**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

**ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**

Лабораторна робота №8

**з дисципліни**

“ **Програмування 1. Структурний підхід“**

Підготувала

студентка групи ФІ-12

Бекешева Анастасія

(№ залікової книжки - 2)

**Київ – 2021**

**ЗМІСТ**

[Лабораторна робота №8 1](#_Toc90923725)

[Умова задачі 3](#_Toc90923726)

[Використані змінні 4](#_Toc90923727)

[Використані типи даних користувача 4](#_Toc90923728)

[Опис проектних рішень 5](#_Toc90923729)

[Програмний код 6](#_Toc90923730)

[LinkedList.hpp 6](#_Toc90923731)

[LinkedList.cpp 7](#_Toc90923732)

[Main.cpp 9](#_Toc90923733)

Умова задачі

1. Створити функцію, яка повертає вказівник на другий елемент списку.

2. Створити функцію, яка повертає вказівник на останній елемент

списку.

Використані змінні

|  |  |  |
| --- | --- | --- |
| Тип змінної | Ім’я | Призначення |
| Link\* | head | Вказівник на перший елемент списку |
| Link\* | current | Вказівник на поточний елемент |
| string | str | Елемент списку |
| bool | stop | Значення для зупинки програми |
| string | buff | Тимчасовий рядок для зберігання даних |
| int | choice | Число для вибору функції |
| Link\* | ptr | Вказівник на елемент списку |
| Link\* | ptr1 | Вказівник на елемент списку |
| Link\* | next | Вказівник на елемент списку |

Використані типи даних користувача

|  |  |
| --- | --- |
| Тип даних користувача | Призначення |
| struct Link{  string str;  Link \*next;  }; | Елемент списку містить вказівник на рядок, а також вказівник на наступний елемент. |

Опис проектних рішень

1. Введення списку(функція – get\_list) – користувач вводить данні будь-якого типу через enter. Дається вказівник на початок(current) та новий елемент(temp). Введений елемент потрапляє у тимчасовий рядок buff. На місце нового елементу потрапляє рядок buff. Вказівник після нового вказує на NULL. Поки вказівник після поточного не вказує на NULL, створюються нові елементи. Це виконується поки користувач не введе стоп слово – “stop”.
2. Виведення списку(функція – print\_list) – через пробіл виводяться данні введені користувачем за допомогою функції get\_list. Дається вказівник на нульовий елемент та наступний. Виводиться значення на яке вказує поточний вказівник. Це виконується поки наступний вказівник не дорівнюватиме NULL.
3. Виведення вказівника на другий елемент(функція return\_second) – дається вказівник на наступний після нульового елемента(ptr) та ще на наступний(ptr1). Виводиться вказівник та сам елемент через тире(“-”).
4. Виведення вказівника на останній елемент(функція return\_last) – дається вказівник на нульовий елемент(ptr) та наступний(ptr1). Цикл триває по є ptr1: при кожному кроці дається вказівник на ще наступний елемент та якщо при якомусь кроці наступний після ptr1 вказівник має значення NULL виводиться вказівник ptr1, цикл завершується.

Програмний код

## LinkedList.hpp

#include <stdio.h>

#include <string>

#include <vector>

using namespace std;

struct Link{

string str;

Link \*next;

};

void get\_list(Link \*head);

void print\_list(Link \*head);

void return\_second(Link \*head);

void return\_last(Link \*head);

## 

## LinkedList.cpp

#include "LinkedList.hpp"

#include <iostream>

#include <string>

void get\_list(Link \*head){

bool stop = false;

string buff;

cout << "Enter new elements" << endl << "If you want to stop enter \"-s\"" << endl;

while (!stop){

cin >> buff;

if (buff == "-s"){

stop = true;

}

else{

Link \*current = head;

Link \*temp = new Link;

temp->str = buff;

temp->next = NULL;

if(head == NULL){

head = temp;

return;

}

while(current->next != NULL){

current = current->next;

}

current->next = temp;

}

}

}

void print\_list(Link \*head){

while(head != NULL){

cout << head->str << " ";

head = head->next;

}

cout << endl;

}

void return\_second(Link \*head){

Link \*ptr = head->next;

Link \*ptr1 = ptr->next;

cout << ptr1 << " - " << ptr1->str <<endl;

}

void return\_last(Link \*head){

Link \*ptr = head;

Link \*ptr1 = ptr->next;

while(ptr1){

if(ptr1->next == NULL){

cout << ptr1 << " - " << ptr1->str << endl;

break;

}

ptr = ptr1;

ptr1 = ptr1->next;

}

}

## Main.cpp

#include <iostream>

#include <string>

#include "LinkedList.hpp"

int main(){

bool stop = false;

int choice;

Link \*head = new Link;

head->next = NULL;

cout << "1. get list from input" << endl

<< "2. print list" << endl

<< "3. print second pointer" << endl

<< "4. print last pointer" << endl

<< "5. stop" << endl;

while (!stop){

cout << "Your choice: " << endl;

cin >> choice;

switch(choice){

case 1: get\_list(head);

break;

case 2: print\_list(head);

break;

case 3: return\_second(head);

break;

case 4: return\_last(head);

break;

case 5: stop = true;

break;

default:

cout << "The input is invalid";

stop = true;

}

}

}

A screenshot of a computer

Description automatically generated