

Proiect de lecție / unitate de învățare

Unitatea de învățământ: Colegiul Național Petru Rareș Suceava

Disciplina: Informatică

Clasa: a X-a intensiv informatică

Data¹: _____

Profesor: Tiberiu Socaciu

Unitatea de învățare: Stringuri

Tema lecției: Stringuri. Transpose File

Tipul lecției: practică, de laborator

Scopul lecției: sistematizare și consolidare abilități și deprinderi practice

Competențe generale:

- a) **CG1:** Elaborarea algoritmilor de rezolvare a problemelor
- b) **CG2:** Implementarea algoritmilor într-un limbaj de programare

Competențe specifice:

- a) **CS1:** Analiza problemei în scopul identificării metodei de programare adecvate pentru rezolvarea problemei
- b) **CS2:** Aplicarea creativă a metodelor de programare pentru rezolvarea unor probleme intradisciplinare sau interdisciplinare sau a unor probleme cu aplicabilitate practică
- c) **CS3:** Utilizarea tehnicilor moderne în implementarea aplicațiilor

Obiective operaționale². La finalul lecției, elevul poate:

- a) **O1:** să demonstreze posibilitatea de a folosi corect noțiunile prezentate
- b) **O2:** să aplice noțiunile prezentate într-un caz concret
- c) **O3 :** să analizeze relațiile dintre funcțiile programului
- d) **O4:** să descrie în limbaj științific mecanismele de interdependență dintre modulele programului
- e) **O5:** să formuleze propria opțiune pentru o decizie optimă

Strategii didactice:

- a) **Mijloace de învățământ:** calculatoarele din laboratoarele de informatică, directorul FTP_Elevi -> Materiale_Profesorii prezent pe toate cele 30 stații de lucru, tableta grafică precum și resursele incluse în prezentul plan de lecție³ respectiv:

Link pentru enunțul și tema lecției	TBC ⁴
Link pentru mind map-ul lecției ⁵	TBC ⁶
Link pentru produsul final al lecției ⁷	TBC ⁸
Link și cod QR pentru activitatea independentă ⁹ și pentru tema acasă a elevilor	TBC ¹⁰
Generator de numere aleatoare	https://g.co/kgs/s3eGCQ4

- b) **Metode de învățământ:** expunerea, algoritimizarea, demonstrația, modelarea, problematizarea, conversația euristică, descoperirea sau brainstormingul.
- c) **Forme de activitate:** individuală, frontală, lucrul pe grupe pentru orele de laborator, lucrul pe grupe mici, lucrul în perechi.
- d) **Metode de evaluare:** evaluare orală, evaluare scrisă, observarea sistematică a elevilor sau prin teste și/sau fișe de evaluare.

Bibliografie:

- a) cursul *Make Technology your Friend*
- b) site-ul LeetCode, www.leetcode.ro
- c) Eusebiu Toader, Tiberiu Socaciu, Pregătirea interviului de angajare ca programator C++. 33 teme de interviu rezolvate, MatrixRom, 2023,

¹ Se va completa, după caz

² rezultă din derivarea competențelor specifice

³ link-ul lecției va fi disponibil elevilor și înafara orelor de curs pentru ca aceștia să poată accesa oricând materialele folosite

⁴ Se va completa ulterior

⁵ plan de lecție alternativ, necesar pentru susținerea activităților A și B din ”dirijarea învățării”

⁶ Se va completa ulterior

⁷ activitatea C

⁸ Se va completa ulterior

⁹ în clasă

¹⁰ Se va completa ulterior

<https://www.matrixrom.ro/produs/pregatirea-interviului-de-angajare-ca-programator-c-33-teme-de-interviu-rezolvate/>.

Succesiunea activităților de instruire

Nr. crt.	Evenimentul didactic	Durata (min.)	Obiective operaționale	Activitatea de instruire		Strategia didactică			
				Activitatea profesorului	Activitatea elevilor	Mijloace de învățământ	Metode și procedee	Forme de organizare	Forme de evaluare
1.	Moment organizatoric Prezentarea temei abordate și a obiectivelor lecției Captarea atenției	2		Salutul, efectuarea prezenței, verificarea funcționării rețelei de calculatoare, a FTP-ului. Prezentarea temei și a obiectivelor lecției.		AdServio	Conversația euristică	Frontal	Observarea sistematică a elevilor
2.	Dirijarea învățării	10	O1	Activitatea A	Notează lecția pe caiet sau calculator, adresează întrebări	Videoproiector	Învățare activă, prin cooperare	Frontal	
		10	O2	Activitatea B	Notează lecția pe caiet sau calculator, adresează întrebări	Videoproiector	Învățare activă, prin cooperare	Individual Frontal	Observarea sistematică a elevilor
		10	O3	Activitatea C	Notează lecția pe caiet sau calculator, adresează întrebări	Videoproiector	Învățare activă, prin cooperare	Frontal	
3	Transferul cunoștințelor/ consolidarea cunoștințelor.	10	O4	Problematizare legată de relația dintre funcțiile programului. Se invită un elev la tablă pentru a realiza un modul necesar rezolvării problemei	Implementează funcția pe calculator	Rețeaua de calculatoare Videoproiector	Algoritmizarea	Individual Frontal	Orala
4	Fixarea cunoștințelor și asigurarea feedback-ului	7	O5				Problematizare a	Frontal	Scrisa
5	Tema pentru acasă	1	O1, O2, O3, O4, O5	Solicită elevilor să completeze ca temă pentru acasă o problema de pe LeetCode, aleasa aleator, împreună Generează sarcina de lucru pentru acasă în Classroom	Notează tema de casă pe caiet sau calculator	Generator de numere aleatoare Google Classroom			Scrisa

Enunt

Având în vedere un fișier text.txt, transpune conținutul său.

Puteți presupune că fiecare rând are același număr de coloane și fiecare câmp este separat de caracterul ' '.

Exemplu:

Dacă fișierul.txt are următorul conținut:

nume de vârstă

Alice 21

Ryan 30

Rezultatul următoarelor:

numele Alice Ryan

vârsta de 21 30 de ani

Rezolvare. Solutia 1.

```
// varianta 1:
// citim linie cu linie fisierul
// pentru fiecare linie desfacem in cuvinte
// si completam o matrice de cuvinte
// apoi scriem matricea in fiser prin parcurgere
// pe coloane, apoi pe linie

# define MAXLINII      100
# define MAXCOLOANE    100
# define MAXLUNGCUVANT 20

# include <iostream>
# include <fstream>
using namespace std;
# include <string.h>

char cuvinte[MAXLINII][MAXCOLOANE][MAXLUNGCUVANT + 1];
// pentru NULL de la capat
int m = 0, n = 0;

int proceseaza_cuvntele_din_linie_si_pune_le_in_matrice(int numarul_de_linie,
char* linia_din_fisier)
{
char* pointer_catre_token = strtok(linia_din_fisier, " ");
int n_ul_liniei = 0;
while (pointer_catre_token != NULL)
{
if (strlen(pointer_catre_token) > MAXLUNGCUVANT)
{
cout << "Cam lung cuvantul, nu crezi?" << endl;
return 0;
}
if (n_ul_liniei == MAXCOLOANE)
{
cout << "Cam multe coloane, nu crezi?" << endl;
return 0;
}
strcpy(cuvinte[numarul_de_linie][n_ul_liniei],
pointer_catre_token);
pointer_catre_token = strtok(NULL, " ");
n_ul_liniei++;
}
return n_ul_liniei;
}
```

```

}

void citeste()
{
    ifstream fin("file.txt");
    char linie[MAXCOLOANE * MAXLUNGCUVANT + MAXCOLOANE - 1 + 1];
    fin.getline(linie, MAXCOLOANE * MAXLUNGCUVANT + MAXCOLOANE
        - 1 + 1);
    n = proceseaza_cuvntele_din_linie_si_pune_le_in_matrice(0,
        linie);
    m = 1;
    while (fin.getline(linie, MAXCOLOANE * MAXLUNGCUVANT +
        MAXCOLOANE - 1 + 1))
    {
        if (m == MAXLINII)
        {
            cout << "Cam multe linii, nu crezi?" << endl;
            return;
        }
        int n_ul_liniei =
            proceseaza_cuvntele_din_linie_si_pune_le_in_matrice(
                m, linie);
        if (n_ul_liniei != n)
        {
            cout << "Fisierul nu e o matrice de cuvinte" << endl;
            return;
        }
        m++;
    }
}

void scrie()
{
    ofstream fout ("file2.txt");
    for (int j = 0; j < n; j++)
    {
        for (int i = 0; i < m; i++)
            fout << cuvinte[i][j] << " ";
        fout << endl;
    }
}

int main()
{
    citeste();
    scrie();
}

```

Rezolvarea 2

```

# include <iostream> // cin si cout ca streamuri
# include <fstream> // file.txt ca stream
# include <sstream> // linia care e un string ca stream
# include <string.h> // pentru strcpy
using namespace std;

char matrice [20][30][10];
int m, n;

void citeste_matricea_din_fisier()
{
    char cuvant[10];

```

```

string linie;
m = 0;
ifstream fin("file.txt");
while (getline(fin, linie))
{
    n = 0;
    istringstream linia_ca_stream(linie);
    while (linia_ca_stream >> cuvant)
    {
        strcpy(matrice[m][n], cuvant);
        // in loc de: for (i...) matrice[m][n][i] = cuvant[i]
        n = n + 1; // il fac dupa, ca sa pot folosi pe n
    }
    m = m + 1;
}

// void = functie procedurala
void afiseaza_transpusa()
{
    for (int j = 0; j < n; j++)
    {
        for (int i = 0; i < m; i++)
            cout << matrice[i][j] << " ";
        cout << endl;
    }
}

int main()
{
    citeste_matricea_din_fisier();
    afiseaza_transpusa();
    return 0;
}

```