



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Факультет
Кафедра

Факультет автоматизации и информатики
Автоматизированные системы управления

ЛАБОРАТОРНАЯ РАБОТА 3
на тему: «Обработка элементов структур»

Студент

ПМ-23-2

группа

подпись, дата

Первушин Е.А.

фамилия, инициалы

Руководитель

доцент

ученая степень, ученое звание

подпись, дата

Мирошников А.И.

фамилия, инициалы

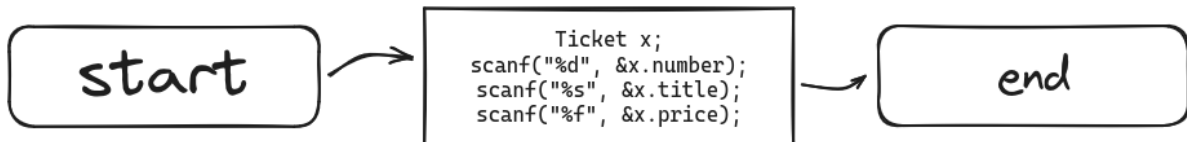
Липецк 2023

ЦЕЛЬ РАБОТЫ: Освоить работу с массивами, элементами которых являются переменные структурного типа данных языка C

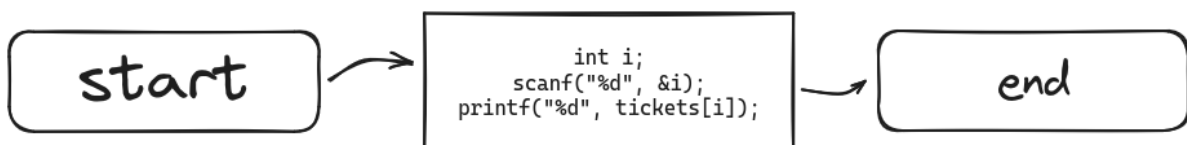
ЗАДАНИЕ: (вариант 9)

Разработать программу, выполняющую приведенные ниже действия. Предусмотреть обработку ошибок и удобный интерфейс с выбором любого пункта меню и возвратом к меню после выполнения выбранного действия:

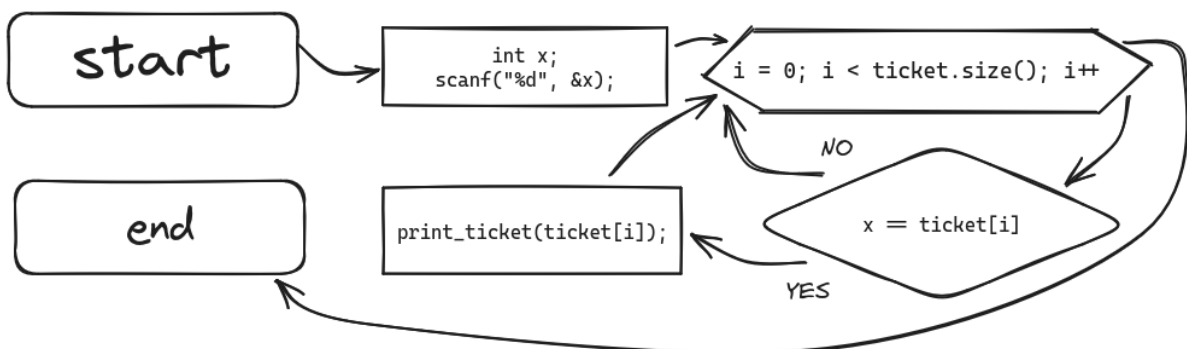
1. Создание и ввод элементов массива структур



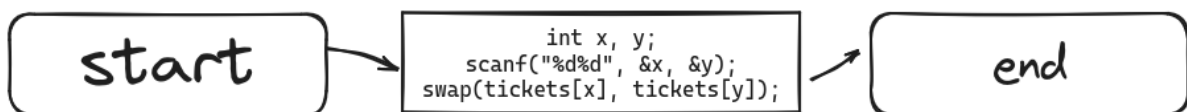
2. Установка/получение значений полей элемента массива структур по заданному индексу в массиве (индекс вводится пользователем)



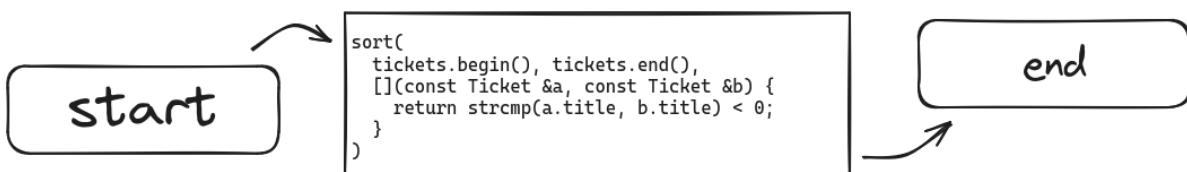
3. Поиск элемента массива структур, в котором значение выбранного поля совпадает со значением поля, введенным пользователем



4. Обмен значений соответствующих полей двух заданных элементов массива структур



5. Упорядочение элементов массива структур по возрастанию/убыванию значений заданного поля



ТЕКСТ ПРОГРАММЫ

```
#include <stdio.h>
#include <cmath>
#include <algorithm>
#include <vector>
#include <cstring>
using namespace std;

/* COLORS TEXT IN TERMINAL */
const char WHITE[] = "\033[38;5;255m";
const char GRAY[] = "\033[38;5;246m";
const char BLUE[] = "\033[38;5;27m";
const char RED[] = "\033[38;5;196m";
const char GREEN[] = "\033[38;5;47m";
const char PURPLE[] = "\033[38;5;200m";
const char NOCOLOR[] = "\033[0m";

void clear() { if (system("clear") != 0) { system("cls"); } }

struct Ticket {
    unsigned int number;
    char title[200];
    float price;
};

void print_ticket(Ticket ticket) {
    printf(
        "%s{ %s%d%s; %s%s%s; %s%.2f %s}%s",
        GRAY, WHITE, ticket.number,
        GRAY, WHITE, ticket.title,
        GRAY, WHITE, ticket.price,
        GRAY, NOCOLOR
    );
}

void print_ticket(vector<Ticket> &tickets) {
    while (true) {
        printf("\n %синдекс билета: %s", GREEN, NOCOLOR);
        int index; scanf("%d", &index);
        if (index >= 0 && index < tickets.size()) {
            printf("\n "); print_ticket(tickets[index]);
            printf("\n"); break;
        } else {
            printf("\n %ОШИБКА: %s нет элемента с таким индексом\n", RED, NOCOLOR);
        }
    }
}

void print_tickets(vector<Ticket> &tickets) {
    for (int i = 0; i < tickets.size(); i++) {
        printf(" %s%d ", PURPLE, i);
        print_ticket(tickets[i]);
        printf("\n");
    }
}

void input_ticket(Ticket *ticket) {
    printf("\n %номер билета: %s", GREEN, NOCOLOR); scanf("%d", &ticket->number);
    printf(" %сзвание билета: %s", GREEN, NOCOLOR); scanf("%s", &ticket->title);
    printf(" %цена билета: %s", GREEN, NOCOLOR); scanf("%f", &ticket->price);
}

void create_ticket(vector<Ticket> &tickets) {
    Ticket temp; input_ticket(&temp);
    tickets.push_back(temp);
}

void edit_ticket(vector<Ticket> &tickets) {
    while (true) {
        printf("\n %синдекс билета: %s", GREEN, NOCOLOR);
        int index; scanf("%d", &index);
    }
}
```

```

        if (index >= 0 && index < tickets.size()) {
            input_ticket(&tickets[index]);
            break;
        } else {
            printf("\n %sОШИБКА: %s нет элемента с таким индексом\n", RED, NOCOLOR);
        }
    }
}

void search_ticket(vector<Ticket> &tickets) {
    printf("\n %s1 %s-> %sпоиск по номеру%s\n", RED, GRAY, GREEN, NOCOLOR);
    printf(" %s2 %s-> %sпоиск по названию%s\n", RED, GRAY, GREEN, NOCOLOR);
    printf(" %s3 %s-> %sпоиск по цене%s\n", RED, GRAY, GREEN, NOCOLOR);

    int point = -1;
    printf("\n %sвведи номер пункта:%s ", PURPLE, NOCOLOR);
    scanf("%d", &point);

    switch (point) {
        case 1: {
            printf("\n %sномер: %s", GREEN, NOCOLOR);
            int temp; scanf("%d", &temp);

            printf("\n");
            for (int i = 0; i < tickets.size(); i++) {
                if (tickets[i].number == temp) {
                    print_ticket(tickets[i]); printf("\n");
                }
            }

            break;
        } case 2: {
            printf("\n %sназвание: %s", GREEN, NOCOLOR);
            char temp[200]; scanf("%s", &temp);

            printf("\n");
            for (int i = 0; i < tickets.size(); i++) {
                if (strcmp(tickets[i].title, temp) == 0) {
                    print_ticket(tickets[i]); printf("\n");
                }
            }

            break;
        } case 3: {
            printf("\n %sцена: %s", GREEN, NOCOLOR);
            float temp; scanf("%f", &temp);

            printf("\n");
            for (int i = 0; i < tickets.size(); i++) {
                if (tickets[i].price == temp) {
                    print_ticket(tickets[i]); printf("\n");
                }
            }

            break;
        } default: {
            printf("\n %sОШИБКА: %sданные не подходят\n", RED, NOCOLOR);
            scanf("%*[^\\r\\n]");
        }
    }
}

void swap_ticket(vector<Ticket> &tickets) {
    printf("\n %sиндекс 1 билета: %s", GREEN, NOCOLOR);
    int index1; scanf("%d", &index1);

    if (!(index1 >= 0 && index1 < tickets.size())) {
        printf("\n %sОШИБКА: %s нет элемента с таким индексом\n", RED, NOCOLOR);
        return;
    }

    printf(" %sиндекс 2 билета: %s", GREEN, NOCOLOR);
    int index2; scanf("%d", &index2);

    if (!(index2 >= 0 && index2 < tickets.size())) {

```

```

        printf("\n %sОШИБКА: %s нет элемента с таким индексом\n", RED, NOCOLOR);
        return;
    }

    swap(tickets[index1], tickets[index2]);
}

void sort_ticket(vector<Ticket> &tickets) {
    sort(
        tickets.begin(), tickets.end(),
        [](const Ticket &a, const Ticket &b) { return strcmp(a.title, b.title) < 0; }
    );
}

int main() {
    clear();
    vector<Ticket> tickets;

    while (true) {
        clear();
        printf("\n      %sОБРАБОТКА ЭЛЕМЕНТОВ СТРУКТУР %s(ticket)%s\n\n", BLUE, GRAY, NOCOLOR);

        if (tickets.size() > 0) {
            print_tickets(tickets);
            printf("\n");
        }

        printf(" %s1 %s-> %sсоздать билет%s\n", RED, GRAY, GREEN, NOCOLOR);
        printf(" %s2 %s-> %sизменить билет%s\n", RED, GRAY, GREEN, NOCOLOR);
        printf(" %s3 %s-> %sполучить билет по индексу%s\n", RED, GRAY, GREEN, NOCOLOR);
        printf(" %s4 %s-> %sпоиск билетов%s\n", RED, GRAY, GREEN, NOCOLOR);
        printf(" %s5 %s-> %sпоменять местами билеты%s\n", RED, GRAY, GREEN, NOCOLOR);
        printf(" %s6 %s-> %sсортировать билеты по названию%s\n", RED, GRAY, GREEN, NOCOLOR);

        printf("\n %s0 %s-> %sзавершить программу %s\n", RED, GRAY, GREEN, NOCOLOR);

        while (true) {
            int point = -1;
            printf("\n %sвведи номер пункта:%s ", PURPLE, NOCOLOR);
            scanf("%d", &point);

            switch (point) {
                case 0: { return 0; }
                case 1: { create_ticket(tickets); break; }
                case 2: { edit_ticket(tickets); break; }
                case 3: { print_ticket(tickets); continue; }
                case 4: { search_ticket(tickets); continue; }
                case 5: { swap_ticket(tickets); break; }
                case 6: { sort_ticket(tickets); break; }
                default: {
                    printf("\n %sОШИБКА: %sданные не подходят\n", RED, NOCOLOR);
                    scanf("%*[^\\r\\n]");
                    continue;
                }
            }

            break;
        }
    }

    return 0;
}

```

ПРИМЕР ВЫПОЛНЕНИЯ

→ lab-3 git:(main) x g++ -o main main.cpp

→ lab-3 git:(main) x ./main

/* clear */

ОБРАБОТКА ЭЛЕМЕНТОВ СТРУКТУР (ticket)

- 1 -> создать билет
- 2 -> изменить билет
- 3 -> получить билет по индексу
- 4 -> поиск билетов
- 5 -> поменять местами билеты
- 6 -> сортировать билеты по названию

0 -> завершить программу

введи номер пункта: 1

номер билета: 69

название билета: Eva-Elfie

цена билета: 150

/* clear */

ОБРАБОТКА ЭЛЕМЕНТОВ СТРУКТУР (ticket)

0 { 69; Eva-Elfie; 150.00 }

- 1 -> создать билет
- 2 -> изменить билет
- 3 -> получить билет по индексу
- 4 -> поиск билетов
- 5 -> поменять местами билеты
- 6 -> сортировать билеты по названию

0 -> завершить программу

введи номер пункта: 4

- 1 -> поиск по номеру
- 2 -> поиск по названию
- 3 -> поиск по цене

введи номер пункта: 1

номер: 69

{ 69; Eva-Elfie; 150.00 }

введи номер пункта: 0

→ lab-3 git:(main) x