

Politechnika Śląska w Gliwicach

Wydział Automatyki, Elektroniki i Informatyki



Laboratorium Programowania Komputerów

Projekt semestralny

| | |
|-------------------------------|---------------------|
| autor: | Jakub Łagódka |
| użytkownik laboratoryjny: | sroda1530ppJL |
| prowadzący: | dr inż. Piotr Pecka |
| rok akademicki: | 2018/2019 |
| kierunek: | informatyka |
| rodzaj studiów: | SSI |
| semestr: | 5 |
| termin laboratorium / ćwiczeń | środa: 15:30-17:00 |
| sekcja: | Warunek |
| termin oddania sprawozdania: | 2019-02-28 |
| data oddania sprawozdania: | 2018-11-21 |

1. Treść zadania

Napisać program który wczytuje i zlicza słowa z pliku tekstowego w języku angielskim (bez polskich liter) np. książkę z projektu Gutenberg, program ma mieć zaimplementowane dwa algorytmy wyszukiwania powtarzających się słów (wykorzystać dziedziczenie i funkcje wirtualne) - pierwszy wykorzystuje drzewo binarne (zaimplementować odpowiednią klasę), drugi wykorzystuje klasę Map z biblioteki Stl. Program ma tworzyć raport w pliku tekstowym.

2. Analiza, projektowanie

2.1. Algorytmy, struktury danych, ograniczenia specyfikacji

Program ma za zadanie zliczyć słowa nie powtarzające się w pliku tekstowym, gdzie słowa są zapisane w alfabecie angielskim (26 liter w języku ascii). W tym celu należy każde słowo porównać z tymi które wystąpiły już wcześniej. Jeśli dane słowo już się pojawiło, to można pobrać następne słowo. Jeśli natomiast nie pojawiło się nigdzie wcześniej, wtedy dopisujemy je do wybranej struktury danych. Znalezione słowa są zapisywane w jednej, wybranej poprzez parametr programu, z dwóch dostępnych struktur danych. Jeśli w parametrze podano „tree”, kolejne słowa będą zapisywane w drzewie binarnym.

Drzewo (ang. tree) jest strukturą danych zbudowaną z elementów, które nazywamy **węzłami** (ang. node). Dane przechowuje się w węzłach drzewa. Węzły są ze sobą powiązane w sposób hierarchiczny za pomocą **krawędzi** (ang. edge), które zwykle przedstawia się za pomocą strzałki określającej hierarchię. Pierwszy węzeł drzewa nazywa się **korzeniem** (ang. root node) Drzewo binarne jest to drzewo, w którym stopień każdego wierzchołka jest nie większy od 3.

Natomiast jeśli podano „map”, słowa będą zapisane w kontenerze klasy map.

W tej implementacji stworzono vector 27 kontenerów map i w zależności od pierwszej litery danego słowa słowo trafia do danego kontenera. Pozwoliło to na kilkukrotne przyspieszenie wyszukiwania powtarzających się słów.

W programie wykorzystano biblioteki: <string>, <map>, <vector>, <iostream>, <fstream>, <windows.h> oraz przestrzeń nazw std. Stworzono 6 powiązanych ze sobą klas, w tym jedną abstrakcyjną z funkcjami wirtualnymi. Do deklaracji klas użyto wzorców. Został użyty także mechanizm wyjątków. Program generuje raport w pliku tekstowym. W raporcie zostały zawarte informacje o czasie działania programu, wybranym algorytmie przechowywania danych i o liczbie znalezionych słów.

3. Specyfikacja zewnętrzna

3.1 Obsługa programu

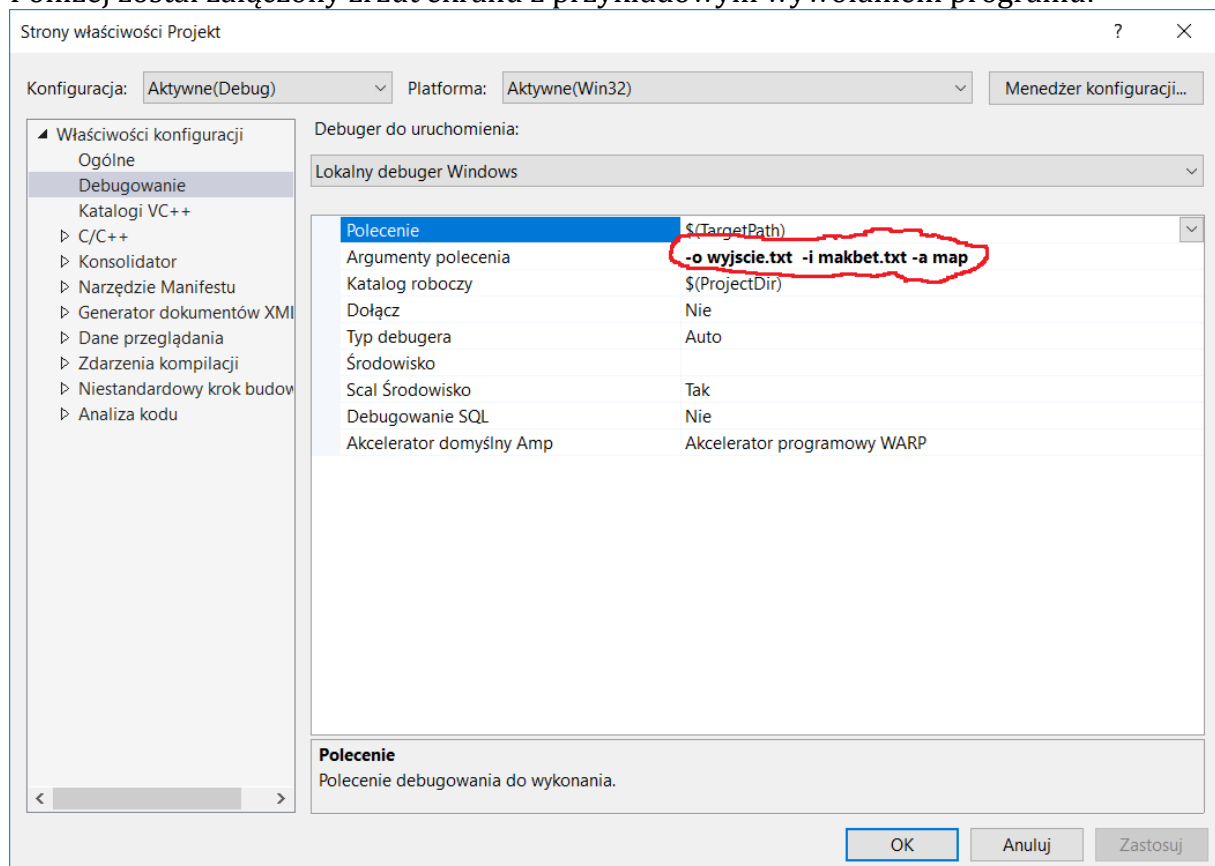
Program jest uruchamiany wraz z odpowiednimi parametrami z linii poleceń. Parametry mogą być podane w dowolnej kolejności. Parametry niezbędne do wywołania programu:

-i nazwa pliku wejściowego

-o nazwa pliku wyjściowego

-a wybrany algorytm: tree(drzewo binarne)/map(kontener map)

Poniżej został załączony zrzut ekranu z przykładowym wywołaniem programu:



W celu wyświetlenia pomocy, należy podać jedynie nazwę pliku bez żadnych parametrów lub jedynie z parametrem `-h`.

3.2 Format danych wejściowych

Program należy wywoływać z podaniem wszystkich wymienionych parametrów (i żadnych innych). Ich liczba jest sprawdzana przy uruchamianiu programu. W przypadku pliku wejściowego program sprawdza czy udało się znaleźć i utworzyć plik o podanej nazwie. Jeśli nie program wyrzuci wyjątek, wyświetli odpowiedni komunikat i zatrzyma działanie. Podobnie dla pliku wyjściowego program sprawdza czy udało go się stworzyć i zapisać w nim dane. Jeśli chodzi o wybrany algorytm, program sprawdza czy wpisano „tree” lub „map”. Jeśli nie, to program wyrzuci wyjątek i zakończy działanie.

4. Specyfikacja wewnętrzna

W głównej funkcji programu `main()`, tworzony jest obiekt klasy `file`. Wywoływane są 4 metody. Najpierw jest wywoływana metoda `read_parameters()`, która wczytuje parametry z linii poleceń i sprawdza ich poprawność. Następnie metoda `Run_program()` otwiera plik wejściowy, wczytuje słowo po słowie i tworzy odpowiednią strukturę danych. Na koniec usuwa stworzoną strukturę danych. Kolejną wywoływaną metodą jest `Make_report()`, która tworzy raport i zapisuje w nim odpowiednie informacje. Na koniec wywoływana jest metoda `Console()`, która wyświetla na konsoli informacje o zakończeniu działania programu.

W programie jest 7 powiązanych ze sobą klas.:

- Klasa `exception` odpowiada za obsługę wyjątków.
- Klasa `Parameters` zajmuje się obsługą i sprawdzeniem poprawności wczytanych parametrów.
- Klasa `Words_counter` jest klasą abstrakcyjną o metodach czysto wirtualnych.
- Klasa `Binary_tree` dziedziczy po abstrakcyjnej klasie `Words_counter`. Zajmuje się ona implementacją i obsługą struktury drzewa binarnego
- Klasa `Cointainer` dziedziczy po klasie `Words_counter`. Zajmuje się ona implementacją i obsługą kontenera klasy `map`.
- Klasa `File` dziedziczy po klasie `Parameters`, `Binary_tree` i `Cointainer`. Zajmuje się ona obsługą plików i także tutaj wykonuje się główna część programu. Dzięki dziedziczeniu posiada ona niemal wszystkie funkcjonalności innych klas.
- Klasa `Node` symbolizuje pojedynczy węzeł drzewa binarnego

5. Testowanie

Program był testowany na różnych większych i mniejszych plikach. Początkowo wyniki nie były poprawne i oba algorytmy pokazywały różne wyniki. Po paru poprawkach nie stwierdzono błędów w wynikach. Testowano dla paru książek w języku angielskim.

6. Wnioski

Program pozwolił mi na przećwiczenie umiejętności programowania obiektowego. Przede wszystkim przetestowałem w praktyce użycie klas abstrakcyjnych i dziedziczenia wirtualnego w języku C++. Bardzo użyteczny okazał się też mechanizm wyjątków i klasa kontenerów STL, która okazała się bardzo pomocna. Pisanie tego programu oka-

zało się również motywacją do samodzielnego studiowania pewnych zagadnień programistycznych, które okazało się że nie do końca poprawnie rozumiałem.