Lista

Wygenerowano za pomocą Doxygen 1.12.0

Rozdział 1

Cel Projektu

Celem projektu jest stworzenie implementacji listy dwukierunkowej działającej na stercie w języku C++. Program ma być zrealizowany w formie klasy, która będzie oferowała określone funkcje do manipulacji listą. Działanie klasy zostanie przetestowane w funkcji main.

Klasa listy dwukierunkowej powinna udostępniać 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 pod wskazanym indeksem
- Wyświetl całą listę
- Wyświetl listę w odwrotnej kolejności
- Wyświetl następny element
- Wyświetl poprzedni element
- Czyść całą listę
- Testowanie klasy, każda z metod klasy powinna być przetestowana w funkcji main, aby zweryfikować poprawność ich działania.

Rozdział 2

Indeks klas

2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja klasy DoublyLinkedList

Implementacja podwójnie powiązanego listy.

Metody publiczne

- DoublyLinkedList ()

  Konstruktor klasy DoublyLinkedList.
- ~DoublyLinkedList ()

  Destruktor klasy DoublyLinkedList.
- void AddAtBeg (int value)

  Dodaje element na początek listy.
- void AddAtEnd (int value)

  Dodaje element na koniec listy.
- void AddAtIndex (int index, int value)

  Dodaje element na określony indeks listy.
- void RemoveFromBeg ()

  Usuwa element z początku listy.
- void RemoveFromEnd ()

  Usuwa element z końca listy.
- void RemoveAtIndex (int index)

  Usuwa element z określonego indeksu listy.
- void display () const

  Wyświetla wszystkie elementy listy od początku do końca.
- void DisplayReverse () const

  Wyświetla wszystkie elementy listy w odwrotnej kolejności.
- void Clear ()

  Czyści całą listę.

3.1.1 Opis szczegółowy

Implementacja podwójnie powiązanego listy.

Ta klasa implementuje listę podwójnie powiązaną z operacjami dodawania, usuwania oraz wyświetlania elementów w różnych porządkach.

3.1.2 Dokumentacja konstruktora i destruktora

3.1.2.1 DoublyLinkedList()

```
DoublyLinkedList::DoublyLinkedList ()  [inline]
```

Konstruktor klasy DoublyLinkedList.

Inicjalizuje pustą listę.

3.1.2.2 ~DoublyLinkedList()

```
DoublyLinkedList::~DoublyLinkedList ()  [inline]
```

Destruktor klasy DoublyLinkedList.

Zwalnia pamięć zajmowaną przez listę.

3.1.3 Dokumentacja funkcji składowych

3.1.3.1 AddAtBeg()

```
void DoublyLinkedList::AddAtBeg (
            int value)
```

Dodaje element na początek listy.

Tworzy nowy węzeł i dodaje go na początek listy.

Parametry

value	Wartość, którą przechowa nowy węzeł.

3.1.3.2 AddAtEnd()

```
void DoublyLinkedList::AddAtEnd (
            int value)
```

Dodaje element na koniec listy.

Tworzy nowy węzeł i dodaje go na koniec listy.

Parametry

value	Wartość, którą przechowa nowy węzeł.

3.1.3.3 AddAtIndex()

```
void DoublyLinkedList::AddAtIndex (
            int index,
            int value)
```

Dodaje element na określony indeks listy.

Tworzy nowy węzeł i wstawia go w odpowiednie miejsce w liście.

Parametry

index	Indeks, na którym ma zostać dodany element.
value	Wartość, którą przechowa nowy węzeł.

3.1.3.4 Clear()

```
void DoublyLinkedList::Clear ()
```

Czyści całą listę.

Funkcja usuwa wszystkie węzły z listy.

3.1.3.5 display()

```
void DoublyLinkedList::display () const
```

Wyświetla wszystkie elementy listy od początku do końca.

Funkcja wypisuje dane każdego węzła w liście, zaczynając od pierwszego węzła.

3.1.3.6 DisplayReverse()

```
void DoublyLinkedList::DisplayReverse () const
```

Wyświetla wszystkie elementy listy w odwrotnej kolejności.

Funkcja wypisuje dane każdego węzła w liście, zaczynając od ostatniego węzła.

3.1.3.7 RemoveAtIndex()

```
void DoublyLinkedList::RemoveAtIndex (
            int index)
```

Usuwa element z określonego indeksu listy.

Usuwa element znajdujący się na wskazanym indeksie.

Parametry

index	Indeks elementu, który ma zostać usunięty.

3.1.3.8 RemoveFromBeg()

```
void DoublyLinkedList::RemoveFromBeg ()
```

Usuwa element z początku listy.

Usuwa pierwszy element z listy.

3.1.3.9 RemoveFromEnd()

```
void DoublyLinkedList::RemoveFromEnd ()
```

Usuwa element z końca listy.

Usuwa ostatni element z listy.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- C:/gry/Lista/Lista/Lista.cpp

Skorowidz