AKADEMIA NAUK STOSOWANYCH
W NOWYM SĄCZU

Wydział Nauk Inżynieryjnych
Katedra Informatyki

DOKUMENTACJA PROJEKTOWA

ZAAWANSOWANE PROGRAMOWANIE

Algorytm listy dwukierunkowej z zastosowaniem GitHub

Autor:
Rafał Curzydło

Prowadzący:
mgr inż. Dawid Kotlarski

Nowy Sącz 2025

Spis treści

1. Ogólne określenie wymagań

1.1. Wprowadzenie

Celem projektu jest stworzenie klasy w C++, która implementuje listę dwukierunkową działającą na stercie. Dodatkowo, projekt ma na celu zapoznanie się z systemem kontroli wersji Git oraz dokumentowaniem kodu za pomocą narzędzia Doxygen.

1.2. Funkcjonalności Klasy

Klasa powinna implementować następujące metody:

- Dodaj element na początek listy.
- Dodaj element na koniec listy.
- Dodaj element pod wskazany indeks.
- Usuń element z początku listy.
- Usuń element z końca listy.
- Usuń element z pod wskazanego indeksu.
- Wyświetl całą listę.
- Wyświetl listę w odwrotnej kolejności.
- Wyświetl następny element.
- Wyświetl poprzedni element.
- Czyść całą listę.

1.3. Wymagania

Projekt powinien obejmować:

- Generowanie dokumentacji za pomocą Doxygen do formatu PDF.
- Dokumentację projektową w formacie LaTeX.
- Utworzenie repozytorium GitHub z przynajmniej pięcioma commitami, w tym:

- Cofnięcie się o dwa commity.
- Usunięcie jednego commita.
- Praca w dwóch lokalizacjach z synchronizowaniem zmian między nimi.
- Usunięcie pliku w katalogu projektu i przywrócenie go z GitHub.

1.4. Instrukcje

1.5. Instalacja

- Zainstaluj Git i stwórz konto na GitHub.
- Zainstaluj Doxygen oraz narzędzie do generowania dokumentacji.
- Używaj Git w środowisku Visual Studio lub Visual Code.

1.6. Testowanie i Praca z Git

- Wykonaj co najmniej pięć commitów.
- Wykonaj co najmniej jedno cofnięcie się o dwa commity.
- Usuń jeden commit.
- Pobierz projekt z GitHub do nowej lokalizacji i wprowadź zmiany.
- Usuń plik i przywróć go z GitHub.

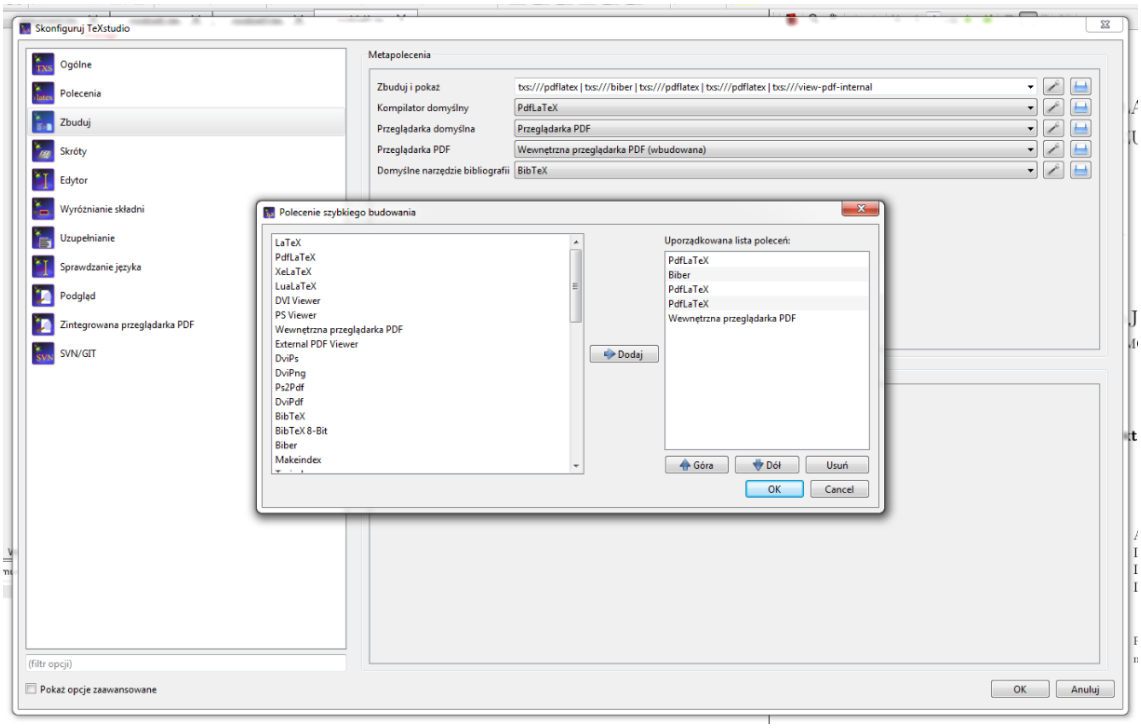
1.7. Instalacja

Poniżej są opisane kroki potrzebne do instalacji LaTeX'a oraz do używania tego szablonu.

Na początku instalujemy TeXLive[1]. Ściągamy plik instalacyjny, zajmuje około 25MB. Podczas instalacji można wybrać do zainstalowania różne kolekcje pakietów. Jeśli nie ma problemów z miejscem na dysku to można zainstalować wszystkie, wtedy nie będzie problemu z brakującymi pakietami i błędami. Po wybraniu kolekcji brakujące pliki są pobierane z internetu. Pełna instalacja programu zajmuje około 8GB. Najlepiej zostawić instalację na noc, ponieważ proces zabiera sporo czasu. Warto ustawić komputer tak, aby się nie wyłączył lub nie uśpił. Warto także przed instalacją zablokować antywirusa, ponieważ może blokować niektóre z komponentów.

[1]Instalka na stronie https://www.tug.org/texlive/acquire-netinstall.html[1].

Następnie instalujemy TEXstudio[2]. Ściągamy plik instalacyjny zajmujący około 120MB. Instalacja przebiega standardowo.



Rys. 1.1. Ustawienie TeXstudio

Następnym krokiem jest ustawienie w TEXStudio kolejności budowania projektu. Należy wybrać zakładkę: „Opcje/Konfiguruj TEXstudio...". W otwartym oknie przechodzimy na zakładkę „Zbuduj". Na rysunku 1.1 (s. 6) pokazany jest zrzut ekranu z konfiguracją. W linijce „Zbuduj i pokaż" klikamy ikonę klucza, żeby przejść do konfiguracji polecenia. W otwartym oknie ustawić kolejność tak jak pokazano na rysunku.

[2]Plik instalacyjny na stronie https://www.texstudio.org[2].

2. Analiza problemu

2.1. Wprowadzenie

Lista dwukierunkowa jest jednym z podstawowych typów struktur danych wykorzystywanych w algorytmach komputerowych. Jest to struktura, która pozwala na przechowywanie elementów w postaci węzłów, z których każdy zawiera dane oraz wskaźniki na poprzedni i następny element. Jest szczególnie użyteczna w przypadkach, gdzie wymagana jest łatwa manipulacja danymi, jak dodawanie, usuwanie czy przeszukiwanie w obu kierunkach. GitHub jest popularną platformą do hostingu kodu źródłowego i wspomagania pracy zespołowej nad projektami programistycznymi.

2.2. Zastosowanie Listy Dwukierunkowej

Lista dwukierunkowa jest wykorzystywana w wielu sytuacjach, gdzie zachodzi potrzeba dynamicznego zarządzania danymi. Oto kilka przykładów:

- **Zarządzanie historią przeglądarki internetowej:** W aplikacjach takich jak przeglądarki internetowe, lista dwukierunkowa pozwala na przechowywanie historii odwiedzonych stron. Można łatwo przemieszczać się do poprzednich i następnych stron.
- **Modelowanie struktury danych w systemach operacyjnych:** Listy dwukierunkowe są wykorzystywane w systemach operacyjnych do zarządzania procesami i zadaniami, na przykład w zarządzaniu kolejkami procesów.
- **Przechowywanie danych w grach komputerowych:** W przypadku gier komputerowych, lista dwukierunkowa może służyć do przechowywania obiektów gry, gdzie każdy obiekt jest połączony z poprzednim i następnym w kolejności.
- **Wyszukiwanie i sortowanie:** Listy dwukierunkowe są użyteczne w algorytmach sortujących oraz przy przechowywaniu elementów w kolejności, które muszą być dynamicznie modyfikowane (np. dodawanie, usuwanie).

2.3. Sposób Działania Listy Dwukierunkowej

Lista dwukierunkowa to struktura danych, której każdy element (węzeł) zawiera dwa wskaźniki: jeden wskazuje na następny element listy, a drugi na element poprzedni.

Dzięki temu możliwe jest przechodzenie zarówno do przodu, jak i do tyłu w liście.

2.3.1. Struktura Węzła

Węzeł listy dwukierunkowej zawiera trzy główne elementy:

- **Dane:** Przechowuje wartość, która może być dowolnym typem danych.
- **Wskaźnik na następny element:** Oznacza lokalizację kolejnego węzła w liście.
- **Wskaźnik na poprzedni element:** Wskazuje na poprzedni węzeł w liście.

2.3.2. Operacje na Liście Dwukierunkowej

Lista dwukierunkowa umożliwia następujące podstawowe operacje:

- **Dodawanie elementu:** Można dodać element na początku, na końcu listy lub w dowolnym miejscu, wskazując odpowiedni indeks.
- **Usuwanie elementu:** Usunięcie elementu odbywa się poprzez zmianę wskaźników poprzedzających i następujących węzłów.
- **Przechodzenie po liście:** Można przejść po wszystkich elementach od początku do końca, lub odwrotnie, z wykorzystaniem wskaźników na poprzedni i następny element.

2.3.3. Zalety Listy Dwukierunkowej

Lista dwukierunkowa oferuje szereg zalet w porównaniu do listy jednokierunkowej:

- Możliwość efektywnego przechodzenia zarówno w przód, jak i w tył.
- Szybsze usuwanie i dodawanie elementów w porównaniu do listy jednokierunkowej, zwłaszcza w przypadku operacji w środkowych częściach listy.
- Lepsza elastyczność w zarządzaniu danymi, zwłaszcza gdy konieczna jest manipulacja w obu kierunkach.

2.4. GitHub

GitHub to platforma do zarządzania kodem źródłowym, która wykorzystuje system kontroli wersji Git. GitHub pozwala na hostowanie repozytoriów, umożliwiając programistom współpracę nad kodem, śledzenie zmian oraz zarządzanie wersjami projektu.

2.4.1. Podstawowe Funkcje GitHub

- **Repozytoria:** GitHub umożliwia tworzenie repozytoriów, w których przechowywany jest kod źródłowy. Repozytoria mogą być publiczne lub prywatne.
- **Commitowanie:** Zmiany w repozytorium są zapisywane w postaci commitów. Każdy commit zawiera zmiany w kodzie oraz komunikat opisujący zmiany.
- **Branching:** GitHub pozwala na tworzenie gałęzi (branch), które umożliwiają równoległą pracę nad różnymi funkcjami projektu bez ryzyka konfliktów.
- **Pull requesty:** Pull requesty są używane do proponowania zmian w repozytorium. Po zatwierdzeniu pull requesta, zmiany są scalane z główną gałęzią projektu.
- **Wersjonowanie:** GitHub pozwala na śledzenie historii zmian w projekcie i łatwe cofanie się do wcześniejszych wersji kodu.

2.4.2. Korzyści z Używania GitHub

- **Współpraca zespołowa:** GitHub umożliwia współpracę wielu programistów nad jednym projektem, umożliwiając łatwe łączenie pracy różnych osób.
- **Kontrola wersji:** GitHub zapewnia pełną kontrolę wersji, co pozwala na śledzenie zmian i łatwe zarządzanie projektem.
- **Bezpieczeństwo:** GitHub pozwala na tworzenie prywatnych repozytoriów oraz kontrolowanie dostępu do kodu.

3. Projektowanie

3.1. Wprowadzenie

Visual Studio 2022 jest jednym z najpopularniejszych zintegrowanych środowisk programistycznych (IDE) używanych do tworzenia aplikacji w wielu językach programowania, w tym w C++. Posiada bogaty zestaw funkcji, które ułatwiają programowanie, debugowanie oraz zarządzanie wersjami kodu. Wersja 2022 tego środowiska wprowadza szereg ulepszeń w zakresie wydajności i łatwości użytkowania.

3.2. Visual Studio 2022 dla C++

Visual Studio 2022 to pełnoprawne środowisko do tworzenia aplikacji w języku C++, oferujące bogaty zestaw narzędzi i funkcji wspierających rozwój aplikacji. IDE to jest wyposażone w zaawansowany edytor kodu, debugger oraz narzędzia do profilowania wydajności aplikacji.

3.2.1. Funkcje Visual Studio 2022

- **IntelliSense:** Funkcja ułatwiająca pisanie kodu przez podpowiadanie nazw funkcji, zmiennych i składni. IntelliSense automatycznie uzupełnia kod na podstawie kontekstu.
- **Debugowanie:** Visual Studio 2022 zapewnia zaawansowane narzędzia do debugowania, w tym możliwość debugowania aplikacji lokalnych i zdalnych, ustawianie punktów przerwania oraz analizę błędów w czasie rzeczywistym.
- **Profilowanie:** Narzędzia do profilowania pomagają w optymalizacji kodu przez monitorowanie jego wydajności oraz identyfikowanie wąskich gardeł w aplikacjach.
- **Współpraca:** Integracja z GitHub oraz funkcje do zarządzania wersjami pozwalają na efektywną współpracę nad projektem z innymi programistami.

3.2.2. Domyślny Kompilator C++ w Visual Studio 2022

Domyślnym kompilatorem C++ w Visual Studio 2022 jest MSVC (Microsoft Visual C++)**, który jest częścią platformy kompilacji Microsoft. MSVC to kompilator stworzony przez firmę Microsoft, optymalizujący kod źródłowy pod kątem wydajności i zgodności z systemem Windows.

3.3. Wbudowana Funkcja Git w Visual Studio 2022

Visual Studio 2022 oferuje pełną integrację z systemem kontroli wersji Git, umożliwiając programistom zarządzanie kodem źródłowym bezpośrednio w środowisku IDE. Dzięki temu programiści mogą łatwo współpracować z zespołem i śledzić zmiany w projekcie.

3.3.1. Funkcje Git w Visual Studio 2022

- **Zarządzanie repozytoriami Git:** Visual Studio 2022 pozwala na tworzenie nowych repozytoriów, klonowanie istniejących, a także dodawanie plików i commitowanie zmian.
- **Zarządzanie gałęziami:** Programiści mogą tworzyć, przełączać się i łączyć gałęzie w repozytoriach Git bezpośrednio z poziomu IDE. Ułatwia to pracę nad różnymi funkcjonalnościami w projekcie.
- **Rozwiązywanie konfliktów:** Visual Studio 2022 automatycznie wykrywa konflikty podczas łączenia gałęzi i oferuje graficzny interfejs do ich rozwiązywania.
- **Integracja z GitHub:** Wbudowane narzędzia pozwalają na synchronizowanie lokalnego repozytorium z GitHub, co umożliwia łatwą współpracę z zespołem.
- **Śledzenie historii:** Dzięki narzędziom Git w Visual Studio 2022 można łatwo śledzić historię zmian w projekcie oraz przeglądać wcześniejsze wersje kodu.

3.4. Using Namespace std;

W języku C++, biblioteki standardowe (takie jak `iostream`, `<vector`, `<string`) są zawarte w przestrzeni nazw `std`. Aby używać funkcji, klas i obiektów z tej przestrzeni, należy wprowadzić odpowiednią przestrzeń nazw za pomocą dyrektywy `using namespace std;`.

3.4.1. Dlaczego używamy `using namespace std;`

Ułatwienie kodowania: Dyrektywa ta umożliwia bezpośrednie korzystanie z elementów przestrzeni nazw `std` bez konieczności poprzedzania ich prefiksem `std::`. Na przykład, zamiast pisać `std::cout`, można po prostu napisać `cout`.

3.5. Przykład użycia

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "Witaj,  Żwiecie !" << endl;
    return 0;
}
```

Listing 1. Przykładowy kod

4. Implementacja

```
void ListaDwuKierunkowa::wyswietl() {
    Powiazanie* aktualny = przod;
    if (!przod) {
        cout << "Lista jest pusta." << endl;
        return;
    }

    while (aktualny) {
        cout << aktualny->dane << " ";
        aktualny = aktualny->nastepny;
    }
    cout << endl;
}
```

Listing 2. Funkcja wyswietl()

Metoda 2 przechodzi przez listę od początku do końca i wypisuje dane wszystkich elementów. Jeśli lista jest pusta, informuje o tym użytkownika.

```
ListaDwuKierunkowa::ListaDwuKierunkowa() {
    przod = nullptr;
    tyl = nullptr;
}
```

Listing 3. Konstruktor ListaDwuKierunkowa

3 kawałek kodu to początek całej klasy, który inicjalizuje listę jako pustą, ustawiając oba wskaźniki (przod i tyl) na nullptr.

```
void ListaDwuKierunkowa::dodajNaPrzod(int x) {
    Powiazanie* nowe = new Powiazanie(x);
    if (!przod) {
        przod = tyl = nowe;
    }
    else {
        nowe->nastepny = przod;
        przod->poprzedni = nowe;
        przod = nowe;
    }
}
```

Listing 4. Metoda dodajNaPrzod()

Metoda 4 dodaje nowy element na początek listy. Jeżeli lista była pusta, element staje się zarówno pierwszym, jak i ostatnim. Jeśli lista nie jest pusta, nowy element zostaje umieszczony przed pierwszym.

```
void ListaDwuKierunkowa::usunIndeks(int index) {
    if (index == 0) {
        usunPrzod();
        return;
    }

    Powiazanie* aktualny = przod;
    for (int i = 0; i < index; ++i) {
        if (!aktualny->nastepny) {
            cout << "Indeks " << index << " nie istnieje." << endl;
            return;
        }
        aktualny = aktualny->nastepny;
    }

    if (!aktualny) {
        cout << "Indeks " << index << " nie istnieje." << endl;
        return;
    }

    if (aktualny == tyl) {
        usunTyl();
        return;
    }

    delete aktualny;
}
```

Listing 5. Metoda dodajNaPrzod()

Metoda 5 usuwa element na określonym indeksie. Sprawdza, czy indeks jest poprawny, i wykonuje odpowiednią operację usuwania (na początku, na końcu lub z dowolnego miejsca w środku listy).

5. Wnioski

Wnioski

Projekt polegał na zaimplementowaniu listy dwukierunkowej działającej na stercie w języku C++. Zrealizowane zostały wszystkie wymagane operacje na liście, takie jak dodawanie, usuwanie oraz wyświetlanie elementów itd. oraz utworzone zostało repozytorium GitHub.

Korzystanie z Gita i GitHub pozwoliło na łatwe śledzenie zmian w projekcie i współpracę z różnymi wersjami kodu. Dodatkowo, użycie Doxygen do generowania dokumentacji pomogło w tworzeniu opisów metod i lepszym zrozumieniu kodu.

5.1. Wnioski praktyczne

Wykonanie tego projektu pokazało:

- Jak efektywnie zarządzać pamięcią w C++,
- Jak działa lista dwukierunkowa i jak można ją zaimplementować,
- Jak używać Gita,
- Jak generować dokumentację za pomocą Doxygen.

Bibliografia

[1] *Strona internetowa TexLive*. URL: https://www.tug.org/texlive/acquire-netinstall.html.

[2] *Wikipedia (Strona główna)*. URL: https://www.wikipedia.org.

Spis rysunków

Spis tabel

Spis listingów