

Тема: Динамічні списки.

Мета: Навчитись працювати з динамічними списками

### **Індивідуальне завдання**

#### **Завдання на оцінку “задовільно”:**

На базі попередньо розробленого функціоналу по роботі з прикладною областю сформуванню односпрямований список елементів розробленої структури.

Реалізувати наступні функції роботи зі списком:

- читання даних з файлу, використовуючи функцію `fscanf`;
- запис даних у файл, використовуючи функцію `fprintf`;
- вивід вмісту списку на екран;
- пошук у списку об'єкта за заданим критерієм;
- додавання об'єкта у кінець списку;
- видалення об'єкта зі списку.

#### **Завдання на оцінку “добре”:**

- виконати завдання на “задовільно”;
- реалізувати діалоговий режим спілкування з користувачем за допомогою меню;
- сортування вмісту списку за одним з критеріїв. При цьому дозволяється міняти місцями не вказівники на об'єкти, а вміст об'єктів (за допомогою третьої змінної);
- переробити метод додавання з можливістю вставлення додаткового елемента після будь-якого елемента списку;
- продемонструвати відсутність витоків пам'яті;
- розробити модульні тести, що демонструють коректність роботи реалізованих функцій
- проект має складатися мінімум з 6 файлів (`main.c`, `test.c`, `list.h`, `list.c`, `data.c`, `data.h`)

#### **Завдання на оцінку “відмінно”:**

- виконати завдання на “добре”, але замість односпрямованого списку треба використовувати двоспрямований;
- Реалізувати сортування вмісту списку за одним з критеріїв. При цьому обов'язково забезпечити, щоб обмін місцями об'єктів здійснювався шляхом обміну їх покажчиків.

## Хід роботи

```
node_t* read_from_file() {  
    FILE* f = fopen(filename, "r");  
  
    struct Employee employee;  
    node_t* head = NULL;  
    head = (node_t*)malloc(sizeof(node_t));  
    head->employer = create();  
    head->next = NULL;  
  
    while (fscanf(f, "%s\n%s\n%s\n%s\n%d\n%d\n", employee.company, employee.mail, employee.fullName,  
        employee.characteristic, &employee.workExperience, &employee.insurance) != EOF) {  
        print(employee);  
        push(head, employee);  
    }  
    pop(&head);  
    fclose(f);  
    return head;  
}
```

Рис.1 - читання даних з файлу

```
Your choice: 6

Company:example
Mail:rabotnik264gmail.com
Fullname:example
Characteristic:example
Work experience in years:27
insurance:Yes

Company:example
Mail:rabotnik881gmail.com
Fullname:example
Characteristic:example
Work experience in years:25
insurance:Yes

Company:example
Mail:rabotnik591gmail.com
Fullname:example
Characteristic:example
Work experience in years:17
insurance:No

Company:example
Mail:rabotnik427gmail.com
Fullname:example
Characteristic:example
Work experience in years:10
insurance:Yes

Company:example
Mail:rabotnik304gmail.com
Fullname:example
Characteristic:example
Work experience in years:24
insurance:Yes

Company:example
Mail:rabotnik392gmail.com
Fullname:example
Characteristic:example
Work experience in years:26
insurance:Yes

Successful.
```

Результат зчитування з файлу.(Дивіться вміст файлу нижче)

```
void write_to_file(node_t* head) {  
    node_t* current = head;  
    FILE* file;  
    file = fopen(filename, "w");  
    fprintf(file, "");  
    fclose(file);  
    while (current != NULL) {  
        fwrite(current->employer);  
        current = current->next;  
    }  
}
```

Рис.2 - запись данных у файл

Company:example  
Mail:rabotnik264gmail.com  
Fullname:example  
Characteristic:example  
Work experience in years:27  
insurance:Yes

Company:example  
Mail:rabotnik881gmail.com  
Fullname:example  
Characteristic:example  
Work experience in years:25  
insurance:Yes

Company:example  
Mail:rabotnik591gmail.com  
Fullname:example  
Characteristic:example  
Work experience in years:17  
insurance:No


Company:example  
Mail:rabotnik427gmail.com  
Fullname:example  
Characteristic:example  
Work experience in years:10  
insurance:Yes

Company:example  
Mail:rabotnik304gmail.com  
Fullname:example  
Characteristic:example  
Work experience in years:24  
insurance:Yes

Company:example  
Mail:rabotnik392gmail.com  
Fullname:example  
Characteristic:example  
Work experience in years:26  
insurance:Yes

Successful.

Зміст списку у консолі.

 text – Блокнот

Файл Правка Формат Вид Справка

---

```
example
rabotnik264gmail.com
example
example
27
1
example
rabotnik881gmail.com
example
example
25
1
example
rabotnik591gmail.com
example
example
17
0
example
rabotnik427gmail.com
example
example
10
1
example
rabotnik304gmail.com
example
example
24
1
example
rabotnik392gmail.com
example
example
26
1
```

Вміст файлу після запису у файл.

```

void print_list(node_t* head) {
    node_t* current = head;

    while (current != NULL) {
        print(current->employer);
        current = current->next;
    }
}

```

Рис.3 - вивід вмісту списку.

```

A list of employees.Select an action from the following.

1.Write list to screen.
2.Generate and add employee.
3.Delete employee by index.
4.Search for employees with insurance.
5.Sorted by employee experience.
6.Reading employees from a file.
7.Writing employees in file.
0.Exit.

Your choice: 1

Company:example
Mail:rabotnik141gmail.com
Fullname:example
Characteristic:example
Work experience in years:27
insurance:No

Successful.

```

Результат роботи функції виводу.

```

void search_by_insurance(node_t* head) {
    node_t* current = head;

    while (current->next != NULL) {
        if (current->employer.insurance == true)
            print(current->employer);
        current = current->next;
    }
}

```

Рис.4 - пошук по заданому критерію.

```

Your choice: 4

Company:example
Mail:rabotnik264gmail.com
Fullname:example
Characteristic:example
Work experience in years:27
insurance:Yes

Company:example
Mail:rabotnik881gmail.com
Fullname:example
Characteristic:example
Work experience in years:25
insurance:Yes

Company:example
Mail:rabotnik427gmail.com
Fullname:example
Characteristic:example
Work experience in years:10
insurance:Yes

Successful.

```

Результат роботи пошуку по заданому критерію страхування.



```
void push(node_t* head, Employee employer) {  
    node_t* current = head;  
  
    while (current->next != NULL) {  
        current = current->next;  
    }  
  
    current->next = (node_t*)malloc(sizeof(node_t));  
    current->next->employer = employer;  
    current->next->next = NULL;  
}
```

Рис.5 - додавання об'єкту у кінець списку

0.Exit.

Your choice: 2

Successful.

A list of employees.Select an action from the following.

- 1.Write list to screen.
- 2.Generate and add employee.
- 3.Delete employee by index.
- 4.Search for employees with insurance.
- 5.Sorted by employee experience.
- 6.Reading employees from a file.
- 7.Writing employees in file.
- 0.Exit.

Your choice: 1

Company:example  
Mail:rabotnik141gmail.com  
Fullname:example  
Characteristic:example  
Work experience in years:27  
insurance:No

Company:example  
Mail:rabotnik500gmail.com  
Fullname:example  
Characteristic:example  
Work experience in years:27  
insurance:No

Successful.

Результат створення та додавання в кінець списку нового робітника.

```
void remove_by_index(node_t** head, int n) {  
  
    int i = 0;  
  
    node_t* current = *head;  
    node_t* temp_node = NULL;  
  
    if (n == 0) {  
        pop(head);  
        return;  
    }  
  
    for (int i = 0; i < n - 1; i++) {  
        if (current->next == NULL) {  
            return;  
        }  
        current = current->next;  
    }  
  
    temp_node = current->next;  
    current->next = temp_node->next;  
    free(temp_node);  
  
    return;  
}
```

Рис. 6 - видалення об'єкту зі списку.

0.Exit.

Your choice: 3

Choose index(0-1):0

Successful.

A list of employees.Select an action from the following.

- 1.Write list to screen.
- 2.Generate and add employee.
- 3.Delete employee by index.
- 4.Search for employees with insurance.
- 5.Sorted by employee experience.
- 6.Reading employees from a file.
- 7.Writing employees in file.
- 0.Exit.

Your choice: 1

Company:example

Mail:rabotnik500gmail.com

Fullname:example

Characteristic:example

Work experience in years:27

insurance:No

Successful.

Результат видалення робітника зі списку за індексом.

```

void menu(node_t* head) {
    printf("\n\n      A list of employees.Select an action from the following.\n\n");
    printf("1.Write list to screen.\n");
    printf("2.Generate and add employee.\n");
    printf("3.Delete employee by index.\n");
    printf("4.Search for employees with insurance.\n");
    printf("5.Sorted by employee experience.\n");
    printf("6.Reading employees from a file.\n");
    printf("7.Writing employees in file.\n");
    printf("0.Exit.\n\n");
    printf("Your choice: ");

    int choice = -1;
    int size_list;

    while (choice < 0 || choice>7) {
        scanf("%d", &choice);
        if (choice < 0 || choice>7)
            printf("You entered an invalid value, please retry\n\nYour choice: ");
    }
    printf("\n\n");

```

```

switch (choice) {
case 0: return;
case 1: print_list(head);
        break;
case 2: push(head, create());
        break;
case 3:
        size_list = size(head);
        choice = -1;
        printf("Choose index(0-%d):", size_list);
        while (choice < 0 || choice > size_list) {
            scanf("%d", &choice);
            if (choice < 0 || choice > size_list)
                printf("You entered an index that is larger than the size of the list, please retry\n\nYour choice: ");
        }
        remove_by_index(&head, choice);
        break;
case 4: search_by_insurance(head);
        break;
case 5: head = sort_by_work_exp(head);
        break;
case 6: head=read_from_file();
        break;
case 7: write_to_file(head);
        break;
}
printf("Successful.");
menu(head);
}

```

Рис.7-8 Діалогове меню.

```

node_t* sort_by_work_exp(node_t* head) {
    node_t* tmp = NULL;

    while (head != NULL) {
        node_t* node = head;
        head = head->next;

        if (tmp == NULL || node->employer.workExperience < tmp->employer.workExperience)
        {
            node->next = tmp;
            tmp = node;
        }
        else
        {
            node_t* current = tmp;
            while (current->next != NULL && !(node->employer.workExperience < current->next->employer.workExperience))
            {
                current = current->next;
            }

            node->next = current->next;
            current->next = node;
        }
    }

    return tmp;
}

```

Рис.9 - сортування за критерієм.

```
Your choice: 1

Company:example
Mail:rabotnik141gmail.com
Fullname:example
Characteristic:example
Work experience in years:27
insurance:No

Company:example
Mail:rabotnik500gmail.com
Fullname:example
Characteristic:example
Work experience in years:27
insurance:No

Company:example
Mail:rabotnik778gmail.com
Fullname:example
Characteristic:example
Work experience in years:10
insurance:No

Company:example
Mail:rabotnik264gmail.com
Fullname:example
Characteristic:example
Work experience in years:27
insurance:Yes

Successful.
```

Список до сортування.

```
Your choice: 1

Company:example
Mail:rabotnik778gmail.com
Fullname:example
Characteristic:example
Work experience in years:10
Insurance:No

Company:example
Mail:rabotnik141gmail.com
Fullname:example
Characteristic:example
Work experience in years:27
Insurance:No

Company:example
Mail:rabotnik500gmail.com
Fullname:example
Characteristic:example
Work experience in years:27
Insurance:No

Company:example
Mail:rabotnik264gmail.com
Fullname:example
Characteristic:example
Work experience in years:27
Insurance:Yes
```

Список після сортування.

```
void push_start(node_t** head, Employee employer) {  
    node_t* new_node;  
    new_node = (node_t*)malloc(sizeof(node_t));  
  
    new_node->employer = employer;  
    new_node->next = *head;  
    *head = new_node;  
}
```

Рис.10 - додавання з можливістю встановлення додаткового елементу.



**Висновок:** в лабораторній роботі отримані навички з роботою динамічних списків, а також зчитування за допомогою `fscanf`, запис `fprintf`, вивід вмісту списку на екран, пошук за заданим критерієм та додавання у кінець списку, видалення об'єкта зі списку