

## Лабораторна робота №15

Романенко Дмитро. Кит120-Д

Тема: Структуровані типи даних.

Мета: Виконати індивідуальне завдання для отримання знань з розробки програм з структурованими типами даних

### Індивідуальне завдання

Прикладна галузь “Список боржників”. Сформувати масив даних боржників.

#### На оцінку “задовільно”.

- розробити функцію, яка читає дані (масив елементів) з файлу;
- розробити функцію, яка записує дані (масив елементів) у файл;
- розробити функцію, яка виводить масив елементів на екран;

#### На оцінку “добре”.

1. Виконати завдання на оцінку “задовільно” згідно варіанту.
2. Розробити функцію, яка буде сортувати масив елементів за заданим критерієм (полем);
3. Реалізувати функцію №1 з категорії “Методи для роботи з колекцією”, на вхід якої потрапляє масив об’єктів. Слід звернути увагу, що усі необхідні дані повинні бути передані як аргументи функції. Наприклад, якщо треба знайти всі машини марки “Форд”, то функція потрібна мати аргумент “марка машини”, та у `main()` викликати цю функцію з потрібним значенням марки.
4. Розробити функцію, яка генерує елемент прикладної галузі згідно з індивідуальним завданням, при цьому:
  - чисельні поля генеруються за допомогою функції `rand()` у діапазоні  $[N..N*10]$ , де  $N$  – номер варіанту;
  - рядкові поля подані у вигляді конкатенації двох рядків:, напр., “Студент №” та унікального числа.
5. Передача об’єктів структури в функцію (та отримання результатів роботи функції, що є об’єкт типу структури) обов’язково повинні передаватися “за вказівником”.

#### На оцінку “відмінно”.

1. Виконати завдання на оцінку “задовільно” та “добре” згідно варіанту.
2. Виконати запис та читання масиву структур (з заздалегідь відомою кількістю елементів) у двох форматах: текстовому та бінарному.

### Хід роботи

```

#include "Debtor.h"

int main() {
    int size = 3;
    struct Debtor* debtors;
    debtors = (struct Debtor*)malloc(size * sizeof(struct Debtor));
    for (int i = 0; i < size; i++) {
        debtors[i] = createClient();
    }
    printOnScreen(debtors, size);
    sortByDebt(debtors, size);
    writeInFile(debtors, size);

    return 0;
}

```

Рис. 1 - Фрагмент коду головної функції для виконання індивідуального завдання.

```

struct FullName {
    char name[SIZE];
    char surname[SIZE];
};

struct Debtor {
    FullName fullName;
    long int phone;    // номер телефона будет 7значным
    int debt; //сумма задолжности
    bool insurance;    //наличие страховки
};

```

Рис. 2. - Реалізація структури для Боржника.

```

Debtor createClient() {
    Debtor debtor;
    char tmp[SIZE]="";

    printf("Enter customer name:");
    scanf("%s", tmp);
    strcpy(debtor.fullName.name, tmp);

    printf("Enter customer surname:");
    scanf("%s", tmp);
    strcpy(debtor.fullName.surname, tmp);

    printf("Enter the customer's phone number (7-digit):");
    scanf("%ld", &debtor.phone);

    printf("Enter the customer loan amount: ");
    scanf("%d", &debtor.debt);
    printf("\n\n");
    debtor.insurance = rand() % 2;

    return debtor;
}

```

Рис. 3. - Функція створення елемента.

```

void printOnScreen(struct Debtor* debtors, int size) {
    for (int i = 0; i < size; i++) {
        printf("Full name: %s %s\n", debtors[i].fullName.name, debtors[i].fullName.surname);
        printf("Tel.: %d\n", debtors[i].phone);
        printf("Debt amount: %d\n", debtors[i].debt);

        printf("%s\n\n", debtors[i].insurance ? "[The client has insurance]" : "");
    }
}

```

Рис.4 - Функція для виводу у консоль.

```

void writeInFile(struct Debtor* debtors, int size) {
    FILE* file;
    file = fopen(filename, "w");

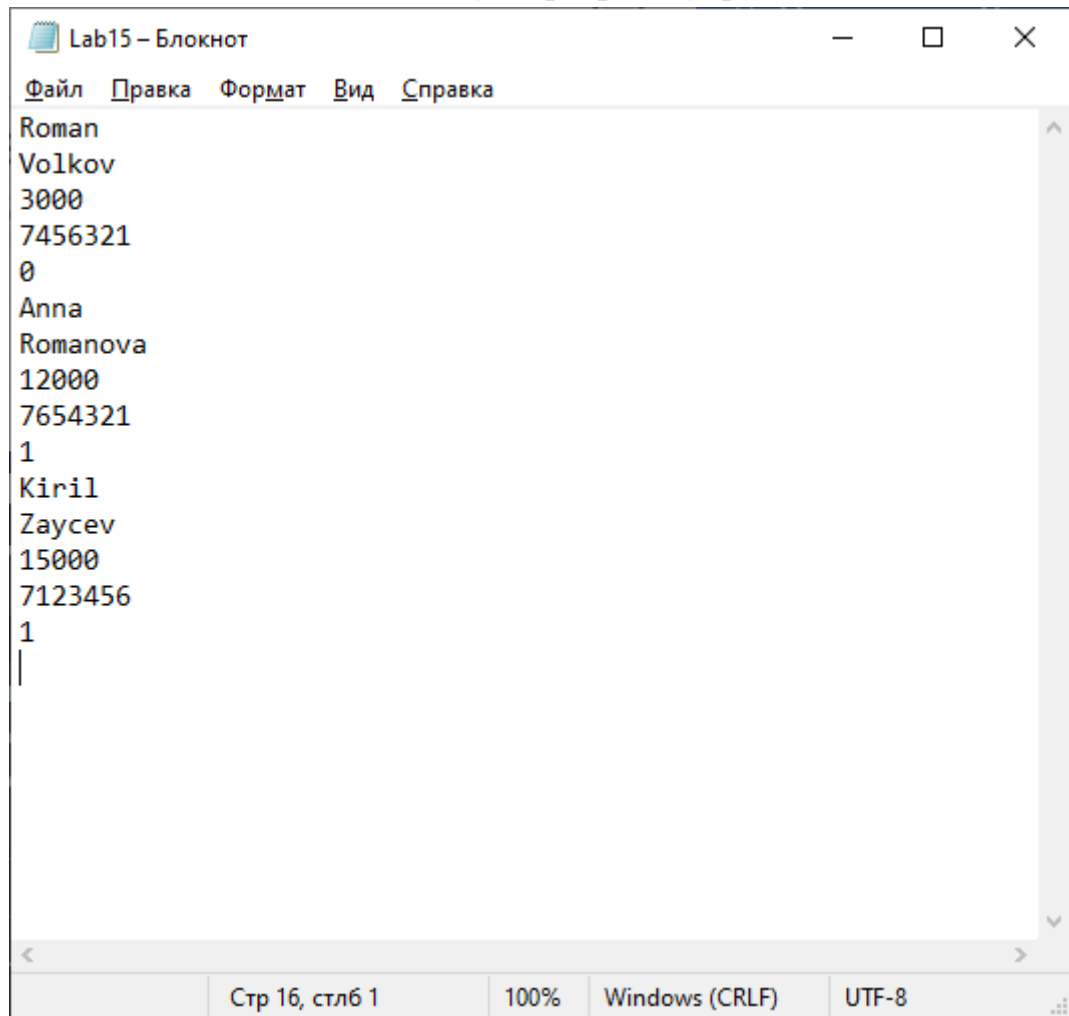
    for (int i = 0; i < size; i++) {
        fprintf(file, debtors[i].fullName.name);
        fprintf(file, "\n");
        fprintf(file, debtors[i].fullName.surname);
        fprintf(file, "\n");
        fprintf(file, "%d\n", debtors[i].debt);
        fprintf(file, "%d\n", debtors[i].phone);
        fprintf(file, "%d\n", debtors[i].insurance);
    }

    fclose(file);
}

```

Рис.5 - Функція для запису до файлу

1. У файлі результат запису інформації трьох боржників.  
(Після запису з програми у файл)



2. Програма зчитує з файлу дані та створює масив співробітників.  
Після запису у консоль масив проходить сортується та записується у файл.

```
Full name: Kiril ZaycevTel.: 7123456
Debt amount: 15000
[The client has insurance]

Full name: Anna RomanovaTel.: 7654321
Debt amount: 12000
[The client has insurance]

Full name: Roman VolkovTel.: 7456321
Debt amount: 3000
```

**Висновок:** Розроблено програму для виконання лабораторної роботи 15, а саме масив структур Боржників та реалізовані для нього функції: створення елементів, запису у консоль та файл, зчитування з файлу, а також сортування масиву.