yazı tipi, logo, simge, sembol, ticari marka içeren bir resim

Açıklama otomatik olarak oluşturuldu

**GAZİ ÜNİVERSİTESİ TEKNOLOJİ FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ**

**BMT-210 VERİ YAPILARI VE ALGORİTMALAR LABORATUVARI 1.Şube**

**Doç.Dr. Adem Tekerek**

**5. HAFTA LAB ÖDEVİ**

21181617001- Mehmet DEMİR

İçindekiler

[1 Kod İçeriği 2](#_Toc162299513)

[2 Konsol Çıktısı 3](#_Toc162299514)

# 1 Kod İçeriği

#include <iostream>

#include <stdlib.h>

#include <stdio.h>

using namespace std;

// Tek bağlantılı liste düğümü

struct Node {

int data;

Node\* next;

};

class LinkedList {

private:

Node\* head;

public:

// Constructor

LinkedList() : head(NULL){}

// Liste başına ekleme yapan method

void addFirst(int value) {

Node\* newNode = (Node\*)malloc(sizeof(Node));

newNode->data = value;

newNode->next = head;

head = newNode;

}

// Liste sonuna ekleme yapan method

void addLast(int value) {

Node\* newNode = (Node\*)malloc(sizeof(Node));

newNode->data = value;

newNode->next = NULL;

if (head == NULL) {

head = newNode;

return;

}

Node\* temp = head;

while (temp->next != NULL) {

temp = temp->next;

}

temp->next = newNode;

}

// Belirtilen konumdan sonra ekleme yapan method

void addAfter(int position ,int value){

Node\* newNode = (Node\*)malloc(sizeof(Node));

newNode -> data =value;

newNode -> next =NULL;

Node\*temp = head;

for(int i =1; i < position && temp != NULL; i++){

temp = temp -> next;

}

if(temp ==NULL){

printf("Belirtilen konum bulunamadi !");

return;

}

newNode -> next =newNode;

}

// İlk elemanı silen method

void deleteFirst() {

if (head == NULL) {

printf("Liste bos, silinecek eleman yok!");

return;

}

Node\* temp = head;

head = head->next;

free(temp);

}

// Son elemanı silen method

deleteLast(){

if(head == NULL){

printf("Liste bos silinecek elemen yok ");

}

if(head -> next ==NULL){

free(head);

head = NULL;

}

Node\* temp = head;

while(temp ->next -> next != NULL){

temp = temp -> next;

}

free(temp -> next);

temp -> next = NULL;

}

// Listeden silinecek elemanın değeri girilerek silen method

void deleteValue(int value){

if ( head == NULL ){

printf("Liste bos, silinecek eleman yok");

}

if( head -> data ==value){

deleteFirst();

}

Node\*temp = head;

while(temp -> next != NULL && temp -> data != value){

temp = temp -> next;

}

if( temp -> next == NULL){

printf("Silinecek eleman bulunamadi");

}

Node \* delNode = temp -> next;

temp -> next = temp -> next -> next;

free(delNode);

}

// Bağlı listedeki elemanları ekrana yazdıran method

void yazdir() {

Node\* temp = head;

while (temp != NULL) {

printf("%d ",temp->data);//cout << temp->data << " ";

temp = temp->next;

}

cout << endl;

}

// Bağlı listede bulunan eleman değeri girilerek arama yapmak

void search(int value) {

Node\* temp = head;

while (temp != NULL) {

if (temp->data == value) {

cout << "Aranan eleman bulundu!" << endl;

return;

}

temp = temp->next;

}

cout << "Listede aranan eleman bulunamadi!" << endl;

}

// Destructor

~LinkedList() {

while (head != NULL) {

Node\* temp = head;

head = head->next;

free(temp);

}

}

};

int main() {

LinkedList list;

int secim;

do {

printf("\n1. Liste basina eleman ekle\n ");

printf("2. Liste sonuna eleman ekle\n ");

printf("3. Belirtilen konumdan sonra eleman ekle\n");

printf("4. Ilk elemani sil\n");

printf("5. Son elemani sil\n");

printf("6. Deger girerek eleman sil\n");

printf("7. Listeyi goster\n");

printf("8. Eleman ara\n");

printf("9. Cikis\n");

printf("Seciminizi yapin:");

cin >> secim;

switch (secim) {

case 1: {

int value;

printf("Eklenecek degeri girin:");

cin >> value;

list.addFirst(value);

break;

}

case 2: {

int value;

printf("Eklenecek degeri girin:");

cin >> value;

list.addLast(value);

break;

}

case 3: {

int position, value;

printf("Konumu girin:");

cin >> position;

printf("Eklenecek degeri girin: ");

cin >> value;

list.addAfter(position, value);

break;

}

case 4:

list.deleteFirst();

break;

case 5:

list.deleteLast();

break;

case 6: {

int value;

printf("Silinecek degeri girin: ");

cin >> value;

list.deleteValue(value);

break;

}

case 7:

printf("Liste: ");

list.yazdir();

break;

case 8: {

int value;

printf("Aranacak degeri girin: ");

cin >> value;

list.search(value);

break;

}

case 9:

printf("Program sonlandiriliyor...\n");

break;

default:

printf("Gecersiz secim! Lutfen tekrar deneyin.\n");

break;

}

} while (secim != 9);

return 0;

}

# 2 Konsol Çıktısı

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu