

Gliwice, 16.06.2022

Programowanie II

Projekt zaliczeniowy

"Co zjeść?"

Aleksandra Kacprzak, Tomasz Bury gr. 1/1

1. Opis projektu.

Aplikacja „Co zjeść?” ma na celu pomóc w codziennym wyborze posiłków. Poprzez podanie przez użytkownika składników, jakie ma w domu, program wyszukuje w bazie posiłków, które są możliwe do stworzenia. Jednocześnie, użytkownik może wyświetlić dokładny przepis wykonania danej potrawy. Dzięki możliwości dodawania własnych przepisów, usuwania ich oraz wyświetlania pełnej bazy użytkownik może mieć wszelkie swoje ulubione przepisy w jednym miejscu.

2. Wymagania

Program zawiera następujące funkcjonalności:

- Możliwość wyświetlania przepisów wpisanych do bazy.
- Dodawanie własnych przepisów.
- Usuwanie istniejących przepisów.
- Dynamiczne przygotowywanie listy składników możliwych do wybrania przez użytkownika na podstawie przepisów wprowadzonych do programu. Dzięki temu można mieć pewność, że dla każdego składnika wyświetlanego na liście można znaleźć przynajmniej jedną zawierającą go potrawę.
- Możliwość wybierania przez użytkownika składników, które posiada i w następstwie generowanie listy potraw, które są możliwe do wykonania z nich.
- Wyświetlenie przepisu dla wybranej potrawy z wygenerowanej listy możliwych posiłków.
- Interfejs graficzny, który ułatwia obsługę aplikacji.

3. Przebieg realizacji

Projekt „Co zjeść?” składa się z następujących plików:

- main.cpp – główny plik .cpp zawierający funkcję main(), w której wywołana jest funkcja początkowa programu. Dołączony jest również plik mainwindow.h oraz biblioteka QApplication.
- functions.h – plik nagłówkowy zawierający deklaracje funkcji wykorzystywanych w programie, klasę recipes oraz biblioteki potrzebne do działania programu.
- mainwindow.h – plik nagłówkowy z klasą QMainWindow odpowiadającą za GUI.

Opis bibliotek użytych w projekcie zawarty jest w poniższej tabeli.

Nazwa biblioteki	Opis biblioteki	Co zostało użyte w programie?
iostream	Standardowa biblioteka wyjścia i wejścia.	Strumień wejścia i wyjścia: cout, cin, i powiązane z nimi funkcje (przykładowo czyszczące strumień).
fstream	Biblioteka pozwalająca na odczyt plików i zapis do nich.	Zmienne ifstream oraz ofstream, otwieranie plików oraz zapis do nich a także zamykanie plików.
regex	Biblioteka służąca do obsługi wyrażeń regularnych, za pomocą których możliwe jest walidowanie łańcuchów znakowych.	Zmienna regex, do której wpisane zostało wyrażenie regularne, zmienna smatch, która jest niezbędna do funkcji regex_match, której celem jest stwierdzenie czy dany łańcuch znakowy spełnia warunki wyrażenia regularnego.
QApplication	Biblioteki wynikające z użycia framework'u Qt odpowiedzialne za obsługę GUI.	Zostały użyte funkcje odpowiadające za programowanie przycisków, wyświetlanie GUI oraz wyświetlanie okien dialogowych.
QVBoxLayout		
QCheckBox		
QFrame		
QLabel		
QMessageBox		
QFont		

- recipesBase.txt – plik tekstowy pełniący funkcję bazy. Zapisane są w nim: nazwa potrawy, składniki potrzebne do wykonania jej oraz przepis. Każdy kolejny „rekord” jest w kolejnej linii. Schemat każdej linii prezentuje się następująco:

1. name='nazwaPotrawy' ingredients=[skladnik1,skladnik2] recipe='przepis'

- mainwindow.cpp – plik zawierający definicje wszystkich funkcji wykorzystywanych w programie oraz zawierający bibliotek wynikające z użycia frameworka Qt (opisanych powyżej). Krótki opis wszystkich głównych funkcji, z pominięciem funkcji odpowiadających za działanie GUI, jest w tabeli poniżej.

Nazwa funkcji	Opis funkcji
vector<string> separateBy(string str, char by, bool deleteSpaces)	Rozdziela podany łańcuch tekstowy, który jest podzielony określonym znakiem. Trzeci argument określa, czy w danym tekście usuwamy spacje czy zamierzamy je pozostawić. Zwraca ona wektor, w którym zawarte są podzielone fragmenty (zmienne typu string).
vector<recipes> analyseRecipeFile(fstream &file)	Analizuje plik z przepisami i porządkuje powiązane z nimi informacje do obiektów klasy recipes. Obiekty te zawierają nazwę, składniki oraz przepis. Przewiduje ona możliwość pustego pliku i w takiej sytuacji zwraca pusty

	wektor. W innym przypadku analizuje plik i linijka po linijce tworzy obiekt, które następnie zapisuje w wektorze. Zwraca ona wektor z kolejnymi obiektami klasy recipes.
void addRecipe()	Pozwala ona dodawać użytkownikowi własne przepisy. Pyta użytkownika kolejno o nazwę przepisu, składniki oraz przepis, które następnie zapisuje w odpowiedni sposób w pliku z przepisami.
void deleteRecipe()	Pozwala ona usuwać użytkownikowi przepisy. Wyświetla listę zawierającą nazwy przepisów i pyta się użytkownika, który chce usunąć. Następnie szuka posiłku o danej nazwie i jeżeli go odnajdzie – usuwa go, w przeciwnym przypadku zgłasza komunikat „Nie znaleziono takiej potrawy”. Funkcja ta działa na zasadzie tworzenia nowego pliku, w którym zapisane są wszystkie linie z pliku pierwotnego, poza tą zawierającą usuwany przepis. Pierwotny plik jest usuwany, a następnie zmieniana jest nazwa nowego pliku na „recipesBase.txt”.
vector <string> searchIngredients()	Tworzy listę składników, które są używane w przepisach. Zwraca wektor zawierający zmienne typu string z kolejnymi składnikami,
vector <string> chooseIngredients(vector <string> Ingredients)	Pyta użytkownika o numery składników, które ten posiada u siebie w domu. Następnie za pomocą wyrażeń regularnych sprawdza poprawność podanego przez użytkownika ciągu znaków. Gdy wejście jest poprawne funkcja zwraca wektor zawierający zmienne string, w których zapisane są wybrane składniki, natomiast w przypadku odwrotnym program wraca do menu głównego.
vector <recipes> searchRecipes(vector <string> chosenIngredients)	Szuka przepisów, które są możliwe do zrobienia z podanych przez użytkownika składników. Jeżeli nie znajdzie żadnego – wyświetla komunikat z taką informacją, natomiast w innym przypadku wypisuje możliwe posiłki i pyta się użytkownika, czy chciałby zobaczyć sposób wykonania któregoś z nich. W przypadku niepoprawnego wejścia program wraca do menu głównego, natomiast, gdy jest poprawne – wyświetla przepis i zwraca wektor z obiektami zawierającymi możliwe przepisy.
void searchingRecipes()	Zbiera funkcje służące do wygenerowania możliwych posiłków z podanych składników. Zawiera funkcje: searchIngredients, chooseIngredients, searchRecipes.

Na początku pliku `mainwindow.cpp` użyta jest instrukcja `using namespace std`, która określa używaną w pliku przestrzeń nazw oraz dołączone są pliki nagłówkowe wraz z bibliotekami Qt.

5. Podsumowanie i wnioski.

Wszystkie założenia dotyczące programu zostały spełnione. Zarówno główny moduł programu, czyli generowanie posiłków, jak i dodatkowe opcje jak dodawanie przepisów, usuwanie ich oraz wyświetlanie udało się zrealizować.

Największym problemem, na który napotkano w czasie tworzenia programu, okazał się trud związany z poznaniem zasad i specyfiki tworzenia aplikacji przy użyciu frameworku Qt.

Program ten jest przydatny dla każdego, kto nie potrafi się zdecydować, jaki posiłek zrobić – może być on dobrym rozwiązaniem dla każdego, kto chciałby stworzyć własną bazę przepisów i w prosty sposób znajdować te, które może wykonać w danej chwili.