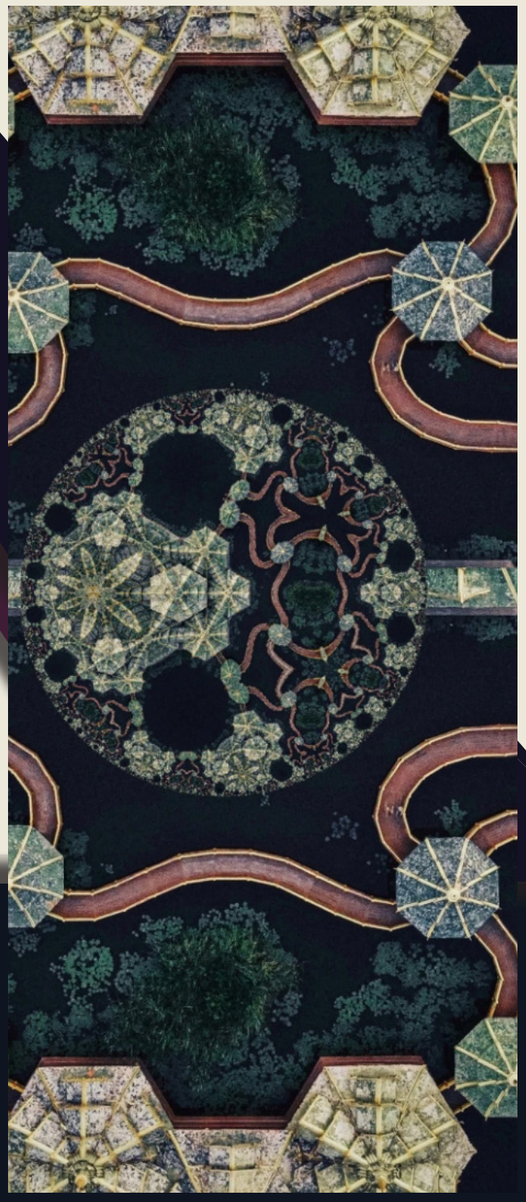
TEORIA FRACTALILOR IN PROGRAMARE



Autori: Brisan Sebastian, Mincu Mircea, Popescu Miruna

Atestat Informatica 2022-2023
Colegiul National "Scoala Centrala" Bucuresti

"Universul este un fractal. Orice semnătură energetică pe care o purtăm se va repeta la infinit, iar şi iar... până când vom schimba acea vibraţie."

- Paige Bartholomew

Cuprins

Suport tehnic

1.1) Introducere
1.2) Fractali - Prezentarea si explicarea temei alese pentru proiect

Coduri sursa

2.1) JavaScript
2.2) CSS
2.3) HTML
2.4) Galerie

C/C++

3.1) Recursivitate
3.2) Grile si Probleme
3.3) Bibliografie

prefata.

Am hotarat sa abordam capitolul "Fractali" intrucat a reprezentat un subiect care ne fascina de foarte mult timp, dar pe care nu-l intelegeam complet.

Proiectul a fost realizat in doua etape. Prima etapa a reprezentat cautarea de informatii despre fractali, fiind puse apoi in word-uri pentru a le integra mai tarziu in proiect, si rezolvarea efectiva a problemelor si a grilelor implicate in proiect.

In a doua parte au fost incadrate intr-un site informatiile si problemele realizate la prima parte.
Sarciniile nu au fost impartite, ci mai degraba preluate.

In cadrul echipei noastre,task-urile au fost organizate pe o anumita structura, iar apoi fiecare membru, in functie de preferintele sale, si-a adoptat o sarcina.

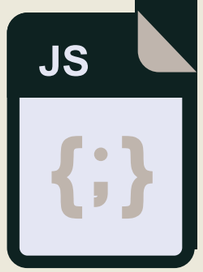
JavaScript (JS)

JavaScript, adesea abreviat în JS, este un limbaj de programare care este una dintre tehnologiile de bază ale World Wide Web, alaturi de HTML si CSS. Incepand cu 2022, 98% dintre site-uri web folosesc JavaScript.

Are interfețe de programare a aplicațiilor (API) pentru lucrul cu text, date, expresii regulate, structuri de date standard și Modelul obiect document (DOM). În practică, browserul web sau alt sistem de rulare furnizează API-uri JavaScript pentru I/O.

Motoarele JavaScript au fost utilizate inițial doar în browserele web, dar acum sunt componente de bază ale unor servere și ale unei varietăți de aplicații. Cel mai popular sistem de rulare pentru această utilizare este Node.js.

Deși Java și JavaScript sunt similare ca nume, sintaxă și bibliotecile standard respective, cele două limbi sunt distincte și diferă foarte mult în design.



Utilizari

Crearea site-urilor web interactive

→ JavaScript face paginile web dinamice. Înainte de JavaScript, paginile web erau construite numai cu HTML și CSS. HTML și CSS sunt capabile doar să creeze pagini statice care pot fi stilate, dar nu interactive în afară de hyperlinkuri.

Dezvoltarea de Jocuri

→ Dezvoltarea de jocuri captivante
JavaScript este folosit frecvent pentru a crea jocuri în browser. Dezvoltatorii folosesc JavaScript pentru a crea puzzle-uri 2D și 3D, jocuri de rol, jocuri de curse, jocuri cu platforme și multe altele. Cele mai populare motoare de joc includ Backbone, DarlingJS și JawsJS. YouTube și Facebook.

Construirea Aplicatiilor

→Cu colecția extinsă de cadre JavaScript, dezvoltatorii pot construi eficient aplicații pentru mobil și web.

→Framework-urile sunt biblioteci de cod JavaScript pre-scris pe care dezvoltatorii le folosesc pentru funcții standard. Vă puteți gândi la un cadru JavaScript ca un plan.

HTML

HyperText Markup Language sau HTML este limbajul standard de marcare pentru documentele concepute pentru a fi afișate într-un browser web. Poate fi asistat de tehnologii precum Cascading Style Sheets (CSS) și limbaje de scripting precum JavaScript.

Browserele web primesc documente HTML de la un server web sau de la stocarea locală și redă documentele în pagini web multimedia. HTML descrie structura unei pagini web din punct de vedere semantic și inițial a inclus indicii pentru aspectul documentului.

Elementele HTML sunt elementele de bază ale paginilor HTML. Cu constructele HTML, imaginile și alte obiecte, cum ar fi formularele interactive, pot fi încorporate în pagina redată. HTML oferă un mijloc de a crea documente structurate prin denotarea semanticii structurale pentru text, cum ar fi titluri, paragrafe, liste, legături, citate și alte elemente. Elementele HTML sunt delimitate de etichete, scrise folosind paranteze unghiulare. Etichete precum <img /> și <input /> introduc direct conținut în pagină. Alte etichete, cum ar fi <p>, înconjoară și oferă informații despre textul documentului și pot include alte etichete ca subelemente.

CSS

Cascading Style Sheets (CSS) este un limbaj pentru foi de stil folosit pentru a descrie prezentarea unui document scris într-un limbaj de marcare precum HTML sau XML (inclusiv dialecte XML, cum ar fi SVG, MathML sau XHTML). CSS este o tehnologie de bază a World Wide Web, alături de HTML și JavaScript.

CSS este conceput pentru a permite separarea prezentării și a conținutului, inclusiv aspectul, culorile și fonturile. Această separare poate îmbunătăți accesibilitatea conținutului; oferă mai multă flexibilitate și control în specificarea caracteristicilor de prezentare; permiteți mai multor pagini web să partajeze formatarea prin specificarea CSS-ului relevant într-un fișier .css separat, ceea ce reduce complexitatea și repetarea conținutului structural; și permiteți ca fișierul .css să fie stocat în cache pentru a îmbunătăți viteza de încărcare a paginii între paginile care partajează fișierul și formatarea acestuia.

Separarea formatării și a conținutului face, de asemenea, fezabilă prezentarea aceleiași pagini de marcare în stiluri diferite pentru diferite metode de randare, cum ar fi pe ecran, în tipărire, prin voce (prin browser sau cititor de ecran bazat pe vorbire) și pe baza Braille. dispozitive tactile.

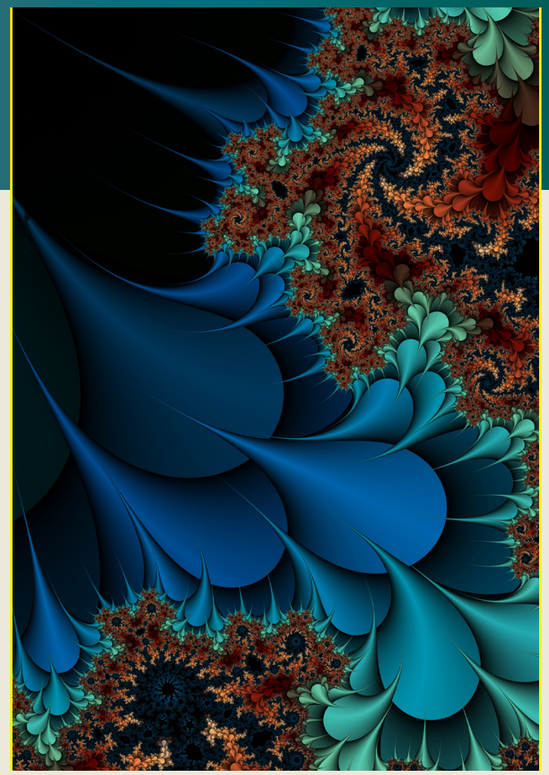
Fractali

Jonathan Coulton, versuri din „Mandelbrot Set”

"Monștri patologici! strigă matematicianul îngrozit
Fiecare dintre ei o așchie în ochiul meu
Urăsc Spațiul Peano și Curba Koch
Mă tem de Multimea lui Cantor
Triunghiul lui Sierpinski mă face să plâng
Și la un milion de mile depărtare un fluture batea din aripi
Într-o zi rece de noiembrie s-a născut un bărbat pe nume Benoit Mandelbrot”

Ce sunt fractalii?

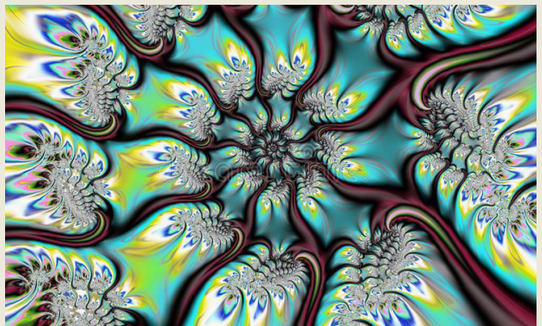
Un fractal este o formă geometrică care are o dimensiune fracțională. Mulți fractali celebri sunt auto-similari, ceea ce înseamnă că sunt alcătuiți din copii mai mici ale lor. Fractalii conțin modele la fiecare nivel de mărire și pot fi creați prin repetarea unei proceduri sau repetarea unei ecuații de nenumărate ori.

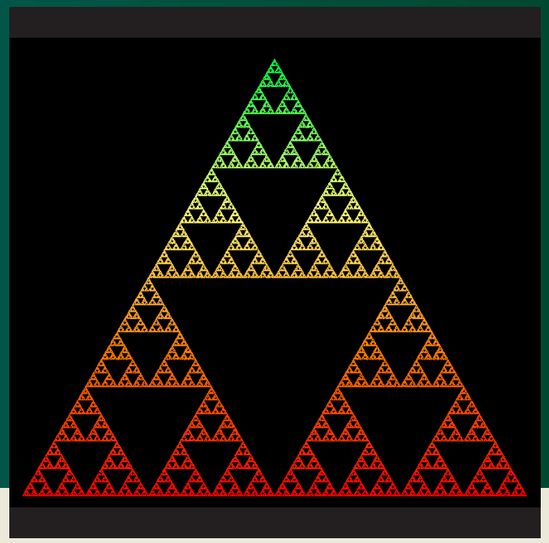


Auto-similaritate

Inițial, acestea par forme extrem de complexe – dar când te uiți mai atent, s-ar putea să observi că ambele urmează un model relativ simplu: toate părțile individuale arată exact la fel, doar mai mici. Același model se repetă iar și iar, la scari mai mici.

În matematică, numim această proprietate auto-similaritate, iar formele care o au se numesc fractali.





Triunghiul lui Sierpinski

Triunghiul Sierpinski, care este numit după matematicianul polonez Wacław Sierpiński, poate fi creat pornind cu un triunghi mare, echilateral, apoi tăind în mod repetat triunghiuri mai mici din centrul său.

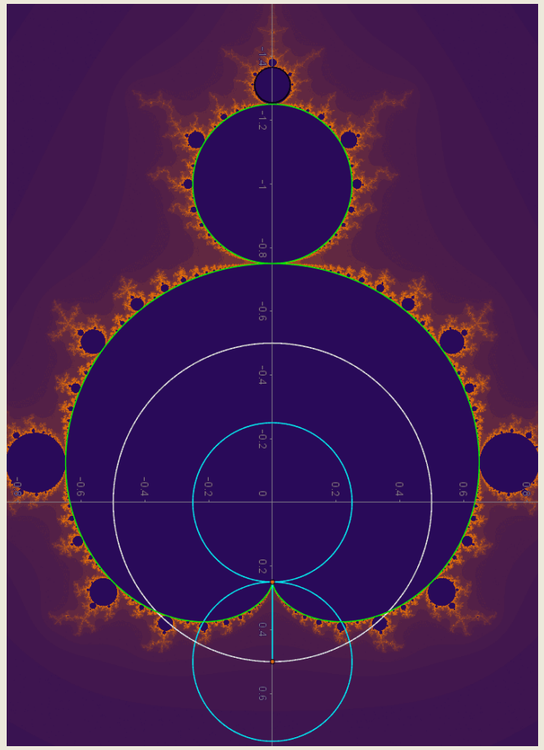
Wacław Sierpiński a fost primul matematician care s-a gândit la proprietățile acestui triunghi, dar a apărut cu multe secole mai devreme în lucrări de artă, modele și mozaicuri.

Setul Mandelbrot

Fractalul de mai sus a fost creat folosind un proces de iterație: se incepe cu un model specific, apoi se repeta iar și iar. Dar un fractal poate fi generat si in mod recursiv.

Mulțimea lui Mandelbrot este un fractal care a devenit cunoscut în afara matematicii atât pentru estetica sa, cât și pentru structura complicată, care are la bază o definiție simplă. Acest lucru se datorează în mare parte eforturilor lui Benoît Mandelbrot și ale altora de a populariza acest domeniu al matematicii.

Mulțimea lui Mandelbrot se definește ca fiind mulțimea acelor puncte c din planul complex pentru care aplicând în mod repetat polinomul complex z^2 + c (pornind de la z = 0) rezultatul rămâne în interiorul unui disc de rază finită.



Secventa de cod pentru verificarea raspunsurilor grila.

JS

```
function checkAnswer() {
  choices = document.getElementsByName("choices");
  for (var i = 0; i < choices.length; i++) {
    if (choices[i].checked) {
      choice = choices[i].value;
    }
  }

  if (choice == questions[pos].answer) {
    correct++;
  }

  pos++;

  renderQuestion();
}
```

Secventa de cod pentru customizarea atributului "buton".

CSS

```
.btn {
  margin-top: 10px;
  margin-bottom: 10px;
  background-color: rgb(179, 77, 0);
  border: none;
  color: white;
  padding: 8px 30px;
  cursor: pointer;
  font-size: 15px;
}

/* Darker background on mouse-over */
.btn:hover {
  background-color: rgb(170, 57, 0);
  color:white;
}

.btn:focus{
  background-color: rgb(179, 77, 0);
  color:white;
}
```

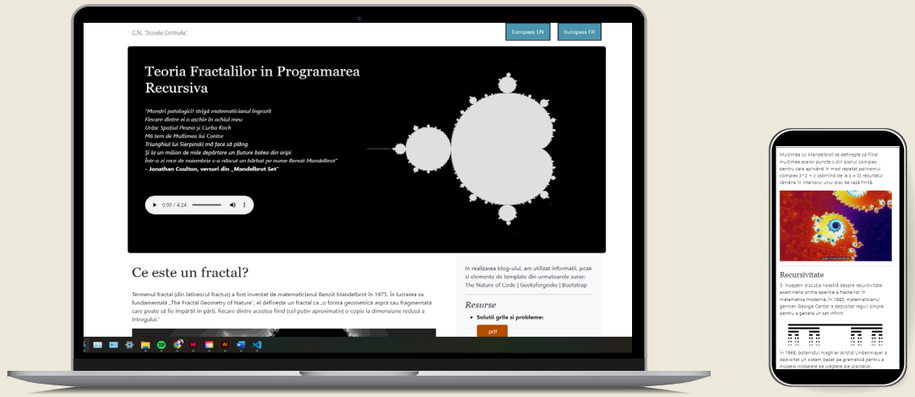

HTML

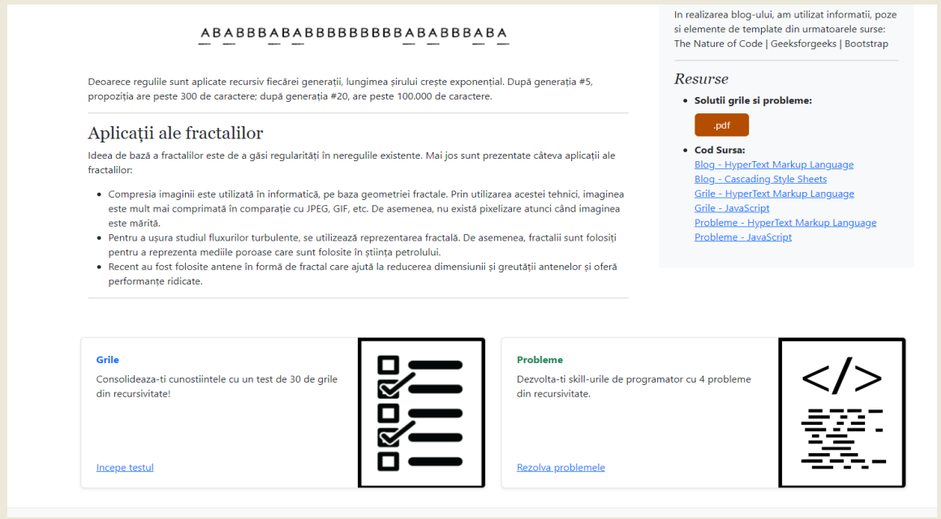
Secventa de cod pentru realizarea unui header.

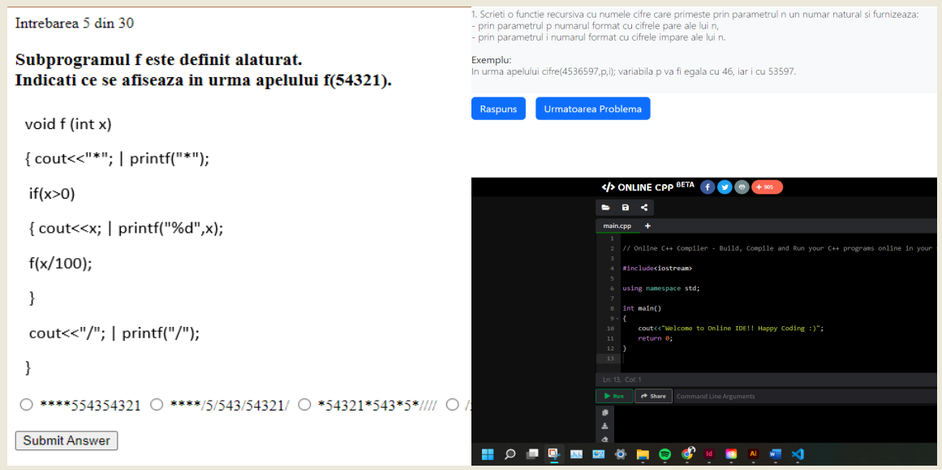
```
<!-- Header -->
<div class="container">
  <header class="blog-header lh-1 py-3">
    <div class="row flex-nowrap justify-content-between align-items-center">
      <div class="col-4 pt-1">
        <a class="link-secondary" href="http://cnscb.ro">C.N. "Scoala Centrala"</a>
      </div>

      <div class="col-4 d-flex justify-content-end align-items-center">

        <div class="dropdown">
          <button class="dropbtn">Europass EN</button>
          <div class="dropdown-content">
            <a href="assets/europass/Europass EN.pdf" target="_blank">CV 1 - Brisan</a
            <a href="assets/europass/CV Mincu Mircea.pdf" target="_blank">CV 2 - Mincu
            <a href="#" target="_blank">CV 3 - Popescu</a>
          </div>
        </div>
        <div class="dropdown">
          <button class="dropbtn">Europass FR</button>
          <div class="dropdown-content">
            <a href="assets/europass/Europass FR.pdf" target="_blank">CV 1 - Brisan</a
            <a href="#" target="_blank">CV 2 - Mincu</a>
            <a href="#" target="_blank">CV 3 - Popescu</a>
          </div>
        </div>
      </div>
    </div>
  </header>
</div>
```







Următorul este un fragment de cod care poate fi folosit pentru a calcula setul Mandelbrot.

```
/*
   Repeta un singur punct x0 + i y0 al lui
Mandelbrot pentru iterațiile „imax".
   Returnează numărul de iterații la care
punctul scapă.
   Întoarce „imax" dacă punctul nu scapă.
*/
int Iteratie(double x0,double y0,long imax)
{
  double x=0,y=0,xnew,ynew;
  int i;

  for (i=0;i<imax;i++) {
    xnew = x * x - y * y + x0;
    ynew = 2 * x * y + y0;
    if (xnew*xnew + ynew*ynew > 4)
      return(i);
    x = xnew;
    y = ynew;
  }

  return(imax);
}
```

Recursivitate

Recursivitatea reprezintă proprietatea unor noțiuni de a se defini prin ele însele.

Exemple:

factorialul unui număr: $N!=N \times (N-1)!$;
ridicarea la putere: $an=a \times a^{n-1}$;
termenul unei progresii aritmetice: $an=an-1+r$;
șirul lui Fibonacci: $F_n=F_{n-1}+F_{n-2}$;
etc.
Să observăm că aceste reguli nu se aplică întotdeauna. Dacă ar fi așa, pentru 3! am obține:

3!=3×2!, 2!=2×1!, 1!=1×0!, 0!=0x(−1)!

De aici am putea deduce că 0!=0 și înlocuind în relațiile de mai sus obținem că n!=0, pentru orice număr natural n. Bineînțeles, nu este corect. De fapt, formula recursivă pentru n! se aplică numai pentru n>0, iar prin definiție 0!=1 .

Astfel, identificăm următoarea definiție pentru n!, acum completă:
n!=
1 dacă n=0,
n x (n−1)! dacă n>0.

Similar, pentru toate formulele de mai sus exista cel puțin o situație în care formula recursivă nu se mai poate aplica, iar rezultatul se determină în mod direct.

În C++, recursivitatea se realizează prin intermediul funcțiilor, care se pot autoapela.

Grile si Probleme

Subprogramul f este definit alaturat, iar variabila intreaga r are valoarea 0 inainte de apel.
Indicati apelul in urma caruia variabila r are valoarea 2.

a) f(21,22,r) | b) f(20,21,r)

c) f(19,20,r) | d) f(18,19,r)

```
void f(int x, int y, int&z)
{ if((x%2)*(y%2)!=0) z=1;
 else { f(x/2,y/2,z); z=z+1; }
}
```

Subprogramul f este definit alaturat.
Indicati ce se afiseaza in urma apelului f(4).

a) 11111 | b) 00000

c) 01010 | d) 01101

```
void f(int x)
{ while(x>1){ x=x-1; f(x-1);}
 cout<<x; | printf("%d",x);
}
```

Subprogramul f este definit alaturat.
Indicati valoarea f(38627).

a) 2 | b) 3

c) 7 | d) 8

```
int f(int n)
{ int c;
 if (n==0) return 9;
 c=f(n/10); if (n%10<c) return n%10;
 return c;
}
```

Subprogramul f este definit alaturat.
Pentru apelul f(20,2020), functia se executa de:

a) 5 ori | b) 9 ori

c) 11 ori | d) 20 de ori

```
int f(int x, int y)
{ if(x<=1 || y<=1) return 0;
 if(x>y) return 1+f(f(x/y,y),y);
 return 1+f(x,f(x,y/x));
}
```


BIBLIOGRAFIE