

Liceul Teoretic „Decebal” Constanța

TEORIE PENTRU BACALAUREAT, PROBA INFORMATICĂ

LUCRARE PENTRU ATESTAREA
COMPETENȚELOR PROFESIONALE

Elev: Donisan Enya

Clasa: a XII-a A

Profesor coordonator: Dragomir Elena

2020 – 2021

Cuprins

1. Motivul alegerii temei	pag. 3
2. Structura aplicației	pag. 3-7
3. Detalii tehnice de implementare	pag. 8-9
4. Limbajul HTML	pag. 10-13
5. Resurse hard și soft necesare	pag. 13-14

1. Motivul alegerii temei

Am ales această temă deoarece doresc să mă specializez în programare, mai ales în limbajul de programare C++. Acest site conține toată teoria de bază pentru acest limbaj, aceste informații fiind folositoare și pentru proba de Informatică de la Bacalaureat.

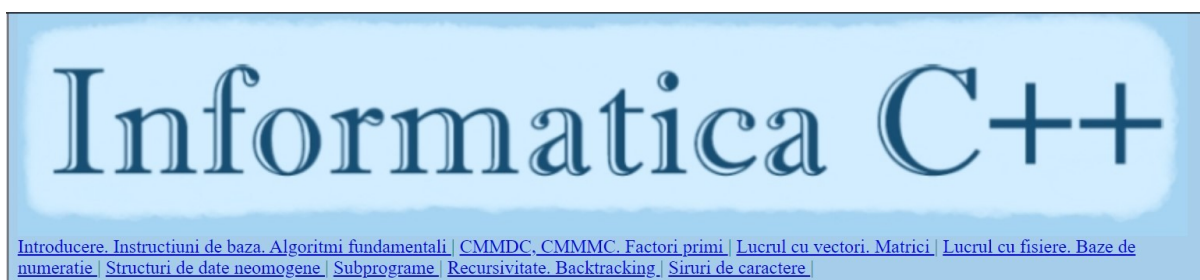
2. Structura aplicației

Programul este simplu, concentrat și este alcătuit din trei segmente:

- Primul segment – format din logo-ul site-ului, data din prezent, un mesaj de întâmpinare ce se modifică în funcție de momentul zilei și link-urile ce vor fi afișate în cel de-al treilea segment.



- Al doilea segment – conține o imagine cu banner-ul site-ului și afișează link-urile ce sunt afișate și în cel de-al treilea segment



- Al treilea segment – afișează link-urile selectate din primul segment.

În următoarele imagini sunt prezentate conținuturile link-urilor ce pot fi apăsate din primul segment:

- “Introducere. Instrucțiuni de baza. Algoritmi fundamentali” – conține informațiile necesare pentru a crea un program în C++.

Teorie pentru Bacalaureat, proba Informatica

Bună ziua!
24-4-2021

Informatica C++

[Introducere. Instrucțiuni de baza. Algoritmi fundamentali](#) |
 [CMMDC, CMMMC. Factori primi](#) |
 [Lucrul cu vectori. Matrici](#) |
 [Lucrul cu fișiere. Baze de numeratie](#) |
 [Structuri de date neomogene](#) |
 [Subprograme](#) |
 [Recursivitate. Backtracking](#) |
 [Șiruri de caractere](#)

[Introducere. Instrucțiuni de baza. Algoritmi fundamentali](#) |
 [CMMDC, CMMMC. Factori primi](#) |
 [Lucrul cu vectori. Matrici](#) |
 [Lucrul cu fișiere. Baze de numeratie](#) |
 [Structuri de date neomogene](#) |
 [Subprograme](#) |
 [Recursivitate. Backtracking](#) |
 [Șiruri de caractere](#)

Introducere

☞ **Tipuri de date**

- intregi: int, long, long long;
- naturale: unsigned, unsigned int, unsigned long;
- reale: float, double;
- caracter: char;
- A,F: int (1=A, 0=F) → adevarat/fals

☞ **Operatori aritmetici: +, -, *, /, %;**

Obs: 1) La impartire, daca ambele numere sunt intregi se realizeaza

- “CMMDC, CMMMC. Factori primi” – conține teoria necesară pentru algoritmi de CMMDC, CMMMC și descompunerea în factori primi.

Teorie pentru Bacalaureat, proba Informatica

Bună ziua!
24-4-2021

Informatica C++

[Introducere. Instrucțiuni de baza. Algoritmi fundamentali](#) |
 [CMMDC, CMMMC. Factori primi](#) |
 [Lucrul cu vectori. Matrici](#) |
 [Lucrul cu fișiere. Baze de numeratie](#) |
 [Structuri de date neomogene](#) |
 [Subprograme](#) |
 [Recursivitate. Backtracking](#) |
 [Șiruri de caractere](#)

[Introducere. Instrucțiuni de baza. Algoritmi fundamentali](#) |
 [CMMDC, CMMMC. Factori primi](#) |
 [Lucrul cu vectori. Matrici](#) |
 [Lucrul cu fișiere. Baze de numeratie](#) |
 [Structuri de date neomogene](#) |
 [Subprograme](#) |
 [Recursivitate. Backtracking](#) |
 [Șiruri de caractere](#)

C.M.M.D.C.; C.M.M.M.C.; Descompunerea in factori primi

1. C.M.M.D.C.:

a) Cu diferite repetate:

```

if(a==0 || b==0)
    cmmdc=a+b;
else
{
    while(a!=b)
        if(a>b)
            a=a-b;
        else
            b=b-a;
    cmmdc=b;
}

```

b) Algoritmul lui Euclid:

```


if(a==0 || b==0)
    cmmdc=a+b;
else
{
    r=a%b;
    while(r!=0)
    {
        a=b;
        b=r;
        r=a%b;
    }
    cmmdc=b;
}

```

Obs.: In urma aplicarii algoritmilor de cmmdc, valorile variabilelor se

- “Lucrul cu vectori. Matrici” – conține teoria necesară pentru a putea lucra cu vectori și matrici.

Teorie pentru
Bacalaureat, proba
Informatica



Bună ziua!
24-4-2021

Informatica C++

[Introducere](#) | [Instrucțiuni de baza](#) | [Algoritmi fundamentali](#) | [CMMDC, CMMMC, Factori primi](#) | [Lucrul cu vectori, Matrici](#) | [Lucrul cu fișiere](#) | [Baze de numeratie](#) | [Structuri de date neomogene](#) | [Subprograme](#) | [Recursivitate](#) | [Backtracking](#) | [Șiruri de caractere](#)

[Introducere](#) | [Instrucțiuni de baza](#) | [Algoritmi fundamentali](#) | [CMMDC, CMMMC, Factori primi](#) | [Lucrul cu vectori, Matrici](#) | [Lucrul cu fișiere](#) | [Baze de numeratie](#) | [Structuri de date neomogene](#) | [Subprograme](#) | [Recursivitate](#) | [Backtracking](#) | [Șiruri de caractere](#)

Lucrul cu vectori

1. Inserarea unui element pe o pozitie data

```

n++;
for(i=n;i>=poz+1;i--)
    v[i]=v[i-1];
v[poz]=x;

```

2. Eliminarea unui element pe o pozitie data

```

for(i=poz;i<n;i++)
    v[i]=v[i+1];
n--;

```

3. Permutarea circulara cu o pozitie spre dreapta

```

aux=v[n];
for(i=n;i>1;i--)
    v[i]=v[i-1];
v[1]=aux;

```

- “Lucrul cu fișiere. Baze de numeratie” – conține teoria necesară pentru a putea lucra cu fișiere de intrare și ieșire, dar și algoritmi pentru schimbarea bazei de numeratie.

Teorie pentru
Bacalaureat, proba
Informatica



Bună ziua!
24-4-2021

Informatica C++

[Introducere](#) | [Instrucțiuni de baza](#) | [Algoritmi fundamentali](#) | [CMMDC, CMMMC, Factori primi](#) | [Lucrul cu vectori, Matrici](#) | [Lucrul cu fișiere](#) | [Baze de numeratie](#) | [Structuri de date neomogene](#) | [Subprograme](#) | [Recursivitate](#) | [Backtracking](#) | [Șiruri de caractere](#)

[Introducere](#) | [Instrucțiuni de baza](#) | [Algoritmi fundamentali](#) | [CMMDC, CMMMC, Factori primi](#) | [Lucrul cu vectori, Matrici](#) | [Lucrul cu fișiere](#) | [Baze de numeratie](#) | [Structuri de date neomogene](#) | [Subprograme](#) | [Recursivitate](#) | [Backtracking](#) | [Șiruri de caractere](#)

Lucrul cu fișiere


```
#include <fstream>
```

☞ Deschiderea fișierelor:

- `ifstream varfis("nume.in");`
Ex: `ifstream fin("bac.in");`
- `ofstream varfis("nume.out");`
Ex: `ofstream fout("bac.out");`

- “Structuri de date neomogene” – conține teoria necesară pentru a putea programa structuri de date neomogene.

Teorie pentru Bacalaureat, proba Informatica



Bună ziua!
24-4-2021

Informatica C++

[Introducere](#) | [Instrucțiuni de baza](#) | [Algoritmi fundamentali](#) | [CMMDC](#), [CMMMC](#) | [Factori primi](#) | [Lucrul cu vectori](#), [Matrici](#) | [Lucrul cu fișiere](#), [Baze de numeratie](#) | [Structuri de date neomogene](#) | [Subprograme](#) | [Recursivitate](#), [Backtracking](#) | [Șiruri de caractere](#)

[Introducere](#) | [Instrucțiuni de baza](#) | [Algoritmi fundamentali](#) | [CMMDC](#), [CMMMC](#) | [Factori primi](#) | [Lucrul cu vectori](#), [Matrici](#) | [Lucrul cu fișiere](#), [Baze de numeratie](#) | [Structuri de date neomogene](#) | [Subprograme](#) | [Recursivitate](#), [Backtracking](#) | [Șiruri de caractere](#)

Structuri de date neomogene

❖ Se folosesc atunci cand dorim sa memoram intr-o variabila mai multe tipuri diferite

❖ **Definirea structurii:**

```
struct [denumire]
{
    tip1 camp1_1, camp1_2,...;
    tip2 camp2_1, camp2_2,...; }[variabile];
```

❖ **Declaraarea variabilelor:**

- > Varianta 1: Odata cu definirea structurii
- > Varianta 2: [struct] denumire var1, var2,...;

Ex.: struct produs a,b,c;

❖ **Operatii cu structuri:**

- > **atribuire:** p1=p2 → toate caracterele din p2 se copiaza in p1
- > **accesul la un camp se face prin intermediul operatorului punct (':')**: variabila.camp


Ex.: Pentru citirea datelor produsului p1:

```
cin.get(p1.den,25)
cin.get(i);
```

```
struct produs
{
    char den[25], um[25];
    float pret, gr;
    unsigned cant, zi, an;
}; p1, p2;
```

- “Subprograme” – conține teoria necesară pentru a putea crea subprograme.

Teorie pentru Bacalaureat, proba Informatica



Bună ziua!
24-4-2021

Informatica C++

[Introducere](#) | [Instrucțiuni de baza](#) | [Algoritmi fundamentali](#) | [CMMDC](#), [CMMMC](#) | [Factori primi](#) | [Lucrul cu vectori](#), [Matrici](#) | [Lucrul cu fișiere](#), [Baze de numeratie](#) | [Structuri de date neomogene](#) | [Subprograme](#) | [Recursivitate](#), [Backtracking](#) | [Șiruri de caractere](#)

[Introducere](#) | [Instrucțiuni de baza](#) | [Algoritmi fundamentali](#) | [CMMDC](#), [CMMMC](#) | [Factori primi](#) | [Lucrul cu vectori](#), [Matrici](#) | [Lucrul cu fișiere](#), [Baze de numeratie](#) | [Structuri de date neomogene](#) | [Subprograme](#) | [Recursivitate](#), [Backtracking](#) | [Șiruri de caractere](#)

Subprograme

❖ Subprogramele sunt secvențe de instrucțiuni care rezolvă o parte dintr-o problemă

❖ Se da un vector cu n elemente nr.nat. Sa se calculeze suma oglinzitelor numerelor prime si sa se afiseze vectorul ordonat

❖ **Subprograme:**

❖ Citirea vectorului	2
❖ Verificarea numarului prim	1
❖ Formare oglindit	1
❖ Suma oglinzitelor	1
❖ Ordonare vector	2
❖ Afisare vector ordonat	2

❖ Există două tipuri de subprograme:

1. Care efectuează calcule si RETURNEAZA o valoare
2. Care execută o operație si NU RETURNEAZA o valoare (poate FURNIZA una sau mai multe valori)

❖ **Structura subprogramelor:**

➔ se folosește pentru a FURNIZA ultima valoare a parametrului către subprogramul apelat

--forma generala: tip [&p1,tip [&p2,tip [&p3,...

--ex.: int a,int b,int &c

--se declara parametrii de intrare a si b de tip int, iar parametrul c este si de iesire

Obs.: Tablourile (vectorii si matricile) se transmit automat prin referinta.

❖ **return expresie**—se foloseste in subprogramele care returneaza o valoare pentru a RETURNA valoarea dorita

❖ Apelul subprogramelor se face prin nume_program ([lista parametrii actuali]) efectiv;

❖ **Obs.:** Daca subprogramul returneaza o valoare aceasta se apeleaza in cadrul altei expresii, iar daca nu returneaza o valoare se apeleaza ca o comanda.

- **Lista parametrii actuali:** lista de variabile sau valori care se vor transmite catre parametrii formali;

- “Recursivitate. Backtracking” – conține teoria necesară pentru a putea rezolva probleme de recursivitate și backtracking.

Teorie pentru Bacalaureat, proba Informatica



Bună ziua!
24-4-2021

Informatica C++

[Introducere](#) | [Instrucțiuni de baza](#) | [Algoritmi fundamentali](#) | [CMMDC, CMMMC](#) | [Factori primi](#) | [Lucrul cu vectori](#) | [Matrici](#) | [Lucrul cu fișiere](#) | [Baze de numeratie](#) | [Structuri de date neomogene](#) | [Subprograme](#) | [Recursivitate, Backtracking](#) | [Siruri de caractere](#)

[Introducere](#) | [Instrucțiuni de baza](#) | [Algoritmi fundamentali](#) | [CMMDC, CMMMC](#) | [Factori primi](#) | [Lucrul cu vectori](#) | [Matrici](#) | [Lucrul cu fișiere](#) | [Baze de numeratie](#) | [Structuri de date neomogene](#) | [Subprograme](#) | [Recursivitate, Backtracking](#) | [Siruri de caractere](#)

Recursivitate

Exercițiul #497

Subprogramul r este definit mai jos.

```
int f(int a, int b)
{ if (a==b) return 0;
  if (b!=0) return a+b;
  return f(a+1,b-1);
}
```

Scrieți valorile $f(5,5)$ și $f(16,12)$ în această ordine, separate prin exact un spațiu.

Subiect Bacalaureat 2014, sesiunea iunie/iulie

$f(5,5)$
 $a=5$
 $b=5$
 $r=0$
 $f(10, 21)$
 $a=10$
 $b=21$
 $r: f(12, 19)$


$f(12, 19)$
 $a=12$
 $b=19$
 $r: f(14, 15)$

$f(14, 15)$
 $a=14$
 $b=15$
 $r: f(16, 12)$

$f(16, 12)$
 $a=16$
 $b=12$
 $r: 16+12=28$

- “Siruri de caractere” – conține teoria necesară pentru a putea lucra cu șiruri de caractere.

Teorie pentru Bacalaureat, proba Informatica



Bună ziua!
24-4-2021

Informatica C++

[Introducere](#) | [Instrucțiuni de baza](#) | [Algoritmi fundamentali](#) | [CMMDC, CMMMC](#) | [Factori primi](#) | [Lucrul cu vectori](#) | [Matrici](#) | [Lucrul cu fișiere](#) | [Baze de numeratie](#) | [Structuri de date neomogene](#) | [Subprograme](#) | [Recursivitate, Backtracking](#) | [Siruri de caractere](#)

[Introducere](#) | [Instrucțiuni de baza](#) | [Algoritmi fundamentali](#) | [CMMDC, CMMMC](#) | [Factori primi](#) | [Lucrul cu vectori](#) | [Matrici](#) | [Lucrul cu fișiere](#) | [Baze de numeratie](#) | [Structuri de date neomogene](#) | [Subprograme](#) | [Recursivitate, Backtracking](#) | [Siruri de caractere](#)

Siruri de caractere

Declarare: char var[dim], *pointer;
Ex.: char s[100], *p;

Obs.:

- Sirurile de caractere se termina automat cu caracterul NULL. Pentru declarare se adauga o unitate la dimensiunea maxima prevazuta in problema.
- Valorile in siruri se memoreaza incepand cu pozitia 0.
s[0] → primul caracter din sir
- Memorarea caracterelor se face dupa codul ASCII.

'A' = 65

+32

'a' = 97

↑

-32

'a' - 'A' = 32

- Datorita codurilor ASCII, literele mici sunt considerate mai

3. Detalii tehnice de implementare

Pentru edifiatarea paginii WEB am folosit NotePad++. Sunt mai multe programe cu care se pot crea pagini web, chiar fără a cunoaște limbajul HTML, cum sunt Microsoft FrontPage sau Macromedia Dreamweaver.

Editoarele de text sunt programele de bază, și cele mai folosite, pentru editarea paginii web.

Avantajul folosirii lor este simplitatea, pentru crearea paginii web sunt necesare câteva din codurile invizibile ale limbajului HTML, astfel se creează documente rapid și ușor, în plus ocupă foarte puțin spațiu și resurse de memorie, acestea însă necesită cunoașterea limbajului HTML.

Introducerea imaginilor

Imaginile sunt foarte importante într-o pagină web. De aceea este recomandat să fie folosite în mod corect. Inserarea lor se face cu ajutorul tag-ului.

```
<IMG SRC="Icon.jpg"height="90" width="150">
```

"src" este prescurtarea pentru "source" (sursa). Acest atribut se folosește pentru a indica locația fotografiei.

"height" reprezintă lungimea imaginii, iar "width" reprezintă lățimea imaginii, iar mărimile acestora sunt măsurate în pixeli.

Introducerea textului

```
<FONT face="ARIAL" SIZE="6" COLOR="178ADF">
```

Marimea textului poate fi modificata cu elementul FONT si atributul SIZE. Atributul SIZE poate prelua valorile numerice de la 1 la

7 si pe langa acestea pot fi folosite, ca valoare relativa, semnele "+" sau "-". Textul normal (daca nu este specificat acest atribut) are valoarea 3.

Crearea legăturilor

Pentru ca prin simplul click a unui text sa deschida o alta pagina, se foloseste tag-ul <a href>.

```
<a href="2.html" TARGET="C4"> CMMDC, CMMMC. Factori  
primi </a>|
```

Când se apasă pe textul “Mașini de vânzare”, în cel de al treilea segment se va afișa lista creată în fișierul “1.html”.

Afișarea datei din prezent si a mesajului de întâmpinare din primul segment

```
<SCRIPT>  
var azi=new Date();  
if(azi.getHours()<12) s="Buna dimineata!<BR>";  
if(azi.getHours()>=12&& azi.getHours()<18) s= "Buna ziua!<BR>";  
if(azi.getHours()>=18) s="Buna seara!<BR>";  
document.write(s);  
document.write(azi.getDate()+". "+(azi.getMonth()  
+1)+". "+azi.getFullYear());  
</SCRIPT>
```

4. Limbajul HTML

HTML (Hypertext Markup Language) este un limbaj creat în scopul de a descrie, în mod text, formatul paginilor Web; fisierele create în acest limbaj vor fi interpretate de navigatoare, care vor afișa paginile în forma dorită (cu texte formate, liste, tabele, formule, imagini, hiperlegături, obiecte multimedia etc.). HTML a apărut ca o aplicație ISO standard (apartine standardului SGML - Standard Generalized Markup Language, specializat pentru hipertext și adaptat la Web).

Asa cum se poate deduce din numele limbajului, HTML descrie caracteristicile de format ale elementelor incluse prin procedee de marcare. Acestea pot fi asemănate intuitiv cu marcasele folosite în tipografie pentru a indica scrierea unui text cu un anumit tip de caractere. Fiecare element va fi introdus între două marcase ("tags", în limba engleză) - de început și sfârșit - (uzual) de forma `<marcaj> ... </marcaj>`. Caracterele speciale de delimitare a marcajelor "`<`", "`>`" permit deosebirea acestora de textul propriu-zis. De exemplu, pentru textele aldine (îngrosate), marcajul de început este `` iar de sfârșit - ``.

În informatică, limbajele de marcare sunt foarte convenabile fiindcă comenzile lor pot fi interpretate simplu. LaTeX-ul, de exemplu, este tot un limbaj de marcare; prin interpretarea fișierelor .tex descrise în acest limbaj se va genera formatul dorit al documentelor pe diverse tipuri de sisteme de calcul (în cazul, LaTeX-ului, se obține uzual format PostScript sau PDF). În schimb, procesoarele de documente uzuale nu au un limbaj de marcare standardizat, care să ofere compatibilitate între diverse tipuri de calculatoare și sisteme de operare. Astfel, se poate spune că avantajele aplicării limbajelor de marcare constau în portabilitate și flexibilitate: fișierele create cu ajutorul lor pot fi transferate pe orice tip de sistem, unde vor fi interpretate cu ajutorul unor programe specifice.

De fapt, procesoarele de texte uzuale folosesc adesea procedee de marcare pentru formătări (de exemplu, formătărilor de tip caracter din

Word); în acest caz însă, caracterele de control introduse sunt ascunse iar rezultatul editării este direct vizibil ("What You See Is What You Get"). În schimb, în limbajele de marcare - inclusiv HTML - marcajele sunt introduse în text, astfel încât acestea sunt exclusiv succesiuni de caractere (litere, cifre, caractere speciale) - fișiere de tip text.

Referitor la legătura dintre procesoarele de documente uzuale și limbajul HTML, mai trebuie menționat faptul că ultimele versiuni ale editoarelor de documente oferă facilități de salvare în format HTML - de exemplu, Word, începând cu versiunea Microsoft Office '97. Mai mult, toate produsele incluse în această gamă dedicată biroticii (MS Office) oferă compatibilitate cu formatul HTML.

Procesele de standardizare și de includere a comenzilor de marcare în fișierele HTML permit navigatoarelor să citească și să formateze paginile Web, lucru foarte important în condițiile în care ele conțin nu numai texte alb-negru, ci și culori, imagini, hiperlegături, diverse obiecte. Practic, marcajele HTML asigură controlul asupra modului de afișare a obiectelor corespunzătoare în cadrul programelor de vizualizare a documentelor HTML - navigatoarele.

Limbajul HTML a evoluat în versiuni succesive, odată cu evoluția protocolului HTTP și a programelor de navigare. Astfel, HTML 1.0 era compatibil cu Mosaic, primul program de navigare, dar după apariția unor navigatoare noi, a fost necesară introducerea unui standard oficial Internet pentru construirea paginilor (HTML 2.0) și extinderea sa cu noi facilități: formule matematice, tabele, moduri avansate de descriere a organizării paginilor (începând cu HTML 3.0).

Standardizarea oficială a limbajului HTML a fost realizată de consorțiul WWW și dezvoltată de diverși producători de soft (unii dintre aceștia urmăresc chiar promovarea navigatoarelor proprii prin introducerea unor particularități în formatele oficiale).

Paginile HTML se pot crea cu orice editor de texte de către utilizatorii care cunosc limbajul HTML sau, mai simplu, se pot utiliza editoare speciale, în care obiectele se introduc interactiv iar codul HTML se generează automat. Având în vedere că și în acest caz este utilă cunoașterea marcajelor generate pentru corectarea eventualelor

erori (mai ales în cazul link-urilor), vom prezenta în continuare entitățile care se pot introduce în paginile HTML și marcajele caracteristice acestora:

Elementele limbajului HTML

Toate obiectele HTML sunt introduse între marcaje care le definesc; majoritatea acestora sunt de forma `<tip_obiect>` (la început) și `</tip_obiect>` (la sfârșit). Tipul standard al obiectului poate fi specificat cu majuscule sau minuscule; totuși, se recomandă utilizarea majusculilor fiindcă astfel marcajele ies în evidență.

Majoritatea navigatoarelor permit vizualizarea paginii curente în formatul sursă HTML (forma pe care o interpretează pentru afisarea paginii). La interpretare, programele de navigare ignoră spațiile și `<Enter>`-urile, aplicând formatarea specificată.

Există însă și marcaje cu parametri; acestea au forma `<tip_obiect param1=valoare1 param2=valoare2 ...>`. De exemplu, obiectele de tip imagine sunt introduse cu delimitatorul ``, care are diverși parametri. Pentru definirea hiperlegăturilor se folosește marcajul `<A>`, care are de asemenea parametri proprii.

Codurile de marcare HTML pot fi clasificate în următoarele categorii:

1. marcaje de bază - cele care delimitează pagina / documentul HTML, titlul acesteia și corpul paginii;
2. marcaje pentru structurarea documentului - care permit introducerea de subtitluri, paragrafe, linii de delimitare;
3. marcaje pentru formatarea textului și crearea listelor;
4. marcaje pentru crearea hiperlegăturilor (hyperlinks);
5. marcaje pentru introducerea de obiecte - tabele, formule, imagini sau obiecte multimedia preluate din fișiere, formulare.

Vom descrie în paragrafele următoare, elementele caracteristice fiecăreia din aceste categorii.

Pentru structurarea și organizarea informațiilor din paginile web se pot utiliza frame-uri (marcajul `<FRAME>`), prin care la un moment dat se afișează mai multe ferestre conținând fiecare câte o pagină.

Introducerea si gestiunea frame-urilor se realizează foarte convenabil folosind editoarele HTML.

Mai mentionăm faptul că în ultimele versiuni ale limbajului HTML si ale browser-elor s-a introdus posibilitatea integrării, respectiv lansării în executie prin navigator, a unor aplicatii. Acestea sunt scrise în limbajul Java, un limbaj cu caracteristici distribuite si obiectuale, adaptat programării în Web; ele se numesc "applet"-uri si se introduc în sursele HTML cu marcajul <APP> sau <APPLET> . Ultimele versiuni de editoare HTML permit introducerea interactivă a applet-urilor Java.

5. Resurse hard și soft necesare

Site-ul a fost realizat cu ajutorul a mai multe programe precum NotePad++ și Adobe Photoshop CC 2017. Cerințele de sistem ale programului nu sunt exagerate.

Resurse minime :

1) Hardware :

- Procesor (CPU) : 500 MHz +
- Memorie (RAM): 64 MB
- Spațiu liber pe hard-disk : 30 MB
- Monitor :1024*768
- Un browser oarecare: Internet Explorer, Mozilla Firefox,

Google Chrome

2) Software :

- Sistem de operare : Windows 98/XP

Resurse recomandate:

1) Hardware :

- Procesor (CPU) : 1000 MHz +
- Memorie (RAM): 128 MB
- Spațiu liber pe hard-disk : 30 MB

- Monitor : VGA 32-bit color (1024x768)
- Un browser oarecare: Internet Explorer, Mozilla Firefox, Google Chrome, Microsoft Edge

2) Software :

- Sistem de operare : Windows XP