

AKADEMIA NAUK STOSOWANYCH W NOWYM SĄCZU

Wydział Nauk Inżynierskich
Katedra Informatyki

DOKUMENTACJA PROJEKTOWA ZAAWANSOWANE PROGRAMOWANIE

Algorytm listy dwukierunkowej z zastosowaniem GitHub

Autor:
Sebastian Tatara

Prowadzący:
mgr inż. Dawid Kotlarski

Nowy Sącz 2024

Spis treści

1. Cel projektu	3
2. Analiza problemu	4
2.1. Opis działania algorytmu	4
2.2. Opis narzędzia Git	5
3. Projektowanie	11
4. Implementacja	12
4.1. Ciekawe fragmenty kodu	14
5. Wnioski	16
Literatura	17
Spis rysunków	17
Spis tabel	18
Spis listingów	19

1. Cel projektu

Celem projektu było zapoznanie się z działaniem listy dwukierunkowej w języku C++ oraz implementacja jej na sterze z wykorzystaniem wzorców projektowych Factory i Iterator. Projekt miał na celu również nauczenie się praktycznej obsługi narzędzi Git i GitHub do kontroli wersji oraz Doxygen do generowania dokumentacji automatycznej.

Przewidywanym wynikiem projektu było:

- stworzenie działającej implementacji listy dwukierunkowej,
- możliwość dodawania, usuwania i przeglądania elementów listy,
- wygenerowanie dokumentacji w formacie PDF przy pomocy Doxygena,
- oraz zrozumienie zasad współpracy w repozytorium GitHub.

2. Analiza problemu

Lista dwukierunkowa to jedna z podstawowych dynamicznych struktur danych. Znajduje zastosowanie wszędzie tam, gdzie potrzebny jest szybki dostęp do danych w obu kierunkach — na przykład:

- w systemach zarządzania pamięcią (np. buforowanie stron),
- w edytorach tekstu (historia operacji),
- w implementacjach stosu "cofnij / ponow",
- w systemach kolejkowania z możliwością przemieszczania się po elementach.

2.1. Opis działania algorytmu

Każdy element listy, zwany węzł (Node), zawiera:

- wartość (np. liczba całkowita),
- wskaźnik na element poprzedni,
- wskaźnik na element następny

Struktura wygląda następująco:

$$\text{NULL} \leftarrow [5] \leftrightarrow [10] \leftrightarrow [15] \rightarrow \text{NULL}$$

Dzięki dwukierunkowym wskaźnikom można:

- dodawać elementy na początku lub końcu listy,
- usuwać dowolny element,
- przechodzić po liście w przód i w tył przy pomocy iteratora.

Przykład działania algorytmu (rekurencyjny opis)

1. Dodajemy element 5 → lista: [5]
2. Dodajemy element 10 → lista: [5] ↔ [10]
3. Dodajemy element 15 → lista: [5] ↔ [10] ↔ [15]
4. Usuujemy pierwszy element → lista: [10] ↔ [15]
5. Wyświetlamy od końca → [15] ↔ [10]

Tab. 2.1. Tabela przedstawia porównanie tablicy dynamicznej, listy jednokierunkowej i listy dwukierunkowej pod względem dodawania, usuwania, dostępu i kierunku przeglądania.

Struktura	Dodawanie	Usuwanie	Dostęp	Kierunek przeglądania
Tablica dynamiczna	Szybkie (na koncu)	Powolne (trzeba przesuwac dane)	Szybki (indeksy)	Jednokierunkowy
Lista jednokierunkowa	Szybkie	Szybkie	Wolny (trzeba przechodzic po kolei)	Jednokierunkowy
Lista dwukierunkowa	Szybkie	Szybkie	Wolny	Dwukierunkowy

2.2. Opis narzędzia Git

W projekcie wykorzystano Git i GitHub do kontroli wersji kodu. Dzięki Git można śledzić zmiany w kodzie, przywracać poprzednie wersje i współpracować z innymi osobami. Podczas realizacji projektu wykonano:

- co najmniej 5 commitów,
- cofnięcie o 2 commity,
- usunięcie jednego commita,
- pobranie i synchronizacja projektu z innej lokalizacji

Tworzenie commitów 1

```
Microsoft Windows [Version 10.0.19045.6456]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git init
Initialized empty Git repository in C:/Users/Sebastian/Desktop/ListaDwukierunkowa/.git/

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git add .

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   readme.md

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git commit -m "pierwszy commit"
[master (root-commit) 80eb174] pierwszy commit
 1 file changed, 1 insertion(+)
 create mode 100644 readme.md

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git remote add origin https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git'

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git'

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git branch -M main

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 240 bytes | 240.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git add .
warning: in the working copy of 'Ilist.h', LF will be replaced by CRLF the next time Git touches it

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Ilist.h

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git commit -m "stworzenie pliku Ilist.h"
[main d225b41] stworzenie pliku Ilist.h
 1 file changed, 30 insertions(+)
 create mode 100644 Ilist.h

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 566 bytes | 566.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git
 80eb174..d225b41  main -> main

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git add .
```

Rys. 2.1. Tworzenie commitów 1

Tworzenie commitów 2

```
C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git add .
warning: in the working copy of 'Node.h', LF will be replaced by CRLF the next time Git touches it

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git commit -m "stworzenie pliku Node.h"
[main dfaec14] stworzenie pliku Node.h
1 file changed, 18 insertions(+)
create mode 100644 Node.h

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 509 bytes | 509.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git
d225b41..dfaec14  main -> main

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git add .
warning: in the working copy of 'IListIterator.h', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ListIterator.h', LF will be replaced by CRLF the next time Git touches it

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git commit -m "stworzenie pliku IListIterator.h"
[main c0b43bf] stworzenie pliku IListIterator.h
2 files changed, 64 insertions(+)
create mode 100644 IListIterator.h
create mode 100644 ListIterator.h

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.01 KiB | 1.01 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git
dfaec14..c0b43bf  main -> main

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git reset --hard HEAD~1
HEAD is now at dfaec14 stworzenie pliku Node.h

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin HEAD --force
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git
+ c0b43bf...dfaec14 HEAD -> main (forced update)

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git add .
warning: in the working copy of 'IListIterator.h', LF will be replaced by CRLF the next time Git touches it

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git commit -m "stworzenie pliku IListIterator.h"
[main 7b0207a] stworzenie pliku IListIterator.h
1 file changed, 20 insertions(+)
create mode 100644 IListIterator.h

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 562 bytes | 562.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git
dfaec14..7b0207a  main -> main
```

Rys. 2.2. Tworzenie commitów 2

Tworzenie commitów 3

```
C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git add .
warning: in the working copy of 'ListIterator.h', LF will be replaced by CRLF the next time Git touches it

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git commit -m "stworzenie pliku ListIterator.h"
[main 52542eb] stworzenie pliku ListIterator.h
1 file changed, 44 insertions(+)
create mode 100644 ListIterator.h

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 735 bytes | 735.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git
7b0207a..52542eb main -> main

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git add .
warning: in the working copy of 'DoublyLinkedList.h', LF will be replaced by CRLF the next time Git touches it

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git commit -m "stworzenie pliku DoublyLinkedList.h"
[main 5a1bf2e] stworzenie pliku DoublyLinkedList.h
1 file changed, 48 insertions(+)
create mode 100644 DoublyLinkedList.h

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 746 bytes | 746.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git
52542eb..5a1bf2e main -> main

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git add .
warning: in the working copy of 'DoublyLinkedList.tpp', LF will be replaced by CRLF the next time Git touches it

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git commit -m "stworzenie pliku DoublyLinkedList.tpp"
[main acdc71c] stworzenie pliku DoublyLinkedList.tpp
1 file changed, 121 insertions(+)
create mode 100644 DoublyLinkedList.tpp

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 992 bytes | 992.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git
5a1bf2e..acdc71c main -> main

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git add .
warning: in the working copy of 'ListFactory.h', LF will be replaced by CRLF the next time Git touches it

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git commit -m "stworzenie pliku ListFactory.h"
[main 6f944c6] stworzenie pliku ListFactory.h
1 file changed, 23 insertions(+)
create mode 100644 ListFactory.h

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
```

Rys. 2.3. Tworzenie commitów 3

Tworzenie commitów 4

```
C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git add .
warning: in the working copy of 'ListFactory.h', LF will be replaced by CRLF the next time Git touches it

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git commit -m "stworzenie pliku ListFactory.h"
[main 6f944c6] stworzenie pliku ListFactory.h
1 file changed, 23 insertions(+)
 create mode 100644 ListFactory.h








C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 593 bytes | 593.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git
   acdc71c..6f944c6  main -> main

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git add .
warning: in the working copy of 'main.cpp', LF will be replaced by CRLF the next time Git touches it








C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git commit -m "stworzenie pliku main.cpp"
[main af4ba5a] stworzenie pliku main.cpp
1 file changed, 67 insertions(+)
 create mode 100644 main.cpp

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 916 bytes | 916.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git
   6f944c6..af4ba5a  main -> main
```

Rys. 2.4. Tworzenie commitów 4

	ilist.h	stworzenie pliku ilist.h	27 minutes ago	C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git add . warning: in the working copy of 'main.cpp', LF will be replaced by CRLF the next time Git touches it
	ilistIterator.h	stworzenie pliku ilistIterator.h	19 minutes ago	C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git commit -m "stworzenie zmiennej error"
	ListFactory.h	stworzenie pliku ListFactory.h	14 minutes ago	[main 1e9d4ee] stworzenie zmiennej error 1 file changed, 1 insertion(+), 2 deletions(-)
	ListIterator.h	stworzenie pliku ListIterator.h	18 minutes ago	C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin main
	Node.h	stworzenie pliku Node.h	24 minutes ago	Enumerating objects: 5, done. Counting objects: 100% (5/5), done. Delta compression using up to 12 threads Compressing objects: 100% (3/3), done. Writing objects: 100% (3/3), 322 bytes 322.00 KiB/s, done. Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0) remote: Resolving deltas: 100% (2/2), completed with 2 local objects. To https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git af4ba5a..1e9d4ee main -> main
	main.cpp	stworzenie zmiennej error	now	
	redme.md	pierwszy commit	1 hour ago	C:\Users\Sebastian\Desktop\ListaDwukierunkowa>

Rys. 2.5. Tworzenie commitów do cofnienia

	ilist.h	stworzenie pliku ilist.h	30 minutes ago	C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git add . warning: in the working copy of 'main.cpp', LF will be replaced by CRLF the next time Git touches it
	ilistIterator.h	stworzenie pliku ilistIterator.h	22 minutes ago	C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git commit -m "wypisanie bledu"
	ListFactory.h	stworzenie pliku ListFactory.h	17 minutes ago	[main ca5c008] wypisanie bledu 1 file changed, 1 insertion(+)
	ListIterator.h	stworzenie pliku ListIterator.h	21 minutes ago	C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin main
	Node.h	stworzenie pliku Node.h	28 minutes ago	Enumerating objects: 5, done. Counting objects: 100% (5/5), done. Delta compression using up to 12 threads Compressing objects: 100% (3/3), done. Writing objects: 100% (3/3), 320 bytes 320.00 KiB/s, done. Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0) remote: Resolving deltas: 100% (2/2), completed with 2 local objects. To https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git 1e9d4ee..ca5c008 main -> main
	main.cpp	wypisanie bledu	now	
	redme.md	pierwszy commit	1 hour ago	C:\Users\Sebastian\Desktop\ListaDwukierunkowa>

Rys. 2.6. Tworzenie commitów do cofnienia

Cofnięcie o dwa commity

Node.h	stworzenie pliku Node.h	29 minutes ago	C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git reset --hard HEAD~2 HEAD is now at af4ba5a stworzenie pliku main.cpp
main.cpp	stworzenie pliku main.cpp	18 minutes ago	C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin main --force Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0) To https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git + ca5c008...af4ba5a main -> main (forced update)
redme.md	pierwszy commit	1 hour ago	

Rys. 2.7. Cofnięcie o dwa commity

```

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.01 KiB | 1.01 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git
dfaec14..c0b43bf main -> main

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git reset --hard HEAD~1
HEAD is now at dfaec14 stworzenie pliku Node.h

C:\Users\Sebastian\Desktop\ListaDwukierunkowa>git push origin HEAD --force
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/xSebix10/Projekt-ListaDwukierunkowa.git
+ c0b43bf...dfaec14 HEAD -> main (forced update)

```

Rys. 2.8. Usunięcie podwojonego commita

xSebix10 stworzenie pliku main.cpp		af4ba5a · 4 days ago	9 Commits
DoublyLinkedList.h	stworzenie pliku DoublyLinkedList.h"	4 days ago	
DoublyLinkedList.tpp	stworzenie pliku DoublyLinkedList.tpp"	4 days ago	
IList.h	stworzenie pliku IList.h	4 days ago	
IListIterator.h	stworzenie pliku IListIterator.h	4 days ago	
ListFactory.h	stworzenie pliku ListFactory.h	4 days ago	
ListIterator.h	stworzenie pliku ListIterator.h"	4 days ago	
Node.h	stworzenie pliku Node.h	4 days ago	
main.cpp	stworzenie pliku main.cpp	4 days ago	
redme.md	pierwszy commit	4 days ago	

Rys. 2.9. Widok repozytorium projektu w GitHub

3. Projektowanie

Projekt został wykonany w języku C++17 przy użyciu środowiska Visual Studio 2022. Do generowania dokumentacji użyto narzędzia Doxygen.

W projekcie zastosowano dwa wzorce projektowe:

- Factory – odpowiada za tworzenie obiektów klasy listy, dzięki czemu kod w `main.cpp` nie zależy od konkretnej implementacji.
- Iterator – umożliwia iterowanie po elementach listy w przód i w tył.

Użyte narzędzia i biblioteki

- Język programowania: C++
- Kompilator: MSVC (Microsoft Visual C++ 17)
- Środowisko: Visual Studio 2022
- Narzędzia: Git, GitHub Desktop, Doxygen
- Dodatkowe biblioteki: brak (projekt oparty tylko na standardowej bibliotece STL)

Każda klasa posiada jasno określona odpowiedzialność:

- Node – przechowuje dane i wskaźniki,
- DoublyLinkedList – obsługuje logikę listy,
- ListIterator – iteracja po elementach,
- ListFactory – tworzy listę.

Opis narzędzia Git i GitHub

Git to system kontroli wersji pozwalający śledzić zmiany w kodzie i przywracać wcześniejsze wersje projektu. W ramach pracy wykonano podstawowe operacje: tworzenie repozytorium, commitowanie zmian, cofanie commitów, synchronizację między dwiema lokalizacjami oraz pobieranie projektu z GitHuba.

Doxygen

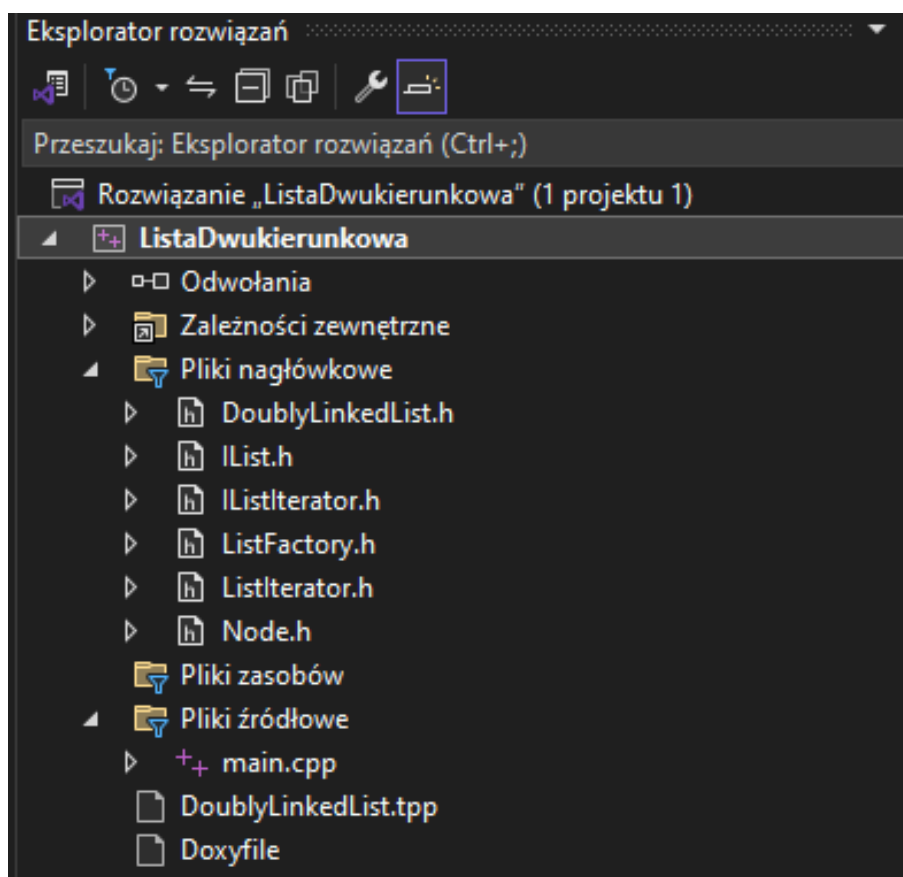
Doxygen umożliwia automatyczne tworzenie dokumentacji technicznej programu na podstawie komentarzy w kodzie. W projekcie użyto go do wygenerowania pliku PDF z opisem klas i metod, co pozwoliło na czytelne przedstawienie struktury programu.

4. Implementacja

W implementacji utworzono kilka klas w osobnych plikach:

Tab. 4.1. Tabela przedstawia pliki projektu wraz z ich zawartoscia i przeznaczeniem.

Plik	Zawartosc
Node.h	Definicja struktury wezla listy
IList.h	Interfejs klasy listy
DoublyLinkedList.h / .tpp	Implementacja metod listy dwukierunkowej
ListIterator.h	Iterator po elementach listy
ListFactory.h	Fabryka tworzaca obiekty listy
main.cpp	Testy funkcjonalnosci programu



Rys. 4.1. Struktura projektu

Fragment kodu main.cpp testującego wszystkie funkcje

```
#include <iostream>
#include "ListFactory.h"

int main() {
    // Tworzymy listę typu int przez wzorzec Factory
    auto *list = ListFactory::create<int>();
    int error=0;
    // Dodajemy elementy
    list->push_back(10);
    list->push_front(5);
    list->push_back(20);
    list->insert_at(1, 7); // lista: 5, 7, 10, 20

    std::cout << "Lista (od początku): ";
    list->print_forward();

    std::cout << "Lista (od końca): ";
    list->print_backward();

    // Iterator
    auto it = list->begin_iterator();
    try {
        std::cout << "\\nIterator - startowy element: " << it.value() << std::endl;

        if (it.has_next()) {
            it.next();
            std::cout << "Iterator - następny element: " << it.value() << std::endl;
        }
        if (it.has_next()) {
            it.next();
            std::cout << "Iterator - następny element: " << it.value() << std::endl;
        }
        if (it.has_prev()) {
            it.prev();
            std::cout << "Iterator - poprzedni element: " << it.value() << std::endl;
        }
    } catch (const std::exception &e) {
        std::cerr << "Błąd iteratora: " << e.what() << std::endl;
    }

    // Usunięcie elementów
    try {
        list->pop_front(); // usunąć 5
        list->pop_back(); // usunąć 20
    } catch (const std::exception &e) {
        std::cerr << "Błąd przy usuwaniu: " << e.what() << std::endl;
    }

    std::cout << "\\nPo usunięciach (od początku): ";
    list->print_forward();

    try {
        if (list->size() > 1) list->remove_at(1); // usunąć 10 (zostaje 7)
    } catch (const std::exception &e) {
        std::cerr << "Błąd remove_at: " << e.what() << std::endl;
    }

    std::cout << "Po remove_at(1): ";
    list->print_forward();

    // Test czyszczenia
    list->clear();
    std::cout << "\\nPo clear() - rozmiar listy: " << list->size() << std::endl;

    delete list; // zwolnienie pamięci
    return 0;
}
```

Rys. 4.2. Main

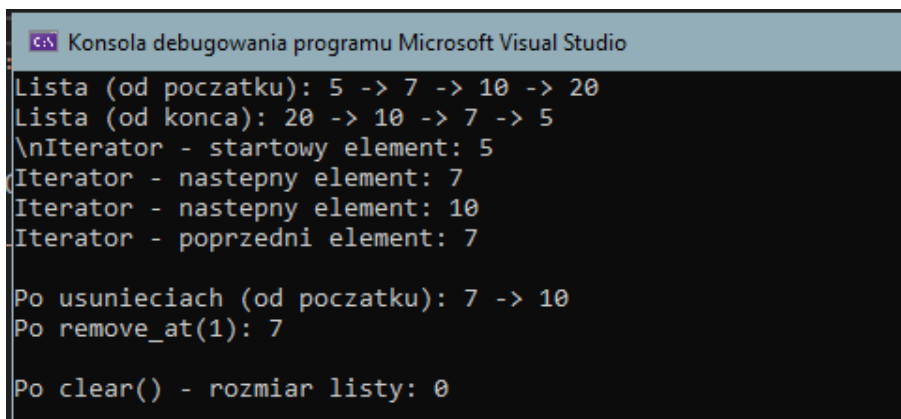
4.1. Ciekawe fragmenty kodu

Przykład metody dodawania elementu na koniec listy:

```
1 void push_back(int value) {  
2     Node* newNode = new Node(value);  
3     if (!head) {  
4         head = tail = newNode;  
5     } else {  
6         tail->next = newNode;  
7         newNode->prev = tail;  
8         tail = newNode;  
9     }  
10 }
```

Listing 1. Przykład metody dodawania elementu na koniec listy

Wynik działania programu



```
Microsoft Visual Studio  
Lista (od poczatku): 5 -> 7 -> 10 -> 20  
Lista (od konca): 20 -> 10 -> 7 -> 5  
\nIterator - startowy element: 5  
Iterator - nastepny element: 7  
Iterator - nastepny element: 10  
Iterator - poprzedni element: 7  
  
Po usunieciach (od poczatku): 7 -> 10  
Po remove_at(1): 7  
  
Po clear() - rozmiar listy: 0
```

Rys. 4.3. Wynik działania programu

Dokumentacja Doxygen

	i
1 redme	1
2 Hierarchical Index	5
2.1 Class Hierarchy	5
3 Class Index	7
3.1 Class List	7
4 File Index	9
4.1 File List	9
5 Class Documentation	11
5.1 DoublyLinkedList< T > Class Template Reference	11
5.1.1 Constructor & Destructor Documentation	12
5.1.1.1 DoublyLinkedList()	12
5.1.1.2 ~DoublyLinkedList()	12
5.1.2 Member Function Documentation	12
5.1.2.1 begin_iterator()	12
5.1.2.2 clear()	12
5.1.2.3 end_iterator()	12
5.1.2.4 insert_at()	12
5.1.2.5 pop_back()	13
5.1.2.6 pop_front()	13
5.1.2.7 print_backward()	13
5.1.2.8 print_forward()	13
5.1.2.9 push_back()	13
5.1.2.10 push_front()	14
5.1.2.11 remove_at()	14
5.1.2.12 size()	14
5.2 IList< T > Class Template Reference	14
5.2.1 Constructor & Destructor Documentation	15
5.2.1.1 ~IList()	15
5.2.2 Member Function Documentation	15
5.2.2.1 clear()	15
5.2.2.2 insert_at()	15
5.2.2.3 pop_back()	15
5.2.2.4 pop_front()	15
5.2.2.5 print_backward()	16
5.2.2.6 print_forward()	16
5.2.2.7 push_back()	16

Rys. 4.4. Fragment dokumentacji doxygen

5. Wnioski

Projekt pozwolił na praktyczne zrozumienie działania listy dwukierunkowej i zastosowanie wzorców projektowych Factory i Iterator w C++.

Dodatkowo umożliwił naukę pracy z:

- systemem kontroli wersji Git i GitHub,
- generatorem dokumentacji Doxygen,
- oraz środowiskiem Visual Studio 2022.

Uzyskane wyniki potwierdziły poprawne działanie wszystkich metod listy oraz skuteczne wykorzystanie wzorców projektowych.

Spis rysunków

2.1. Tworzenie commitów 1	6
2.2. Tworzenie commitów 2	7
2.3. Tworzenie commitów 3	8
2.4. Tworzenie commitów 4	9
2.5. Tworzenie commitów do cofnięcia	9
2.6. Tworzenie commitów do cofnięcia	9
2.7. Cofnięcie o dwa commity	10
2.8. Usunięcie podwojonego commita	10
2.9. Widok repozytorium projektu w GitHub	10
4.1. Struktura projektu	12
4.2. Main	13
4.3. Wynik działania programu	14
4.4. Fragment dokumentacji doxygen	15

Spis tabel

2.1. Porównanie podstawowych struktur danych	5
4.1. Struktura plików projektu	12

Spis listingów

1. Przykład metody dodawania elementu na koniec listy 14