|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |
| Институт искусственного интеллекта |
| Кафедра технологий искусственного интеллекта |

ПРАКТИЧЕСКАЯ РАБОТА № 15

по дисциплине

«Процедурное программирование»

Обучающийся Сысоенко Глеб Максимович

Группа КАБО-01-23

Руководитель *Яковлев Д. А*

Москва 2023

**Практическая работа №15.**

**Тема: «Структуры данных в Си»**

**Цель лабораторной работы:**

Целью данной лабораторной работы освоить на практике работу с структурами на языке Си.

**Описание:**

Написать программу, в которой будут функции, которые будут выполнять действия заданные заданием, со структурами. Выполнить задание в соответствии с вариантом.

**Задание:**

Структура «Школьник».

Минимальный набор полей: фамилия, имя, пол, класс, дата рождения, адрес.

Список: двусвязный упорядоченный (по фамилии) список.

Поиск по фамилии, фильтр по классу.

**Код программы:**

#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include<conio.h>  
typedef struct student {  
 char id;  
 char lastName[50];  
 char firstName[50];  
 char gender;  
 int class;  
 char dateOfBirth[20];  
 char address[100];  
 struct student \*prev;  
 struct student \*next;  
} Student;  
  
void addStudent(Student \*\*head, Student \*\*tail);  
void deleteStudent(Student \*\*head, Student \*\*tail, char lastName[50]);  
void searchStudent(Student \*head, char lastName[50]);  
void displayStudent(Student \*head);  
void displaySearch(Student \*head, char lastName[50]);  
void OutToFileStudent(Student \*head);  
void filter\_output(Student \*head, int op, int val);  
void OutToFileStudent\_Personal(Student \*head, char lastName[50]);  
void searchStudent\_to\_file(Student \*head, char lastName[50]);  
void add\_data\_from\_file(Student \*\*head);  
  
int main() {  
 int choice;  
 Student \*head = NULL;  
 Student \*tail = NULL;  
  
 while (1) {  
 printf("\n\n \*\*\*\*MAIN MENU\*\*\*\*\n");  
 printf("1. Add a new student.\n");  
 printf("2. Delete a student.\n");  
 printf("3. Search for a student.\n");  
 printf("4. Filtred students.\n");  
 printf("5. Display students.\n");  
 printf("6. Output to a file (all).\n");  
 printf("7. Output to a file (1 Student).\n");  
 printf("8. AAdding students from a file\n");  
 printf("9. Exit.\n");  
 printf("Enter your choice: ");  
 scanf("%d", &choice);  
  
 switch (choice) {  
 case 1:  
 addStudent(&head, &tail);  
 break;  
 case 2: {  
 char lastName[50];  
 printf("Enter the last name of the student you want to delete: ");  
 scanf("%s", lastName);  
 deleteStudent(&head, &tail, lastName);  
 break;  
 }  
 case 3: {  
 char lastName[50];  
 printf("Enter the last name of the student you want to search: ");  
 scanf("%s", lastName);  
 searchStudent(head, lastName);  
 break;  
 }  
 case 4: {  
 int order, vb;  
 printf("The value of the class: ");  
 scanf("%d", &order);  
 printf("\nType operation\n"  
 "Equally (=) - 1\n"  
 "More (>) - 2\n"  
 "Less (<) - 3\n"  
 "Your choice: ");  
 scanf("%d", &vb);  
 if (vb > 0 && vb < 4){  
 printf("\n[System]: Operation type successfully.\n");  
 filter\_output(head, vb, order);  
 }  
 else{  
 printf("Invalid choice.\n");  
 break;  
 }  
 break;  
 }  
 case 5:  
 displayStudent(head);  
 break;  
 case 6:  
 OutToFileStudent(head);  
 break;  
 case 7:{  
 char lastName[50];  
 printf("Enter the last name of the student you want to search: ");  
 scanf("%s", lastName);  
 searchStudent\_to\_file(head, lastName);  
 break;  
 }  
 case 8:{  
 add\_data\_from\_file(&head);  
 Student \*temp = head;  
 break;  
 }  
 case 9:  
 exit(0);  
 default:  
 printf("Invalid choice. Please enter a valid choice.\n");  
 }  
 }  
}  
  
void addStudent(Student \*\*head, Student \*\*tail) {  
 Student \*newStudent = (Student \*)malloc(sizeof(Student));  
 printf("Enter the last name of the student: ");  
 scanf("%s", newStudent->lastName);  
 printf("Enter the first name of the student: ");  
 scanf("%s", newStudent->firstName);  
 printf("Enter the gender of the student (M/F): ");  
 scanf(" %c", &newStudent->gender);  
 printf("Enter the class of the student: ");  
 scanf("%d", &newStudent->class);  
 printf("Enter the date of birth of the student (DD-MM-YYYY): ");  
 scanf("%s", newStudent->dateOfBirth);  
 printf("Enter the address of the student: ");  
 scanf("%s", newStudent->address);  
  
 newStudent->prev = NULL;  
 newStudent->next = NULL;  
  
 if (\*head == NULL) {  
 \*head = newStudent;  
 \*tail = newStudent;  
 } else {  
 Student \*temp = \*head;  
 while (temp->next != NULL) {  
 temp = temp->next;  
 }  
 temp->next = newStudent;  
 newStudent->prev = temp;  
 \*tail = newStudent;  
 }  
 printf("Student added successfully.\n");  
}  
  
void deleteStudent(Student \*\*head, Student \*\*tail, char lastName[50]) {  
 Student \*temp = \*head;  
 while (temp != NULL) {  
 if (strcmp(temp->lastName, lastName) == 0) {  
 if (temp->prev != NULL) {  
 temp->prev->next = temp->next;  
 } else {  
 \*head = temp->next;  
 }  
 if (temp->next != NULL) {  
 temp->next->prev = temp->prev;  
 } else {  
 \*tail = temp->prev;  
 }  
 free(temp);  
 printf("Student deleted successfully.\n");  
 return;  
 }  
 temp = temp->next;  
 }  
 printf("Student not found.\n");  
}  
  
void searchStudent(Student \*head, char lastName[50]) {  
 Student \*temp = head;  
 while (temp != NULL) {  
 if (strcmp(temp->lastName, lastName) == 0) {  
 displaySearch(head, lastName);  
 return;  
 }  
 temp = temp->next;  
 }  
 printf("Student not found.\n");  
}  
  
void filter\_output(Student \*head, int op, int val) {  
 Student \*temp = head;  
 while(temp != NULL) {  
 if(op == 1 && temp->class == val) {  
 printf("\nFirst Name: %s\n", temp->firstName);  
 printf("Last Name: %s\n", temp->lastName);  
 printf("Gender: %c\n", temp->gender);  
 printf("Class: %d\n", temp->class);  
 printf("Date of Birth: %s\n", temp->dateOfBirth);  
 printf("Address: %s\n", temp->address);  
 } else if(op == 2 && temp->class > val) {  
 printf("\nFirst Name: %s\n", temp->firstName);  
 printf("Last Name: %s\n", temp->lastName);  
 printf("Gender: %c\n", temp->gender);  
 printf("Class: %d\n", temp->class);  
 printf("Date of Birth: %s\n", temp->dateOfBirth);  
 printf("Address: %s\n", temp->address);  
 } else if(op == 3 && temp->class < val) {  
 printf("\nFirst Name: %s\n", temp->firstName);  
 printf("Last Name: %s\n", temp->lastName);  
 printf("Gender: %c\n", temp->gender);  
 printf("Class: %d\n", temp->class);  
 printf("Date of Birth: %s\n", temp->dateOfBirth);  
 printf("Address: %s\n", temp->address);  
 }  
 temp = temp->next;  
 }  
}  
void displayStudent(Student \*head) {  
 Student \*temp = head;  
 while (temp != NULL) {  
 printf("\nFirst Name: %s\n", temp->firstName);  
 printf("Last Name: %s\n", temp->lastName);  
 printf("Gender: %c\n", temp->gender);  
 printf("Class: %d\n", temp->class);  
 printf("Date of Birth: %s\n", temp->dateOfBirth);  
 printf("Address: %s\n", temp->address);  
 temp = temp->next;  
 }  
}  
  
void displaySearch(Student \*head, char lastName[50]) {  
 Student \*temp = head;  
 while (temp != NULL) {  
 if (strcmp(temp->lastName, lastName) == 0) {  
 printf("First Name: %s\n", temp->firstName);  
 printf("Last Name: %s\n", temp->lastName);  
 printf("Gender: %c\n", temp->gender);  
 printf("Class: %d\n", temp->class);  
 printf("Date of Birth: %s\n", temp->dateOfBirth);  
 printf("Address: %s\n", temp->address);  
 return;  
 }  
 temp = temp->next;  
 }  
}  
void OutToFileStudent(Student \*head){  
 char variant[50];  
 printf("How name file?\n");  
 printf("Your variant:");  
 scanf("%s", variant);  
 FILE \*file = fopen(variant, "w");  
 Student \*temp = head;  
 while (temp != NULL){  
 fprintf(file, "%s\n", temp->firstName);  
 fprintf(file, "%s\n", temp->lastName);  
 fprintf(file, "%c\n", temp->gender);  
 fprintf(file, "%d\n", temp->class);  
 fprintf(file, "%s\n", temp->dateOfBirth);  
 fprintf(file, "%s\n\n", temp->dateOfBirth);  
 temp = temp->next;  
 }  
 fclose(file);  
 printf("The transfer of data to the file has been completed successfully");  
}  
  
void searchStudent\_to\_file(Student \*head, char lastName[50]) {  
 Student \*temp = head;  
 while (temp != NULL) {  
 if (strcmp(temp->lastName, lastName) == 0) {  
 OutToFileStudent\_Personal(head, lastName);  
 return;  
 }  
 temp = temp->next;  
 }  
 printf("Student not found.\n");  
}  
  
void OutToFileStudent\_Personal(Student \*head, char lastName[50]){  
 char variant[50];  
 printf("\nHow name file?\n");  
 printf("Your variant:");  
 scanf("%s", variant);  
 FILE \*file = fopen(variant, "w");  
 Student \*temp = head;  
 while (temp != NULL){  
 if (strcmp(temp->lastName, lastName) == 0){  
 fprintf(file, "%s\n", temp->firstName);  
 fprintf(file, "%s\n", temp->lastName);  
 fprintf(file, "%c\n", temp->gender);  
 fprintf(file, "%d\n", temp->class);  
 fprintf(file, "%s\n", temp->dateOfBirth);  
 fprintf(file, "%s\n\n", temp->dateOfBirth);  
 }  
 temp = temp->next;  
 }  
 fclose(file);  
 printf("\nThe transfer of data to the file has been completed successfully");  
}  
  
void insert\_ordered(Student \*\*head, char last\_name[], char first\_name[], char gender, int class, char date\_of\_birth[], char address[]) {  
 Student \*new\_node = (Student\*) malloc(sizeof(Student));  
 strcpy(new\_node->lastName, last\_name);  
 strcpy(new\_node->firstName, first\_name);  
 new\_node->gender = gender;  
 new\_node->class = class;  
 strcpy(new\_node->dateOfBirth, date\_of\_birth);  
 strcpy(new\_node->address, address);  
 new\_node->next = NULL;  
 new\_node->prev = NULL;  
  
 if (\*head == NULL) {  
 \*head = new\_node;  
 } else {  
 Student \*temp = \*head;  
 while (temp->next != NULL && strcmp(temp->next->lastName, new\_node->lastName) < 0) {  
 temp = temp->next;  
 }  
 new\_node->next = temp->next;  
 if (temp->next != NULL) {  
 temp->next->prev = new\_node;  
 }  
 temp->next = new\_node;  
 new\_node->prev = temp;  
 }  
}  
  
void add\_data\_from\_file(Student \*\*head) {  
 char variant[50];  
 printf("How name file?\n");  
 printf("Your variant:");  
 scanf("%s", variant);  
 FILE \*file = fopen(variant, "r");  
 if (file == NULL) {  
 printf("Could not open the file.\n");  
 return;  
 }  
  
 char last\_name[50];  
 char first\_name[50];  
 char gender;  
 int class;  
 char date\_of\_birth[12];  
 char address[100];  
  
 while (fscanf(file, "%s %s %c %d %s %[^\n]\n", last\_name, first\_name, &gender, &class, date\_of\_birth, address) != EOF) {  
 insert\_ordered(head, last\_name, first\_name, gender, class, date\_of\_birth, address);  
 }  
 fclose(file);  
 printf("Students added successfully.\n");  
}