

# Documentație proiect ATP

Orzanescu George grupa: 1130

## 1. Descrierea problemei:

Se dorește crearea unei aplicații de gestiune clienți pentru o piscina dedicată cursurilor de înot copii. Aplicația urmărește înregistrarea clienților și posibilitatea efectuării a diferite operațiuni, totul în vederea creșterii eficienței și a posibilității unei analize ulterioare a datelor.

## 2. Proiectarea aplicației:

Aplicația va avea un meniu standard, ce va facilita utilizatorului accesul la diferitele opțiuni.

```
#####
#####                               #####
#####      Sistem gestiune clienti piscina in C      #####
#####                               #####
#####-----#####

1.Creare fisier binar
2.Adaugare clienti
3.Modificarea instructorului repartizat unui client
4.Modificare abonamente grup clienti
5.Stergerea unui client
6.Raport fisier text clienti
7.Raport partial fisier text clienti (in functie de instructor)
8.Afisare client dupa codul de inregistrare
0.Iesire program
```

Datele clienților vor fi înregistrate, secvențial sub forma de articole într-un fișier binar. Articolele vor avea structura:

| Cod client | Nume client | Tip abonament | Preț abonament | Instructor repartizat | Data înregistrării |
|------------|-------------|---------------|----------------|-----------------------|--------------------|
| int        | char        | char          | int            | char                  | char               |

Menționez că pentru prețul abonamentului și tipul abonamentului sunt definite constante, utilizatorul având posibilitatea de a alege dintre acestea dar nu și de a modifica aceste valori.

```
// constante
const char LUNAR[] = {"1 luna"}; const int pretLUNAR=800;
const char TRIMESTRIAL[] ={"3 luni"}; const int pretTRIMESTRIAL=2000;
const char ANUAL[] = {"1 an"}; const int pretANUAL=6600;
```

### 3. In funcție de operațiunea dorita prezentam exemple de utilizare:

1. **Creare de fișier nou** , organizat secvențial cu extensia .dat. Datele vor fi introduse prin intermediul tastaturii.

```
Optiune -->:1

Cum doriti sa denumiti fisierul (.dat): clienti.dat

Fisierul clienti.dat a fost creat cu succes!
Press any key to continue . . .
```

2. **Adăugare de clienți (articole)** in fișierul creat anterior. Datele vor fi introduse prin intermediul tastaturii. Utilizatorul va folosi ctrl + Z pentru a marca sfârșitul introducerii.

```
Optiune -->:2

Codul clientului: 1

Nume client:Matei Bogdan

Tip abonament (Alege 1 pentru lunar | 2 pentru trimestrial | 3 pentru anual):1

Instructor repartizat: Alexandru L.

Codul clientului: 2

Nume client:Ana Maria

Tip abonament (Alege 1 pentru lunar | 2 pentru trimestrial | 3 pentru anual):3

Instructor repartizat: Cristina P.

Codul clientului: ^Z
Press any key to continue . . .
```

3. **Actualizarea instructorului repartizat unui client (articol identificat unic)**. Datele vor fi introduse prin intermediul tastaturii. Utilizatorul va folosi ctrl + Z pentru a marca sfârșitul introducerii.

```
Optiune -->:3

Nume client: Ana Maria

Cod client:2 Nume: Ana Maria
Pret abonament: 6600 Tip abonament: 1 an Instructor: Cristina P.
Data start abonament: Apr 5 2021

Noul instructor:Alexandru L.

Datele au fost modificate cu success!

Nume client: ^Z
Press any key to continue . . .
```

4. **Actualizare abonament pentru un grup de clienți ( grup de articole).** Date vor fi introduse prin intermediul tastaturii. Utilizatorul va folosi ctrl + Z pentru a marca sfârșitul introducerii.

```
Optiune -->:4

Tip abonament| 1 luna | 3 luni | 1 an| :1 an

Client Matei Bogdan cu abonament 1 an gasit.

Noul tip de abonament (Alege 1 pentru lunar | 2 pentru trimestrial | 3 pentru anual) :2

Datele au fost modificate cu success!

Tip abonament| 1 luna | 3 luni | 1 an|:^Z
Press any key to continue . . .
```

5. **Ștergerea unui client (articol identificat unic) din fișierul binar.** Date vor fi introduse prin intermediul tastaturii. Utilizatorul va folosi ctrl + Z pentru a marca sfârșitul introducerii.

```
Optiune -->:5

Introduceti codul clientului de sters:1

Client sters cu succes.....Press any key to continue . . .
```

6. **Raport integral generat ca fișier text .** Utilizatorul va introduce prin intermediul tastaturii numele dorit pentru raport.

```
Optiune -->:6

Cum doriti sa numiti raportul:raport.txt

Raportul a fost creat cu succes!
Press any key to continue . . .
```

≡ raport.txt

|   | Nr. | Cod | Nume cLient      | Tip abonament | Pret RON | Data        | Instructor   |
|---|-----|-----|------------------|---------------|----------|-------------|--------------|
| 1 |     |     |                  |               |          |             |              |
| 2 | 1   | 1   | Victor Pascu     | 3 luni        | 2000     | Apr 18 2021 | Alexandru L. |
| 3 |     |     |                  |               |          |             |              |
| 4 | 2   | 2   | Maria Bosoaca    | 1 luna        | 800      | Apr 18 2021 | Marian P.    |
| 5 | 3   | 3   | Vasile Paraschiv | 1 luna        | 800      | Apr 18 2021 | Alexandru L. |
| 6 | 4   | 4   | Cristina Lascu   | 3 luni        | 2000     | Apr 18 2021 | Marian P.    |
| 7 |     |     |                  |               |          |             |              |

**7. Raport parțial pentru un grup de clienți (grup de articole) în funcție de un criteriu.** Utilizatorul va introduce numele instructorului pentru care se dorește raportul cu clienții săi, apoi va introduce numele dorit pentru raport. Utilizatorul va folosi ctrl + Z pentru a marca sfârșitul.

```
Optiune -->:7

Raport clienti ce au instructor pe:Alexandru L.

Cum doriti sa numiti raportul:raport_partial.txt

Raportul raport_partial.txt a fost creat cu succes!

Raport clienti ce au instructor pe:^Z
Press any key to continue . . .
```

  

```
≡ raport_partial.txt
1
2 | Nr. Cod Nume cLient      Tip abonament      Pret RON      Data      Instructor
3 |
4 | 1  1  Victor Pascu      3 luni          2000      Apr 18 2021  Alexandru L.
5 | 2  3  Vasile Paraschiv  1 luna          800       Apr 18 2021  Alexandru L.
```

**8. Afișare client (articol) în funcție de codul de înregistrare.** Utilizatorul va introduce prin intermediul tastaturii, codul clientului căutat pentru afișare . Utilizatorul va folosi ctrl + Z pentru a marca sfârșitul.

```
Optiune -->:8

Introduceti codul clientului dorit:1

Cod:1 Nume:Victor Pascu Tip abonament:3 luni

Pret abonament:2000 Data:Apr 18 2021 Instructor:Alexandru L.
Introduceti codul clientului dorit:^Z
Press any key to continue . . .
```

**9. Ieșire din aplicație .** Utilizatorul va introduce cu ajutorul tastaturii cifra 0 corespunzătoare opțiunii de ieșire program.

```
Optiune -->:0

Multumesc, o zi buna!!!
```

## Codul sursa

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <stdbool.h>

// constante
const char LUNAR[] = {"1 luna"};
const int pretLUNAR = 800;
const char TRIMESTRIAL[] = {"3 luni"};
const int pretTRIMESTRIAL = 2000;
const char ANUAL[] = {"1 an"};
const int pretANUAL = 6600;

// char nume_fisier[20] = {"clienti.dat"}; //doar pentru testare
char nume_fisier[20];

typedef struct
{ // structura de stocat date

    int codClient;
    char nume[20];
    char tipAbonament[14];
    int pretAbonament;
    char instructor[20];
    char data[20];
} Client;

void mesajInitial() // mesaj principal
{

    printf("\t\t\t#####\n#####");
    printf("\n\t\t\t#####\n#####");
    printf("\n\t\t\t##### Sistem gestiune clienti piscina in C\n#####");
    printf("\n\t\t\t#####\n#####");
    printf("\n\t\t\t#####\n#####");
    printf("\n\t\t\t#####\n#####");
    printf("\n\t\t\t-----\n-----\n");
}
```

```

void creare() // creare fisier binar
{
    FILE *f;
    // char nume_fisier[20];
    printf("\n\t\t\tCum doriti sa denumiti fisierul (.dat): ");
    gets(nume_fisier);

    f = fopen(nume_fisier, "wb");
    if (!f)
        printf("\n\t\t\tFisierul nu a putut fi creat!\n");
    else
        printf("\n\t\t\tFisierul %s a fost creat cu succes!\n", nume_fisier);
    fclose(f);
}

void adaugare(char *fisier) // adaugare clienti
{
    FILE *f;
    Client a;
    int abonament = 0;

    f = fopen(fisier, "rb+");
    fseek(f, 0, SEEK_END);

    if (!f)
        printf("\n\t\t\tFisierul nu a putut fi deschis !");
    else
    {
        printf("\n\t\t\tCodul clientului: ");
        scanf("%d", &a.codClient);

        while (!feof(stdin))
        {
            printf("\n\t\t\tNume client:");
            getchar();
            gets(a.nume);
            printf("\n\t\t\tTip abonament (Alege 1 pentru lunar | 2 pentru tri
mestrial | 3 pentru anual):");
            scanf("%d", &abonament);
            if (abonament == 1)
            {
                strcpy(a.tipAbonament, LUNAR);
                a.pretAbonament = pretLUNAR;
            }
            else if (abonament == 2)
            {
                strcpy(a.tipAbonament, TRIMESTRIAL);
                a.pretAbonament = pretTRIMESTRIAL;
            }
        }
    }
}

```

```

    }
    else
    {
        strcpy(a.tipAbonament, ANUAL);
        a.pretAbonament = pretANUAL;
    }
    printf("\n\t\t\tInstructor repartizat: ");
    getchar();
    gets(a.instructor);
    strcpy(a.data, __DATE__);
    fwrite(&a, sizeof(Client), 1, f);
    printf("\n\t\t\tCodul clientului: ");
    scanf("%d", &a.codClient);
}
fclose(f);
}
}

```

```

void modificare_client_instructor(char *fisier)
// actualizare instructor client (dupa numele clientului)
{
    FILE *f;
    Client a;
    char nume[20];
    int i;

    f = fopen(fisier, "rb+");
    if (!f)
        printf("\n\t\t\tFisierul %s nu poate fi deschis", fisier);
    else
    {
        printf("\n\t\t\tNume client: ");
        gets(nume);
        while (!feof(stdin))
        {
            rewind(f);
            fread(&a, sizeof(Client), 1, f);
            i = false;
            while ((!feof(f)) && (!i)) // ! false is true
            {
                if (strcmp(nume, a.nume) == 0)
                {
                    i = true;
                    printf("\n\t\t\tCod client:%d Nume: %s \n\t\t\tPret abonam
ent: %d Tip abonament: %s Instructor: %s \n\t\t\tData start abonament: %s\n",
a.codClient, a.nume, a.pretAbonament, a.tipAbonament, a.instructor, a.data);
                    printf("\n\t\t\tNoul instructor:");
                    gets(a.instructor);

```

```

        fseek(f, ftell(f) - sizeof(Client), 0);
        fwrite(&a, sizeof(Client), 1, f);
        if (!feof(stdin))
            printf("\n\t\t\tDatele au fost modificate cu success!\n");
    }
    fread(&a, sizeof(Client), 1, f);
}
if (!i)
{
    printf("\n\t\t\tClientul nu a fost gasit.\n");
}
printf("\n\t\t\tNume client: ");
gets(ume);
}
fclose(f);
}
}

```

```

void modificare_abonament_clienti(char *fisier)
// actualizare tip abonament clienti(dupa tipul de abonament)
{
    FILE *f;
    Client a;
    char abonament[20];
    int i, d;

    f = fopen(fisier, "rb+");
    if (!f)
        printf("\n\t\t\tFisierul %s nu a putut fi deschis", fisier);
    else
    {
        printf("\n\t\t\tTip abonament| 1 luna | 3 luni | 1 an| :");
        gets(abonament);
        while (!feof(stdin))
        {
            rewind(f); // ne mutam la inceputul fisierului de fiecare data
            fread(&a, sizeof(Client), 1, f);
            i = false;

            while (!feof(f))
            {
                if (strcmp(abonament, a.tipAbonament) == 0)
                {
                    i = true;
                    printf("\n\t\t\tClient %s cu abonament %s gasit.\n", a.num
e, a.tipAbonament);
                }
            }
        }
    }
}

```



```

        printf("\n\t\t\tNoul tip de abonament (Alege 1 pentru luna
r | 2 pentru trimestrial | 3 pentru anual):");
        scanf("%d", &d);
        getchar();
        if (d == 1)
        {
            strcpy(a.tipAbonament, LUNAR);
            a.pretAbonament = pretLUNAR;
        }
        else if (d == 2)
        {
            strcpy(a.tipAbonament, TRIMESTRIAL);
            a.pretAbonament = pretTRIMESTRIAL;
        }
        else
        {
            strcpy(a.tipAbonament, ANUAL);
            a.pretAbonament = pretANUAL;
        }
        fseek(f, ftell(f) - sizeof(Client), 0);
        fwrite(&a, sizeof(Client), 1, f);
        fseek(f, 0, 1);
        if (!feof(stdin))
            printf("\n\t\t\tDatele au fost modificate cu success!\n
n");
    }
    fread(&a, sizeof(Client), 1, f);
}
if (i == false)
{
    printf("\n\t\t\tNu a fost gasit nici un client cu acest tip de
abonament");
}
printf("\n\t\t\tTip abonament| 1 luna | 3 luni | 1 an|:");
gets(abonament);
}
fclose(f);
}
}

void stergere_client(char *fisier) //stergere clienti
{
    FILE *f;
    FILE *tempf;
    Client a;
    int i = false;
    int j;

```

```

f = fopen(fisier, "rb");
if (!f)
    printf("\n\t\t\tFisierul %s nu a putut fi deschis", fisier));
tempf = fopen("temp.dat", "wb");
if (!tempf)
    printf("\n\t\t\tFisierul temp.dat nu a putut fi creat!");
printf("\n\t\t\tIntroduceti codul clientului de sters:");
scanf("%d", &j);

while (fread(&a, sizeof(Client), 1, f))
{
    if (a.codClient != j)
    {
        fwrite(&a, sizeof(Client), 1, tempf);
    }
    else
    {
        i = true;
    }
}
(i) ? printf("\n\t\t\tClient sters cu succes.....\n") : printf("\n\t\t\tCl
ientul cu codul %d nu exista", j);
fclose(f);
fclose(tempf);

remove(fisier);
rename("temp.dat", fisier);
}

void raport_integral(char *fisier) // creare raport fisier text
{
    {
        FILE *f, *t;
        Client a;
        int n;
        char fisierTxt[20];
        f = fopen(fisier, "rb");
        if (!f)
            printf("\n\t\t\tFisierul %s nu a putut fi deschis", fisier);
        else
        {
            printf("\n\t\t\tCum doriti sa numiti raportul:");
            gets(fisierTxt);
            t = fopen(fisierTxt, "w");
            fprintf(t, "\n Nr. Cod Nume cLient \t\tTip abonament \t\tPret RON
\t\tData \t\tInstructor\n");
            fread(&a, sizeof(Client), 1, f);
            n = 0;

```

```

        while (!feof(f))
        {
            fprintf(t, "\n %-3d %-3d %-20s %-20s %-3d      %-
16s %s", ++n, a.codClient, a.num, a.tipAbonament, a.pretAbonament, a.data, a.
instructor);
            fread(&a, sizeof(Client), 1, f);
        }
        printf("\n\t\t\tRaportul a fost creat cu succes!\n");
        fclose(t);
        fclose(f);
    }
}

void raport_partial(char *fisier) // creare raport partial text
{
    FILE *f, *t;
    Client a;
    int n;
    char fisierTxt[20];
    char nume_instr[20];
    f = fopen(fisier, "rb");
    if (!f)
        printf("\n\t\t\tFisierul %s nu a putut fi deschis", fisier);
    else
    {
        printf("\n\t\t\tRaport clienti ce au instructor pe:");
        gets(nume_instr);
        while (!feof(stdin))
        {
            rewind(f);
            printf("\n\t\t\tCum doriti sa numiti raportul:");
            gets(fisierTxt);
            t = fopen(fisierTxt, "w");
            fprintf(t, "\n Nr. Cod Nume cClient \t\tTip abonament \t\tPret RON
\t\tData \t\tInstructor\n");

            fread(&a, sizeof(Client), 1, f);
            n = 1;
            while (!feof(f))
            {
                if (strcmp(nume_instr, a.instructor) == 0)
                {
                    fprintf(t, "\n %-3d %-3d %-20s %-20s %-3d      %-
16s %s", n++, a.codClient, a.num, a.tipAbonament, a.pretAbonament, a.data, a.
instructor);
                }
            }
        }
    }
}

```

```

        fread(&a, sizeof(Client), 1, f);
    }
    if (n > 1)
        printf("\n\t\t\tRaportul %s a fost creat cu succes!\n", fisier
Txt);
    else if (n == 1)
    {
        printf("\n\t\t\tNu exista client cu instructorul %s!", nume_in
str);

        fclose(t); // nu merge sters fisierul daca nu este inchis
        remove(fisierTxt);
    }
    printf("\n\t\t\tRaport clienti ce au instructor pe:");
    gets(nume_instr);
    fclose(t);
}
fclose(f);
}
}

```

```

void raport_ecran(char *fisier) // afisare client pe ecran in functie de codul
sau
{
    FILE *f;
    Client a;
    int n;
    int i;

    f = fopen(fisier, "rb");
    if (!f)
        printf("\n\t\t\tFisierul %s nu a putut fi deschis", fisier);
    else
    {
        printf("\n\t\t\tIntroduceti codul clientului dorit:");
        scanf("%d", &n);

        while (!feof(stdin))
        {
            rewind(f);
            fread(&a, sizeof(Client), 1, f);
            i = false;
            while (!feof(f) && !i)
            {
                if (n == a.codClient)
                {
                    i = true;

```

```

        printf("\n\t\t\tCod:%d Nume:%s Tip abonament:%s \n\n\t\t\tPret abonament:%d Data:%s Instructor:%s", a.codClient, a.num, a.tipAbonament, a.pretAbonament, a.data, a.instructor);
    }
    fread(&a, sizeof(Client), 1, f);
}
if (!i)
    printf("\n\t\t\tNu exista clientul cu codul:%d!", n);
printf("\n\t\t\tIntroduceti codul clientului dorit:");
scanf("%d", &n);
}
fclose(f);
}
}

```

```

void meniu() //afisare meniu
{
    int optiune = 0;
    do
    {
        printf("\n\n");
        mesajInitial();
        printf("\n\n\n\t\t\t1.Creare fisier binar");
        printf("\n\t\t\t2.Adaugare clienti");
        printf("\n\t\t\t3.Modificarea instructorului repartizat unui client");
        printf("\n\t\t\t4.Modificare abonamente grup clienti");
        printf("\n\t\t\t5.Stergerea unui client");
        printf("\n\t\t\t6.Raport fisier text clienti");
        printf("\n\t\t\t7.Raport partial fisier text clienti (in functie de instructor)");
        printf("\n\t\t\t8.Afisare client dupa codul de inregistrare");
        printf("\n\t\t\t0.Iesire program");

        printf("\n\n\n\t\t\tOptiune -->:");
        scanf("%d", &optiune);
        fflush(stdin);

        switch (optiune)
        {
            case 1:
                creare();
                system("PAUSE");
                break;
            case 2:
                adaugare(num_fisier);
                system("PAUSE");
                break;
            case 3:

```

```

        modificare_client_instructor(ume_fisier);
        system("PAUSE");
        break;
    case 4:
        modificare_abonament_clienti(ume_fisier);
        system("PAUSE");
        break;
    case 5:
        stergere_client(ume_fisier);
        system("PAUSE");
        break;
    case 6:
        raport_integral(ume_fisier);
        system("PAUSE");
        break;
    case 7:
        raport_partial(ume_fisier);
        system("PAUSE");
        break;
    case 8:
        raport_ecran(ume_fisier);
        system("PAUSE");
        break;
    case 0:
        printf("\n\n\n\t\t\tMultumesc, o zi buna!!!\n\n\n\n");
        break;
    default:
        printf("\n\n\n\t\t\tAlegerea nu este valida!!! Reincercati...");
        system("PAUSE");
        break;
    }
} while (optiune != 0);
}

void main() // functie principala
{
    meniu();
}

```