Lista jednokierunkowa

Autor	Daniel Cogiel
Data	15.01.2022

1. Analiza tematu.

Opis zadania

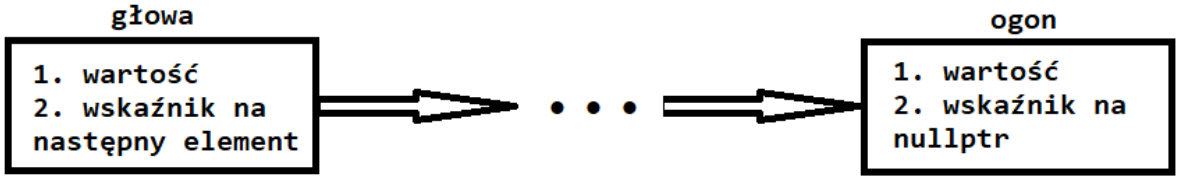
1) Należy zaimplementować listę jednokierunkową. Należy w szablonie klasy zawrzeć:

- inteligentne wskaźniki,
- konstruktory kopiujące/przenoszące, konstruktor bezargumentowy,
- destruktor
- operatory przypisania/przeniesienia,
- dodawanie elementów do kontenera,
- usuwanie wybranego elementu,
- wyszukiwanie elementu,
- sortowanie zawartości (różne kryteria sortowania, wybór czy rosnąco, czy malejąco),
- serializacja i deserializacja (zapis i odczyt z plików binarnych);

2) Należy zaimplementować dodatkowe klasy, na których będzie testowana w/w klasa (szablon). Dla tych dodatkowych klas należy dopisać metodę umożliwiającą wczytywanie z pliku tekstowego danych.

Analiza postawionego zadania

Lista jednokierunkowa będzie się składać z członów zawierających wartość danego typu oraz wskaźnik na element następny. Zostało to przedstawione na poniższym rysunku.



Do realizacji takiej struktury zostaną stworzone dwie klasy (szablony): klasa `List_element`, która reprezentuje jeden element listy oraz klasa `My_forward_list` reprezentująca całą strukturę i zawierająca dwie wartości: wskaźnik na głowę oraz wskaźnik na ogon. Umieszczenie wskaźników właśnie na te dwa elementy: początkowy i końcowy, ułatwi zaimplementowanie funkcji dodawania i usuwania elementów z początku i końca. W projekcie zostaną użyte inteligentne wskaźniki `shared_ptr`, z powodu swojej prostoty w użytkowaniu i braku potrzeby dealokowania pamięci.

2. Specyfikacja zewnętrzna programu.

Instrukcja działania

Program jest wywoływany w konsoli. Aby wczytywać do niego pliki wejściowe i wyjściowe oraz dokonać wyboru, dla jakiego typu bądź klasy chcemy testować program, zostały dodane odpowiednie przełączniki:

`-c [1;4]` – wybór, dla jakiego typu chcemy testować listę (1 – `double`, 2 – `string`, 3 – `Czlowiek`, 4 – `Samochod`)

`-ti „nazwa_pliku.txt”` – podanie nazwy wejściowego pliku tekstowego

`-to „nazwa_pliku.txt”` – podanie nazwy wyjściowego pliku tekstowego

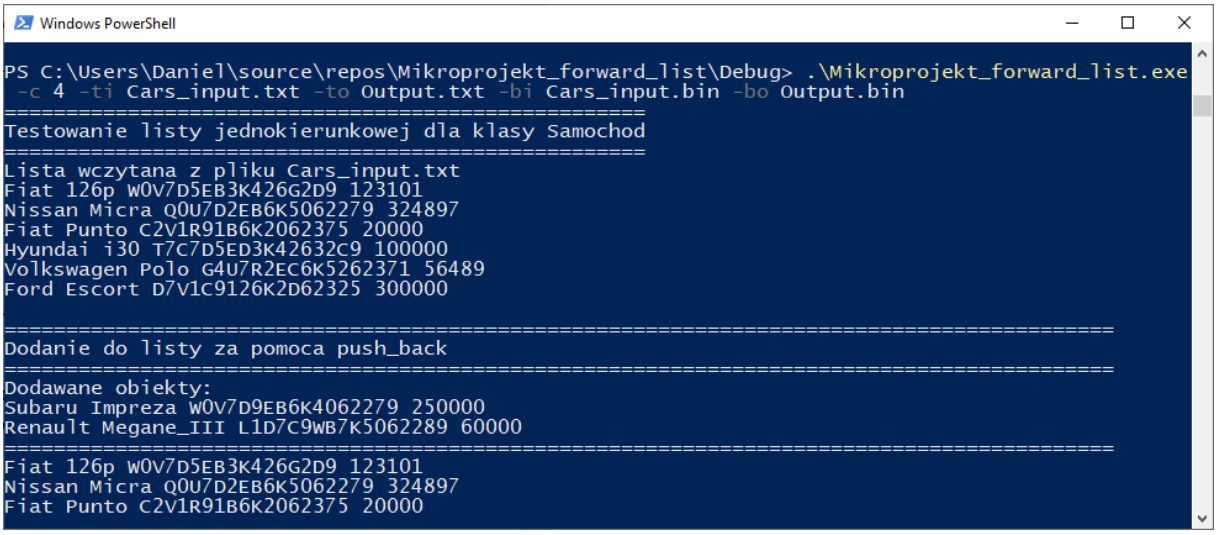
`-bi „nazwa_pliku.bin”` – podanie nazwy wejściowego pliku binarnego

`-bo „nazwa_pliku.bin”` – podanie nazwy wyjściowego pliku binarnego

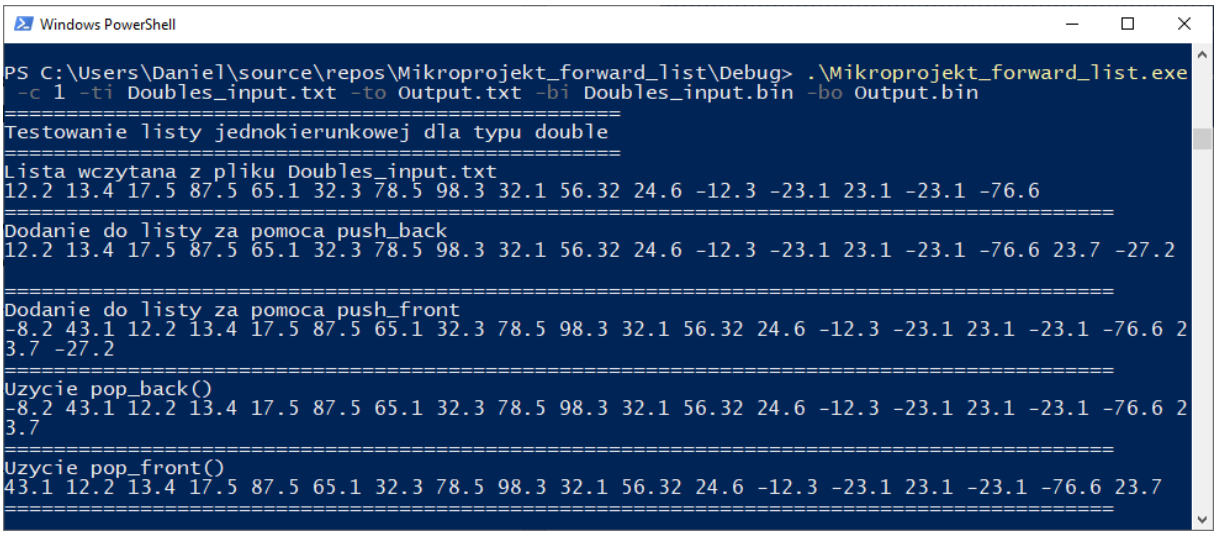
Pliki, na których będziemy testować program, należy umieścić w tym samym folderze, co plik wykonywalny .exe.

Przykłady działania

Przykład wywołania programu (testowanie dla klasy Samochod) z argumentami: `-c 4 -ti Cars_input.txt -to Output.txt -bi Cars_input.bin -bo Output.bin`



Przykład wywołania programu (testowanie dla typu double) z argumentami: `-c 1 -ti Doubles_input.txt -to Output.txt -bi Doubles_input.bin -bo Output.bin`



3. Specyfikacja wewnętrzna programu.

Ogólna struktura programu i schemat działania

W programie utworzone zostały w sumie 4 klasy: szablon klasy `List_element` reprezentujący element listy, szablon klasy `My_forward_list` reprezentujący listę, klasy `Czlowiek` i `Samochod` służące do testowania stworzonego kontenera. Dla każdej klasy zostały dodane odpowiednie metody zapewniające wymagane funkcjonalności tworzonego kontenera. Program zawiera również 5 funkcji, które służą do wczytywania odpowiednich elementów podanych jako argumenty programu dzięki przełącznikom. Po wywołaniu programu, jest on testowany dla wybranego typu (`double`, `string`, `Czlowiek` bądź `Samochod`). Prezentacja wszystkich funkcjonalności programu dla wybranego typu została zawarta w funkcji `main()` i wyświetla się po jego wywołaniu.

Szczegółowy opis klas i funkcji

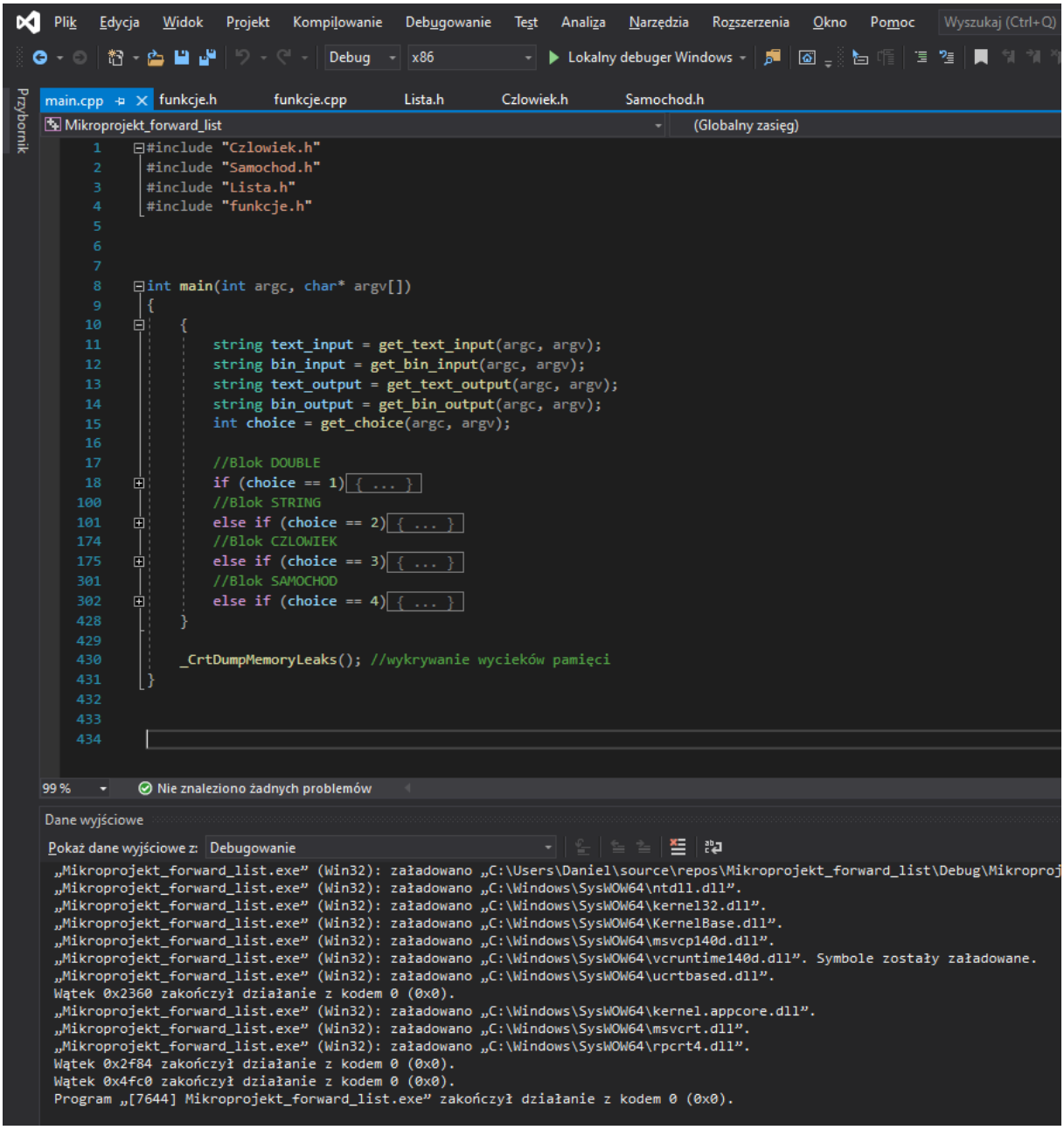
Szczegółowy opis klas i funkcji został zawarty w załączniku „Opis klas i funkcji.pdf”.

4. Testowanie i uruchamianie.

Program został przetestowany dla podstawowych typów oraz klas stworzonych specjalnie dla testowania i działał poprawnie. W przypadku gdy nie zostaną podane żadne argumenty wejściowe, test zawarty w funkcji `main()` po prostu się nie wykona. Niepodanie pliku wejściowego skutkuje tym, że program wykona test na początkowo pustym kontenerze (w czasie testowania program dodaje do listy nowe elementy, a więc lista wyjściowa będzie po prostu o wiele krótsza). Niepodanie pliku wyjściowego skutkuje tym, że program nie zapisze stanu listy do pliku.

Program został także przetestowany pod kątem wycieków pamięci przy pomocy biblioteki `<crtdbg.h>`, a konkretnie instrukcji `_CrtDumpMemoryLeaks()`, zbierającej wszystkie wycieki pamięci.

Program nie generuje żadnych wycieków ze względu na użycie inteligentnych wskaźników `shared_ptr`, co potwierdza załączony zrzut ekranu.



5. Wnioski.

Lista jednokierunkowa jest jednym z prostszych do utworzenia kontenerów, jednak jej struktura czasami utrudnia implementację niektórych funkcji, takich jak np. `sort()`, z powodu braku możliwości cofania się, takiej, jaka jest zawarta w liście dwukierunkowej.