiedy wprowadzone zostaną dane nieprzewidziane przez program.
- Wycieków pamięci.

Przeprowadzenie testów pozwoliło wykryć wady funkcji kontrolujących poprawność wprowadzania danych i przeprowadzić odpowiednią ich modyfikację. Nie wykryto wycieków pamięci.

Przykładowe działanie prezentują poniższe grafiki:

- 1) Uruchomienie programu:

```
Witaj w programie kucharskim!  
  
MENU  
1. Wyszukaj dania  
2. Pokaz ostatnio wybrane dania  
3. Wswietl instrukcje uzytkowania  
4. Pokaz przelicznik gramatur  
5. Zakoncz program  
  
Podaj numer operacji, ktora chceszwykonac i zatwierdz enterem.  
-->_
```

- 2) Wybranie opcji Wyszukaj dania:

```
Wybierz kategorie dan z dostepnych w bazie:  
1. Zupy  
2. Makarony  
3. Na slodko  
4. Sniadanie  
5. Bezglutenowe  
  
Podaj nazwe wybranej kategorii: _
```

- 3) Wybór kategorii:

```
Wybierz kategorie dan z dostepnych w bazie:  
1. Zupy  
2. Makarony  
3. Na slodko  
4. Sniadanie  
5. Bezglutenowe  
  
Podaj nazwe wybranej kategorii: Sniadanie  
Wybrana kategoria: Sniadanie  
Uruchamianie kreatora listy skladnikow ...
```

4) Kreator listy składników:

```
KREATOR LISTY SKŁADNIKÓW

Twoje składniki:

Dostępne operacje:
4 -> wyszukaj propozycji dan
3 -> przelicznik gramatur
2 -> dodaj składnik do listy
1 -> usun ostatnio dodany składnik
0 -> wroc do menu glownego programu

Wprowadz numer operacji, ktora chcesz wykonac i zatwierdz enterem.
-->_
```

5) Dodawanie składnika do listy:

```
KREATOR LISTY SKŁADNIKÓW

Twoje składniki:

Dostępne operacje:
4 -> wyszukaj propozycji dan
3 -> przelicznik gramatur
2 -> dodaj składnik do listy
1 -> usun ostatnio dodany składnik
0 -> wroc do menu glownego programu

Wprowadz numer operacji, ktora chcesz wykonac i zatwierdz enterem.
-->2

DODAWANIE SKŁADNIKA

W celu dodania składnika wpisz jego nazwe i zatwierdz enterem.
Jesli chcesz anulowac dodawanie składnika, wpisz 'anuluj'.

Nazwa składnika: jajka
Podaj gramature: 240g_
```

6) Zatwierdzenie listy i przejście do wyszukiwania propozycji dań:

```
KREATOR LISTY SKŁADNIKÓW

Twoje składniki:
jajka 240g

Dostępne operacje:
4 -> wyszukaj propozycji dan
3 -> przelicznik gramatur
2 -> dodaj składnik do listy
1 -> usun ostatnio dodany składnik
0 -> wroc do menu glownego programu

Wprowadz numer operacji, ktora chcesz wykonac i zatwierdz enterem.
-->4
```

7) Wyszukiwanie propozycji:

```
Szukanie propozycji dan...
```

8) Znalezione propozycje:

```
Wyszukane propozycje dan :  
  
Propozycja 1:  
Nazwa: Czosnkowa jajecznic z cukinia  
Czas przygotowania: 15min  
Mozliwa liczba porcji: 1  
  
Propozycja 2:  
Nazwa: Jajecznic na tostach  
Czas przygotowania: 15min  
Mozliwa liczba porcji: 1  
  
Propozycja 3:  
Nazwa: Marchewkowo-kukurydziane placuszki  
Czas przygotowania: 30min  
Mozliwa liczba porcji: 2  
  
Propozycja 4:  
Nazwa: Pelnoziarniste pancakes z batatami  
Czas przygotowania: 15min  
Mozliwa liczba porcji: 4
```

9) Wyświetlenie przepisu:

```
Aby wyswietlic przepis konkretnego dania, podaj jego nazwe.  
Jesli chcesz wrocic do menu, wpisz menu.  
  
Nazwa dania: Jajecznic na tostach
```

```
JAJECZNICA NA TOSTACH  
  
Potrzebne skladniki:  
chleb 140g  
maslo klarowane 80g  
maslo 40g  
jajko 120g  
smietana 15g  
natka pietruszki 12g  
szczypiorek 5g  
ser gruyere 30g  
  
Przepis:  
1. Na patelni na srednim ogniu, rozgrzej maslo klarowane i gdy bedzie juz bardzo gorace wloz chleb.  
2. Smaz z obu stron do ladnego zarumienienia, po czym wyjmij z patelni i owin w recznik papierowy aby nie ostygl.  
3. Na sredniej patelni rozgrzej na malym ogniu maslo, wlej roztrzepane jajka i smaz caly czas mieszajac na malym ogniu.  
4. Po 3-4 minutach jajka powinny byc juz usmazone, jezeli lubisz bardziej scieta, smaz przez 5-6 minut.  
5. Pod koniec smazenia wmieszaj smietanke, dopraw sola i pieprzem i wmieszaj musztarde, ziola i ser.  
6. Rozloz tosty na talerze i wyloz na nie jajecznic i posyp pietruszka.  
  
Wprowadz:  
-> 2 aby wybrac danie  
-> 1 aby wrocic do propozycji dan  
-> 0 aby wrocic do menu glownego  
-->
```


10) Dodanie przepisu do listy ostatnio wybranych i wyświetlenie jej:

```
10 ostatnio wybranych dan:
1. Jajecznica na tostach
2. Gazpacho
3. Pasta jajeczna
4. Krem z brokułow
5. Zupa gulaszowa
6. Krem z pieczonej papryki
7. Placuszki bezglutenowe z bananem
8. Mini-pizza z baklazana
9. Bananowo-jablkowe babeczki gryczane
10. Owsianka z masłem orzechowym i bananem

Wpisz:
'reset'      - aby wyczyszcic liste
'menu'       - aby wrocic do menu
'wyswietl'   - aby wyswietlic wybrany przepis
-->
```

6. Podsumowanie

Praca nad programem pozwoliła udoskonalić pracę z plikami tekstowymi, zastosować wiedzę na temat klas w praktyce i stworzyć samodzielny produkt, którego bazę danych można łatwo poszerzać, a z programu wygodnie korzystać. Napisane algorytmy działają poprawnie, czas wykonywania wszystkich operacji jest na tyle szybki, że nie jesteśmy w stanie dostrzec żadnych opóźnień.