Agnieszka Janowicz

Nr indeksu 246399

TKOM

**Biblioteka do parsowania plików XML aplikacji Cocktails DIY**

**1. Wstęp**

Przedmiotem realizacji jest biblioteka do parsowania plików XML, ograniczona do przetwarzania plików XML o strukturze odpowiadającej rekordom bazy danych koktajli aplikacji Cocktails DIY (w przygotowaniu).

Aplikacja korzysta z plików XML zawierających dane koktajlu o następującej strukturze:

*<cocktail>*

*<name>Nazwa koktajlu</name>*

*<image>Ścieżka do zdjęcia koktajlu</image>*

*<recipe>*

*<base>Opis przygotowania – baza drinka</base>*

*<optional>Opis przygotowania – opcjonalne</optional>*

*</recipe>*

*<ingredients>*

*<ingredient>*

*<name>Nazwa skladnika</name>*

*<quantity>Ilosc</quantity>*

*<important>Czy skladnik jest istotny</important>*

*</ingredient>*

*</ingredients>*

*</cocktail>*

**2. Gramatyka opisana w notacji EBNF**

litera = ? litera łacińskiego alfabetu ?

cyfra = ‘0’ | ‘1’ | ‘2’ | ‘3’ | ‘4’ | ‘5’ | ‘6’ | ‘7’ | ‘8’ | ‘9’

znakinterpunkcyjny = ? znak interpunkcyjny lub spacja oprócz ‘<’, ‘>’?

znak = litera | cyfra | znakinterpunkcyjny

ciąg = {znak}

komentarz = { ‘<!—‘, ciąg, ‘-->’ }

metadane = ‘<?’, ciąg, ‘?>’

nazwaKoktajlu = [komentarz], '<name>', ciąg, '</name>'

obraz = [komentarz], '<image>', ciąg, '</name>'

przepisBazowy = [komentarz],‘<base>’, ciąg, ‘</base>’

przepisOpcjonalny = [komentarz],‘<optional>’, ciąg, ‘</optional>’

przepis = [komentarz],‘<recipe>’, przepispodst, przepisopcjonalny, ‘</recipe>’

składnikNazwa = [komentarz],‘<name>’, ciąg, ‘</name>’

składnikIlość = [komentarz],‘<quantity>’, ciąg, ‘</quantity>’

składnikWazność = [komentarz],‘<important>’, ciąg, ‘</important>’

składnik = [komentarz]‘<ingredient>’, { skladnikNazwa, skladnikIlość, skladnikważność},‘</ingredient>’

przepis = '<recipe>', przepisBazowy, przepisOpcjonalny, ' </recipe>'

składniki = '<ingredients>', { składnik}, '</ingredients>'

daneKoktajlu = nazwaKoktajlu, obraz, przepis, składniki

plik = [metadane], [komentarz] ‘<cocktail>’, daneKoktajlu,‘</cocktail>’

**3. Moduły biblioteki**

**1) Moduł obsługujący parser i skaner**

Moduł wywołujący parser i skaner oraz zbierający wyniki tych operacji. Zapewnia również obsługę błędów (wypisuje odpowiedni komunikat w konsoli z podaniem dokładnej lokalizacji: numeru linii w pliku oraz kolumny).

Moduł udostępnia funkcję parse() przyjmującą jako argumenty nazwę pliku do parsowania oraz obiekt typu Cocktail, do którego jest przypisywany koktajl stworzony poprzez parsowanie pliku. Zwraca kod wykonania (1 jeśli operacja przebiegła poprawnie, 0 w przeciwnym wypadku).

Moduł korzysta z modułów analizy leksykalnej i syntaktycznej do przetwarzania pliku.

**2) Moduł analizy leksykalnej**

Interpretuje znaki otrzymane z modułu obsługującego pliki, grupując je w token, i zwraca token modułowi analizy syntaktycznej.

Moduł udostępnia funkcje scan\_begin() i scan\_end().

Tokeny rozpoznawane przez moduł: 'yes', 'no', ‘<cocktail>’, ‘</cocktail>’, ‘<ingredients>’, ‘</ingredients>’, ‘<ingredient>’, ‘</ingredient>’, ‘<quantity>’, ‘</quantity>’, ‘<important>’, ‘</important>’, ‘<name>’, </name>', ‘<image>’, '</image'>, '<base>', '</base>', '<optional>', '</optional>', '<recipe>', '</recipe>', ciąg (ciąg znaków zdefiniowany w gramatyce).

**3) Moduł analizy syntaktycznej**

Sprawdza otrzymane tokeny pod kątem poprawności – sprawdza, czy wszystkie elementy opisu koktajlu są zawarte w pliku. Wprowadza dane do struktur. Elementy nienależące do założonej struktury pliku (tzn. różniące się od struktury podanej w punkcie 1.) zostają zignorowane.

Moduł dołączony jest do modułu analizy leksykalnej.

**4) Moduł główny**

Moduł zawierający API biblioteki. Komunikuje się z modułem obsługującym parser i skaner. Umożliwia dodanie i modyfikację nowych wpisów koktajli oraz zapisanie ich w standardzie XML podanym w punkcie 1. Biblioteka udostępnia również API do obsługi elementów typu Cocktail i Ingredient (dodawanie, modyfikację i zapis do pliku).

API udostępniane przez bibliotekę:

- bool exportCocktailToFile(std::string filename);

- void createCocktail(std::string);

- void createCocktail(std::string, std::string, std::string, std::string);

- void setName(std::string name);

- void setImagePath(std::string path);

- void setBaseRecipe(std::string base);

- void setOptionalRecipe(std::string optional);

- void addIngredient(std::string name, std::string quantity, bool important);

- void addIngredient(Ingredient \*i);

- std::string getName();

- std::string getImagePath();

- std::string getBaseRecipe();

- std::string getOptionalRecipe();

- std::list<Ingredient\*> getIngredientList();

- Cocktail\* getCocktail();

**5) Inne**

Biblioteka wykorzystuje trzy pomocnicze moduły – stack, position i location – Open Source'owe narzędzia Bison udostępnione na licensji GPL 3.0. Moduły te są wykorzystywane do wykrywania aktualnej pozycji w pliku (do obsługi błędów).

**4. Przewidywane struktury danych**

* struktura Cocktail – struktura zawierająca pola opisujące dane koktajlu
  + nazwa koktajlu
  + ścieżka do obrazu
  + opis bazowy
  + opis opcjonalny
  + lista składników (elementów typu Ingredient)
* struktura Ingredient – struktura zawierająca pola opisujące dane składnika koktajlu
  + nazwa składnika
  + ilość
  + istotność

**5. Opis algorytmów**

**1) Analiza leksykalna**

Analiza leksykalna wykorzystuje narzędzie Flex (darmowe oprogramowanie na licensji BSD generujące skanery).

Algorytm analizujący tekst wejściowy (analizator leksykalny) działa na zasadzie automatu skończonego. Konkretne znaki w tekście przetwarzanym będą powodowały przejście do określonego stanu. Wynikiem tej analizy będzie lista tokenów.

**2) Analiza syntaktyczna**

Po analizie leksykalnej nastąpi analiza syntaktyczna – następuje sprawdzenie, czy wszystkie wymagane dane są zawarte w pliku oraz sprawdzana jest ich poprawność.

Na tym etapie wszelkie dane wczytane z pliku, które nie są zawarte w zbiorze wymaganych wpisów (komentarze, metadane, elementy posiadające dodatkowe znaczniki) zostają pominięte w procesie kopiowania danych do struktur i zostają usunięte.

Analiza syntaktyczna przeprowadzana jest przy pomocy dodatkowego narzędzia Bison (w wersji 3.0.2). Oprogramowanie to jest otwarte i udostępniane na licensji GPL 3.0.

W razie wystąpienia błędu działanie programu zostanie przerwane i zostanie wygenerowany komunikat o błędzie (wraz z podaniem lokalizacji w pliku, w której wystąpił błąd).

**6. Wymagania funkcjonalne i niefunkcjonalne**

**1) Wymagania funkcjonalne**

* wczytanie pliku XML

Program wczytuje plik XML o nazwie określonej przez użytkownika o strukturze i gramatyce opisanej w punktach 1. I 2. Zapisuje wczytane dane w odpowiedniej strukturze w pamięci.

* modyfikacja pobranych danych

Program umożliwia modyfikację pól struktury opisującej koktajl.

* stworzenie struktury Coctail

Program umożliwia stworzenie struktury typu Coctail i uzupełnienie jej.

* zapisanie pliku XML

Program umożliwia zapisanie struktury Coctail do pliku XML o strukturze określonej w punkcie 1.

**2) Wymagania niefunkcjonalne**

* program potrafi wczytać plik XML pochodzący z innych źródeł, jeśli zawiera wymagane pola opisane w punkcie 1.

**7. Przykład testowy**

*<?xml version="1.0" encoding="UTF-8"?>*

*<cocktail>*

*<!—Pomyslec nad nazwa -->*

*<name>Srubokret</name>*

*<image>path</image>*

*<recipe>*

*<base>Wymieszaj wodke i sok. Mozesz dodac plasterek pomaranczy.</base>*

*<optional>Udekoruj szklanke plasterkiem pomaranczy.</optional>*

*</recipe>*

*<ingredient>*

*<name>wodka</name>*

*<quantity>60ml</quantity>*

*<important>yes</important>*

*</ingredient>*

*<ingredient>*

*<name>sok pomaranczowy</name>*

*<quantity>150ml</quantity>*

*<important>yes</important>*

*</ingredient>*

*<ingredient>*

*<name>pomarancza</name>*

*<quantity>plasterek</quantity>*

*<important>no</important>*

*</ingredient>*

*</cocktail>*

**7. Opis testowania**

Zawarte tu przykłady testów były przeprowadzane w następujący sposób: program testujący wywołuje odpowiednią funkcję CocktailParsera, a następnie wypisuje kod wykonania operacji oraz dane wyczytane ze struktury Cocktail oraz zapisuje przepis w nowym pliku. Następnie stworzony plik był parsowany przez program w celu sprawdzenia poprawności konstrukcji.

1) Dodane komentarze w wielu miejscach

Zawartość parsowanego pliku:

*<? metadata ?>*

*<cocktail>*

*<-- komentarz -->*

*<name>Srubokret</name>*

*<-- komentarz -->*

*<image>path</image>*

*<-- komentarz -->*

*<recipe>*

*<-- komentarz -->*

*<base>Wymieszaj wodke i sok. Mozesz dodac plasterek pomaranczy.</base>*

*<-- komentarz -->*

*<optional>Udekoruj szklanke plasterkiem pomaranczy.</optional>*

*</recipe>*

*<-- komentarz -->*

*<ingredients>*

*<-- komentarz -->*

*<ingredient>*

*<-- komentarz -->*

*<name>wodka</name>*

*<-- komentarz -->*

*<quantity>60ml</quantity>*

*<-- komentarz -->*

*<important>yes</important>*

*</ingredient>*

*<ingredient>*

*<name>sok pomaranczowy</name>*

*<quantity>150ml</quantity>*

*<important>yes</important>*

*</ingredient>*

*<ingredient>*

*<name>pomarancza</name>*

*<quantity>plasterek</quantity>*

*<important>no</important>*

*</ingredient>*

*</ingredients>*

*</cocktail>*

Import successfull

Name: Srubokret

Image path: path

Base recipe: Wymieszaj wodke i sok. Mozesz dodac plasterek pomaranczy.

Optional recipe: Udekoruj szklanke plasterkiem pomaranczy.

Ingredients: 3

wodka 60ml 1

sok pomaranczowy 150ml 1

pomarancza plasterek 0

2. Dodatkowe nieprawidłowe znaczniki

*<? metadata ?>*

*<cocktail>*

*<name>Srubokret</name>*

*<image>path</image>*

*<nieprawidlowy>znacznik</znacznik>*

*<recipe>*

*<base>Wymieszaj wodke i sok. Mozesz dodac plasterek pomaranczy.</base>*

*<optional>Udekoruj szklanke plasterkiem pomaranczy.</optional>*

*</recipe>*

*<ingredients>*

*<ingredient>*

*<name>wodka</name>*

*<quantity>60ml</quantity>*

*<important>yes</important>*

*</ingredient>*

*<ingredient>*

*<nieprawidlowy>znacznik</nieprawidlowy>*

*<name>sok pomaranczowy</name>*

*<quantity>150ml</quantity>*

*<important>yes</important>*

*</ingredient>*

*<ingredient>*

*<name>pomarancza</name>*

*<quantity>plasterek</quantity>*

*<important>no</important>*

*</ingredient>*

*</ingredients>*

*</cocktail>*

*<zly>znacznik</zly>*

Import successfull

Name: Srubokret

Image path: path

Base recipe: Wymieszaj wodke i sok. Mozesz dodac plasterek pomaranczy.

Optional recipe: Udekoruj szklanke plasterkiem pomaranczy.

Ingredients: 3

wodka 60ml 1

sok pomaranczowy 150ml 1

pomarancza plasterek 0

3. Nieprawidłowa struktura - brak sekcji <ingredients>

*<cocktail>*

*<name>Srubokret</name>*

*<image>path</image>*

*<recipe>*

*<base>Wymieszaj wodke i sok. Mozesz dodac plasterek pomaranczy.</base>*

*<optional>Udekoruj szklanke plasterkiem pomaranczy.</optional>*

*</recipe>*

*<ingredients>*

*<--brak skladnikow -->*

*</cocktail>*

zwraca błąd: 11.1-11: syntax error, unexpected </cocktail>, expecting <ingredient> or string

4. Nieprawidłowa struktura - zduplikowany znacznik <name>

*<cocktail>*

*<name>Srubokret</name>*

*<image>path</image>*

*<name>Srubokret2</name>*

*<recipe>*

*<base>Wymieszaj wodke i sok. Mozesz dodac plasterek pomaranczy.</base>*

*<optional>Udekoruj szklanke plasterkiem pomaranczy.</optional>*

*</recipe>*

*<ingredients>*

*<ingredient>*

*<name>wodka</name>*

*<quantity>60ml</quantity>*

*<important>yes</important>*

*</ingredient>*

*<ingredient>*

*<name>sok pomaranczowy</name>*

*<quantity>150ml</quantity>*

*<important>yes</important>*

*</ingredient>*

*<ingredient>*

*<name>pomarancza</name>*

*<quantity>plasterek</quantity>*

*<important>no</important>*

*</ingredient>*

*</ingredients>*

*</cocktail>*

Zwraca błąd: 4.1-6: syntax error, unexpected <name>, expecting <recipe> or string

5. Nieprawidłowa struktura - odwrócona kolejność znaczników

*<cocktail>*

*<image>path</image>*

*<name>Srubokret</name>*

*<recipe>*

*<base>Wymieszaj wodke i sok. Mozesz dodac plasterek pomaranczy.</base>*

*<optional>Udekoruj szklanke plasterkiem pomaranczy.</optional>*

*</recipe>*

*<ingredients>*

*<ingredient>*

*<name>wodka</name>*

*<quantity>60ml</quantity>*

*<important>yes</important>*

*</ingredient>*

*<ingredient>*

*<name>sok pomaranczowy</name>*

*<quantity>150ml</quantity>*

*<important>yes</important>*

*</ingredient>*

*<ingredient>*

*<name>pomarancza</name>*

*<quantity>plasterek</quantity>*

*<important>no</important>*

*</ingredient>*

*</ingredients>*

*</cocktail>*

Zwraca błąd2.1-7: syntax error, unexpected <image>, expecting <name> or string

6. Opis wykorzystujący wszystkie znaki interpunkcyjne

*<cocktail>*

*<name>Srubokret</name>*

*<image>path</image>*

*<recipe>*

*<base>Opis zawierajacy znaki</base>*

*<optional>!@#$%^&\*()\_+-={}|[]\:";'?/.,</optional>*

*</recipe>*

*<ingredients>*

*<ingredient>*

*<name>wodka</name>*

*<quantity>60ml</quantity>*

*<important>yes</important>*

*</ingredient>*

*<ingredient>*

*<name>sok pomaranczowy</name>*

*<quantity>150ml</quantity>*

*<important>yes</important>*

*</ingredient>*

*<ingredient>*

*<name>pomarancza</name>*

*<quantity>plasterek</quantity>*

*<important>no</important>*

*</ingredient>*

*</ingredients>*

*</cocktail>*

Import successfull

Name: Srubokret

Image path: path

Base recipe: Opis zawierajacy znaki

Optional recipe: !@#$%^&\*()\_+-={}|[]\:";'?/.,

Ingredients: 3

wodka 60ml 1

sok pomaranczowy 150ml 1

pomarancza plasterek 0

7. Inny przykład koktajlu

*<cocktail>*

*<name>Kamikaze</name>*

*<image>path</image>*

*<recipe>*

*<base>Wymieszaj wodke i likier.</base>*

*<optional>Dodaj sok z cytryny.</optional>*

*</recipe>*

*<ingredients>*

*<ingredient>*

*<name>wodka</name>*

*<quantity>20ml</quantity>*

*<important>yes</important>*

*</ingredient>*

*<ingredient>*

*<name>Blue Curacao</name>*

*<quantity>40ml</quantity>*

*<important>yes</important>*

*</ingredient>*

*<ingredient>*

*<name>cytryna</name>*

*<quantity>1 szt.</quantity>*

*<important>no</important>*

*</ingredient>*

*</ingredients>*

*</cocktail>*

Import successfull

Name: Kamikaze

Image path: path

Base recipe: Wymieszaj wodke i likier.

Optional recipe: Dodaj sok z cytryny.

Ingredients: 3

wodka 20ml 1

Blue Curacao 40ml 1

cytryna 1 szt. 0

Test zawarty jest w pliku TestProgram.cpp. Korzysta on z API CocktailXMLParsera, zatem testuje jednocześnie funkcje dodające/ modyfikujące dane koktajlu.