**SPRAWOZDANIE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Przedmiot** | Wprowadzenie do Informatyki | **Zadanie** | 4.1 |
| **Autor** | Andrii Godlevskyi | **Grupa** | WCY23KY1S1 |
| **Temat** | Struktury dynamiczne​ | | |

1.Treść zadania

Należy napisać program realizujący na liście jednokierunkowej elementów typu „int” lub „char” zaimplementowanej wskaźnikowo następujące operacje:​

Dodaj obiekt,​

Usuń obiekt,​

Znajdź obiekt,​

Wylistuj zapisane obiekty (dane i adres następnika).

1. Metoda realizacji

Tworzymy listę i funkcje operujące na nią i w zależności od wyboru użytkownika wywołujemy potrzebną funkcje i wyprowadzamy wynik na ekran.

1. Założenia / ograniczenia dotyczące danych:

1. Dane wejściowe

Dane obiektów – wczytywane z klawiatury

1. Dane wyjściowe

Dane zapisanych obiektów - wyświetlone na ekranie *(z opisami).*

1. Realizacja
2. Algorytm

Изображение выглядит как текст, рукописный текст, бумага, Бумажное изделие

Автоматически созданное описание

1. Kod zródłowy

#include <stdio.h>

#include <stdlib.h>

//deklaracja Struktury Node

typedef struct Node {

int data;

struct Node\* next;

} Node;

Node\* head = NULL;

//funkcja dodania elementu

void dodajel(int data) {

Node\* new\_node = (Node\*)malloc(sizeof(Node));

new\_node->data = data;

new\_node->next = head;

head = new\_node;

}

//funkcja usuwania elementu

void usunel(int data) {

Node \*current = head;

Node \*previous = NULL;

while (current != NULL) {

if (current->data == data) {

if (previous != NULL) {

previous->next = current->next;

} else {

head = current->next;

}

free(current);

return;

}

previous = current;

current = current->next;

}

printf("Taki element nie istnieje.\n");

}

//funkcja znajdowania elementu

Node\* znajdzel(int data) {

Node\* current = head;

while (current != NULL) {

if (current->data == data) {

return current;

}

current = current->next;

}

return NULL;

}

//funkcja wypisania listy elementow

void listael() {

Node\* current = head;

printf("List of elements:\n");

while (current != NULL) {

printf("Data: %d, Address of next element: %p\n", current->data, (void\*)current->next);

current = current->next;

}

}

int main() {

//deklaracja zmiennych

int choice, data;

//menu wyboru

do {

printf("\n1.Dodaj element \n2. Usun element\n3. Znajdz element\n4. Lista elementow\n5. Wyjscie\n");

printf("Wpisz swoj wybor(numer): ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Wpisz liczbe do dodania: ");

scanf("%d", &data);

dodajel(data);

break;

case 2:

printf("Wpisze liczbe do usuniecia: ");

scanf("%d", &data);

usunel(data);

break;

case 3:

printf("Wpisz liczbe do znajdowania: ");

scanf("%d", &data);

Node\* found\_element = znajdzel(data);

if (found\_element != NULL) {

printf("Element z liczba %d znajduje sie pod adresem: %p\n", data, (void\*)found\_element);

} else {

printf("Brak elementu z liczba %d \n", data);

}

break;

case 4:

listael();

break;

case 5:

printf("Wyjscie z programu\n");

break;

default:

printf("Nie ma takiego wariantu,sprobuj jeszcze raz\n");

}

} while (choice != 5);

return 0;

}

3. Dane wejściowe /wyjściowe

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Złożoność obliczeniowa algorytmu

O(n)