

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №1
По дисциплине ОАиП за II семестр
Тема: «Структуры, объединения, перечисления»

Выполнила:
Студентка
Группы ИИ-18
1-го курса
Троцюк О. А.

Проверил:
Михно Е.В.

Брест 2020

Цель: Изучить структуры, объединения и перечисления в языке Си.

Ход работы:

Задание №1

Создайте структуру (в соответствии с вариантом задания задания), данная структура должна содержать: битовое поле, поле типа - объединение, поле типа - перечисление. Ввести массив структур в соответствии с вариантом. Рассортировать массив в алфавитном порядке по первому полю, входящему в структуру. В программе реализовать меню:

- 1) Ввод массива структур;
- 2) Вывод массива структур;
- 3) Сортировка массива структур;
- 3) Поиск в массиве структур по заданному параметру;
- 4) Изменение заданной структуры;
- 5) Удаление структуры из массива;
- 6) Вывод на экран массива структур;
- 7) Выход.

Задание №2

- 6) Организовать динамический массив структур(в соответствии с вариантом);
- 7) Изменить код программы , полученный в предыдущем задании №1, для динамического массива структур.
- 8) Объявить указатель на массив структур . Организовать работу с полями структур через данный указатель.

Вариант 16

6. Структура «Школьник»: фамилия, имя, отчество; класс; номер телефона; оценки по предметам (математика, физика, русский язык, литература).

Текст программы:

```
#include <iostream>
#include <malloc.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

using namespace std;
void menu() {

    cout << "1 - Enter array of structures"
        << "\n";
    cout << "2 - Display array of structures"
        << "\n";
    cout << "3 - Sort out array of structures"
        << "\n";
    cout << "4 - Search/change/delete a certain structure by a parameter"
        << "\n";
    cout << "5 - Escape"
        << "\n";
}

union marks {
    int maths, physics, russian, literature;
};

typedef enum Mark { MARK_D, MARK_C, MARK_B, MARK_A } Mark;

typedef struct Pupil {
    char first_name[20];
    char last_name[20];
    char generic_name[30];
    unsigned int phone_number : 24;
    char adress[20];
    unsigned int grade : 4;
    Mark maths_mark;
    Mark physics_mark;
    Mark russian_mark;
    Mark literature_mark;
} Scholar;
```

```

void enter_pupil(Pupil *pupils, int &enter, int &n) {
    int grade, phone;
    cout << "Enter first name"
        << "\n";
    cin >> pupils[enter].first_name;
    cout << "Enter last name"
        << "\n";
    cin >> pupils[enter].last_name;
    cout << "Enter generic name"
        << "\n";
    cin >> pupils[enter].generic_name;
    cout << "Enter phone number"
        << "\n";
    cin >> phone;
    pupils[enter].phone_number = phone;
    cout << "Enter adress"
        << "\n";
    cin >> pupils[enter].adress;
    cout << "Enter grade"
        << "\n";
    cin >> grade;
    pupils[enter].grade = grade;
    cout << "Enter maths mark from 0 to 3"
        << "\n";
    int mark;
    cin >> mark;
    pupils[enter].maths_mark = (Mark)mark;
    cout << "Enter physics mark from 0 to 3"
        << "\n";
    cin >> mark;
    pupils[enter].physics_mark = (Mark)mark;
    cout << "Enter russian mark from 0 to 3"
        << "\n";
    cin >> mark;
    pupils[enter].russian_mark = (Mark)mark;
    cout << "Enter literature mark from 0 to 3"
        << "\n";
    cin >> mark;
    pupils[enter].literature_mark = (Mark)mark;
    enter++;
}

```

```

void display_pupil(Pupil *pupils, int &enter, int &i) {
    for (int i = 0; i < enter; i++) {
        cout << "name: " << pupils[i].first_name << "\n";
        cout << "last name: " << pupils[i].last_name << "\n";
        cout << "generic name: " << pupils[i].generic_name << "\n";
        cout << "phone number: " << pupils[i].phone_number << "\n";
        cout << "adress: " << pupils[i].adress << "\n";
        cout << "grade: " << pupils[i].grade << "\n";
        cout << "Maths: " << pupils[i].maths_mark << "\n";
        cout << "Physics: " << pupils[i].physics_mark << "\n";
        cout << "Russian: " << pupils[i].russian_mark << "\n";
        cout << "Literature: " << pupils[i].literature_mark << "\n"
            << "\n";
    }
}

```

```

void sort_out(Pupil *pupils, int &enter) {
    Pupil temp;
    for (int i = 0; i < enter; i++) {
        for (int j = enter - 1; j > i; j--) {
            if ((pupils[j - 1].first_name[0] > pupils[j].first_name[0])) {
                temp = pupils[j - 1];
                pupils[j - 1] = pupils[j];
                pupils[j] = temp;
            }
        }
    }
}

```

```

void parameter() {
    cout << "1 - first name"
        << "\n";
    cout << "2 - last name"
        << "\n";
    cout << "3 - grade"
        << "\n";
}

```

```

void option_menu() {
    cout << "1 - display"
        << "\n";
    cout << "2 - change"

```

```

        << "\n";
    cout << "3 - delete"
        << "\n";
}

```

```

void option(Pupil *pupils, int &enter, int &index_to_change) {
    int opt;
    int not_equal = -1;
    cout << "choose option"
        << "\n";
    option_menu();
    cin >> opt;
    switch (opt) {
    case 1: {
        int condition = index_to_change + 1;
        display_pupil(pupils, condition, index_to_change);
        break;
    }
    case 2: {
        enter_pupil(pupils, index_to_change, not_equal);
        break;
    }
    case 3: {

        for (int i = index_to_change; i < enter; i++) {
            pupils[i] = pupils[i + 1];
        }
        enter--;
    }
    }
}

```

```

void search(Pupil *pupils, int &enter) {
    char name[20];
    char last[20];
    int grd;
    Pupil temp;
    cout << "choose the parameter for searching"
        << "\n";
    int k;
    parameter();
    cin >> k;
}

```

```

switch (k) {
case 1: {
    cout << "enter first name"
        << "\n";
    cin >> temp.first_name;
    for (int i = 0; i < enter; i++) {
        if (strcmp(temp.first_name, pupils[i].first_name) == 0) {
            option(pupils, enter, i);
            break;
        }
    }
    break;
}
case 2: {
    cout << "enter last name"
        << "\n";
    cin >> temp.last_name;
    for (int i = 0; i < enter; i++) {
        if (strcmp(temp.last_name, pupils[i].last_name) == 0) {
            option(pupils, enter, i);
            break;
        }
    }
    break;
}
case 3: {
    cout << "enter first grade"
        << "\n";
    cin >> grd;
    for (int i = 0; i < enter; i++) {
        if (grd == pupils[i].grade) {
            option(pupils, enter, i);
            break;
        }
    }
    break;
}
}

main() {
    int key;

```

```

int n = 5;
int enter = 0;
Pupil *pupils;
pupils = (Pupil *)malloc(n * sizeof(Pupil));
bool process = 1;
while (process) {
    if (enter == n) {
        n *= 2;
        pupils = (Pupil *)realloc(pupils, n * sizeof(Pupil));
    }
    cout << "Enter key"
        << "\n";
    menu();
    cin >> key;
    switch (key) {
    case 1: {
        enter_pupil(pupils, enter, n);
        break;
    }
    case 2: {
        int k = 0;
        display_pupil(pupils, enter, k);
        break;
    }
    case 3: {
        sort_out(pupils, enter);
        break;
    }
    case 4: {
        search(pupils, enter);
        break;
    }
    case 5: {
        exit(1);
        break;
    }
    }
}
}

```


Результаты тестирования программы:

1. Ввод структуры в массив:

```
Enter key
1 - Enter array of structures
2 - Display array of structures
3 - Sort out array of structures
4 - Search/change/delete a certain structure by a parameter
5 - Escape
1
Enter first name
Tom
Enter last name
DeLonge
Enter generic name
Matthew
Enter phone number
1234567
Enter adress
anotherplanet
Enter grade
11
Enter maths mark from 0 to 3
2
Enter physics mark from 0 to 3
1
Enter russian mark from 0 to 3
3
Enter literature mark from 0 to 3
2
Enter key
1 - Enter array of structures
2 - Display array of structures
3 - Sort out array of structures
4 - Search/change/delete a certain structure by a parameter
5 - Escape
```

2. Вывод массива после ввода
еще двух структур:

```
2
name: Tom
last name: Delonge
generic name: Matthew
phone number: 1234567
adress: anotherplanet
grade: 11
Maths: 2
Physics: 1
Russian: 3
Literature: 2
```

```
name: Travis
last name: Barker
generic name: Matthew
phone number: 1111111
adress: earth
grade: 11
Maths: 2
Physics: 3
Russian: 1
Literature: 2
```

```
name: Mark
last name: Hoppus
generic name: Thomas
phone number: 4444444
adress: newblink
grade: 10
Maths: 2
Physics: 1
Russian: 3
Literature: 2
```

3. Сортировка массива
по первой букве имени:

```
name: Mark
last name: Hoppus
generic name: Thomas
phone number: 4444444
adress: newblink
grade: 10
Maths: 2
Physics: 1
Russian: 3
Literature: 2
```

```
name: Tom
last name: Delonge
generic name: Matthew
phone number: 1234567
adress: anotherplanet
grade: 11
Maths: 2
Physics: 1
Russian: 3
Literature: 2
```

```
name: Travis
last name: Barker
generic name: Matthew
phone number: 1111111
adress: earth
grade: 11
Maths: 2
Physics: 3
Russian: 1
Literature: 2
```

4. Поиск по параметру и вывод этой структуры:

```
4 - Search/change/delete a certain structure by a parameter
5 - Escape
4
choose the parameter for searching
1 - first name
2 - last name
3 - grade
1
enter first name
Mark
choose option
1 - display
2 - change
3 - delete
1
name: Mark
last name: Hoppus
generic name: Thomas
phone number: 4444444
adress: newblink
grade: 10
Maths: 2
Physics: 1
Russian: 3
Literature: 2
```

5. Поиск по параметру и удаление структуры “Mark”, вывод структур:

```
Mark
choose option
1 - display
2 - change
3 - delete
3
Enter key
1 - Enter array of structures
2 - Display array of structures
3 - Sort out array of structure:
4 - Search/change/delete a certa
5 - Escape
2
name: Tom
last name: Delonge
generic name: Matthew
phone number: 1234567
adress: anotherplanet
grade: 11
Maths: 2
Physics: 1
Russian: 3
Literature: 2

name: Travis
last name: Barker
generic name: Matthew
phone number: 1111111
adress: earth
grade: 11
Maths: 2
Physics: 3
Russian: 1
Literature: 2
```

6. Вывод массива структур после поиска структуры “Tom” по параметру и замены ее на структуру “Matt”:

```
2
name: Matt
last name: Skiba
generic name: Matthew
phone number: 1234567
adress: thisplanet
grade: 11
Maths: 2
Physics: 1
Russian: 3
Literature: 2
```

```
name: Travis
last name: Barker
generic name: Matthew
phone number: 1111111
adress: earth
grade: 11
Maths: 2
Physics: 3
Russian: 1
Literature: 2
```

Вывод: В ходе лабораторной работы ознакомилась со структурами в языке Си, реализовала ввод, вывод, сортировку, изменение и удаление структур в динамическом массиве.