

Лабораторна робота №19

Тема: Динамічні списки.

Мета: Навчитися використовувати динамічні списки.

Індивідуальне завдання

Завдання на оцінку “задовільно”:

На базі попередньо розробленого функціоналу по роботі з прикладною областю сформувати 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

1. Функція для читання даних з файлу.

```

node_t* read_from_file() { // Функция для чтения из файла
    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;
}

```

Рисунок 1.1 - Функция read_from_file.

```
Company:example
Mail:work349gmail.com
Fullname:example
Characteristic:example
Work experience in years:29
insurance:No

Company:example
Mail:work628gmail.com
Fullname:example
Characteristic:example
Work experience in years:15
insurance:Yes

Company:example
Mail:work753gmail.com
Fullname:example
Characteristic:example
Work experience in years:20
insurance:No

Company:example
Mail:work426gmail.com
Fullname:example
Characteristic:example
Work experience in years:10
insurance:No

Company:example
Mail:work194gmail.com
Fullname:example
Characteristic:example
Work experience in years:18
insurance:Yes

Company:example
Mail:work885gmail.com
Fullname:example
Characteristic:example
Work experience in years:22
insurance:Yes

Successful.
```

Рисунок 1.2 - Приклад роботи коду.

2. Функція для запису у файл.

```
void write_to_file(node_t* head) { // Функція для записування в файл

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

Рисунок 2.1 - Функція write_to_file.

```
Company:example
Mail:work349gmail.com
Fullname:example
Characteristic:example
Work experience in years:29
insurance:No

Company:example
Mail:work628gmail.com
Fullname:example
Characteristic:example
Work experience in years:15
insurance:Yes

Company:example
Mail:work753gmail.com
Fullname:example
Characteristic:example
Work experience in years:20
insurance:No

Company:example
Mail:work426gmail.com
Fullname:example
Characteristic:example
Work experience in years:10
insurance:No

Company:example
Mail:work194gmail.com
Fullname:example
Characteristic:example
Work experience in years:18
insurance:Yes

Company:example
Mail:work885gmail.com
Fullname:example
Characteristic:example
Work experience in years:22
insurance:Yes

Successful.
```

Рисунок 2.2 - Приклад роботи коду.

```
example
work349gmail.com
example
example
29
0
example
work628gmail.com
example
example
15
1
example
work753gmail.com
example
example
20
0
example
work426gmail.com
example
example
10
0
example
work194gmail.com
example
example
18
1
example
work885gmail.com
example
example
22
1
```

Рисунок 2.3 - Вміст файлу text.txt.

3. Функція для виведення змісту списку.

```

void print_list(node_t* head) { // Функція для вивода содержания списка

    node_t* current = head;

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

```

Рисунок 3.1 - Функція print_list.

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:work141gmail.com
 Fullname:example
 Characteristic:example
 Work experience in years:27
 insurance:No

Successful

Рисунок 3.2 - Приклад роботи коду.

4. Функція для пошуку по заданому критерію.

```

void search_by_insurance(node_t* head) { // Поиск по заданному критерию
    node_t* current = head;

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

```

Рисунок 4.1 - Функція search_by_insurance.

```
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: 4

Company:example
Mail:work628gmail.com
Fullname:example
Characteristic:example
Work experience in years:15
insurance:Yes

Company:example
Mail:work194gmail.com
Fullname:example
Characteristic:example
Work experience in years:18
insurance:Yes

Successful.
```

Рисунок 4.2 - Приклад роботи коду.

5. Функція для додавання об'єкта в кінець списку.


```
void push(node_t* head, Employee employer) { // Добавление объекта в конец списка
    Добавление объекта в конец списка

    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;
}
```

Рисунок 5.1 - Функция push.

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: 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:work349gmail.com
Fullname:example
Characteristic:example
Work experience in years:29
insurance:No

Company:example
Mail:work778gmail.com
Fullname:example
Characteristic:example
Work experience in years:10
insurance:No

Successful.

Рисунок 5.2 - Приклад роботи коду.

6. Функція для видалення об'єкта зі списку.

```
void remove_by_index(node_t** head, int n) { // Удаление объекта из списка

    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;
}
```

Рисунок 6.1 - Функція remove_by_index.

```
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: 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:work778gmail.com
Fullname:example
Characteristic:example
Work experience in years:10
insurance:No

Successful.
```

Рисунок 6.2 - Приклад роботи коду.

7. Функція для сортування за критерієм.

```

node_t* sort_by_work_exp(node_t* head) { // Функция для сортировки по критерию
    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;
}

```

Рисунок 7.1 - Функция sort_by_work_exp.

```
Your choice: 1

Company:example
Mail:work349gmail.com
Fullname:example
Characteristic:example
Work experience in years:29
insurance:No

Company:example
Mail:work628gmail.com
Fullname:example
Characteristic:example
Work experience in years:15
insurance:Yes

Company:example
Mail:work753gmail.com
Fullname:example
Characteristic:example
Work experience in years:20
insurance:No

Company:example
Mail:work426gmail.com
Fullname:example
Characteristic:example
Work experience in years:10
insurance:No

Successful.
```

Рисунок 7.2 - Приклад списку до сортування.

Your choice:

5

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:work426gmail.com
Fullname:example
Characteristic:example
Work experience in years:10
insurance:No

Company:example
Mail:work628gmail.com
Fullname:example
Characteristic:example
Work experience in years:15
insurance:Yes

Company:example
Mail:work753gmail.com
Fullname:example
Characteristic:example
Work experience in years:20
insurance:No

Company:example
Mail:work349gmail.com
Fullname:example
Characteristic:example
Work experience in years:29
insurance:No

Successful.

Рисунок 7.3 - Приклад списку після сортування.

8. Меню.

```
int main() {  
  
    node_t* head = NULL;  
    head = (node_t*)malloc(sizeof(node_t));  
    head->employer = create();  
    head->next = NULL;  
    menu(head);  
    head = free_list(head);  
  
    return 0;  
}  
  
void menu(node_t* head) {  
    printf("\n\nA 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");  
}
```

Рисунок 8.1 - Код.


```

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);
}

```

Рисунок 8.2 - Код.

Висновок: Навчився використовувати динамічні списки.