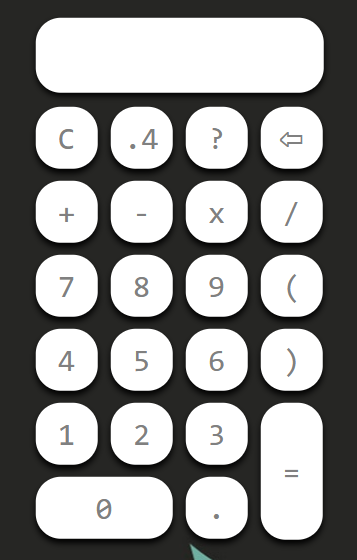
**COLEGIUL NATIONAL DE INFORMATICA “TUDOR VIANU”**

**LUCRARE DE ATESTAT**

**Simple Calculator**



**Profesor coordonator: Autor:**

**Coman Isabela Vlad Costin Andrei**

**clasa a XII-a C**

**2020-2021**

**Cuprins**

1. Introducere ..…………………………………………………………….. 2

1.1. Importanta practica

1.2. Metodologie

1. Prezentarea aplicatiei ..………………………………………………….. 3

2.1. Instalarea aplicatiei

2.2. Prezentare si utilizare

1. Prezentarea programului ..………………………………………………. 5
2. Bibliografie ..……………………………………………………………. 8

**Introducere**

Un calculator de buzunar este un dispozitiv special și relativ inflexibil pentru efectuat calcule. De-a lungul vremii acesta a evoluat in utilitate, ajungand ca in ziua de azi sa fie folosit de elevi, profesori si integrat in majoritatea laptopurilor sau calculatoarelor.

Simple Calculator este o aplicatie web care simuleaza un calculator de birou. Aceasta poate evalua expresii matematice simple, cu numere intregi si rationale, ce contin simbolurile matematice de baza: “+”, “-”, “×”, “÷” ; si paranteze.

Aplicatia a fost o oportunitate de a-mi pune in practica abilitatile de HTML, JavaScript si CSS, lucrand la un proiect de mici dimensiuni care imbina toate aceste limbaje informatice.

**Importanta practica**

Aplicatia simple calculator are ca scop rezolvarea ecuatiilor, in special in cazul problemelor de matematica, chimie sau fizica ce necesita aflarea unui rezultat cu o anumita precizie. In plus, aplicatia are si o functionalitate aditionala ce consta in generarea aleatorie a unui numar.

**Metodologie**

Aplicatia ruleaza in browser (HTML) si utilizeaza elemente de JavaScript si CSS, utilizand libraria Math din JavaScript pentru functii matematice.

**Prezentare aplicatie**

**Instalarea aplicatiei**

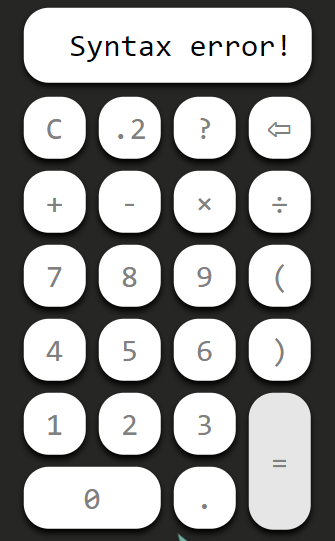
Pentru instalare, toate fisierele din *SimpleCalculator.rar* vor fi extrase intr-un folder. Pentru a porni aplicatia se va deschide fisierul *SimpleCalculator.html .*

Simple Calculator ruleaza pe orice browser, desi se recomanda utilizarea Mozilla Firefox sau Google Chrome.

**Prezentare si utilizare**

Utilizatorul va introduce date (caractere) prin apasarea butoanelor integrate din tastatura calculatorului sub forma unei expresii matematice. Exemplu:

|  |  |  |
| --- | --- | --- |
|  | 🡪 |  |



Prin apasarea butonului “=”, aplicatia computeaza in timp real rezultatul ecuatiei introduse. Pe deasupra, aplicatia verifica si corectitudinea ecuatiei, anuntand utilizatorul in cazul in care ecuatia introdusa contine erori de sintaxa.

Butonul “?” afiseaza doua pop-up-uri in care userul introduce pe rand valoarea minima si respectiv cea maxima a unui interval, sub forma de numere ***intregi***, din care programul va genera un numar ***intreg*** aleatoriu.

|  |  |
| --- | --- |
|  |  |

O alta functionalitate a aplicatiei este aceea de a afisa rezultatul calculat cu o anumita precizie: de doua zecimale, de patru zecimale si afisarea rezultatului cu toate zecimalele. Acest aspect poate fi ales de utilizator prin apasarea repetata a butonului “.2” (precizie de 2 zecimale). In urma apasarii ulterioare a respectivului buton, acesta devine “.4” (precizie de 4 zecimale) si apoi “F” (afisarea intregului rezultat, fara aproximare). Spre exemplu, la efectuarea impartirii 23 / 7 putem obtine urmatoarele rezultate:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Butonul “C”, asemenea calculatoarelor de birou, are functia de a sterge (“clear”) expresia introdusa de utilizator.

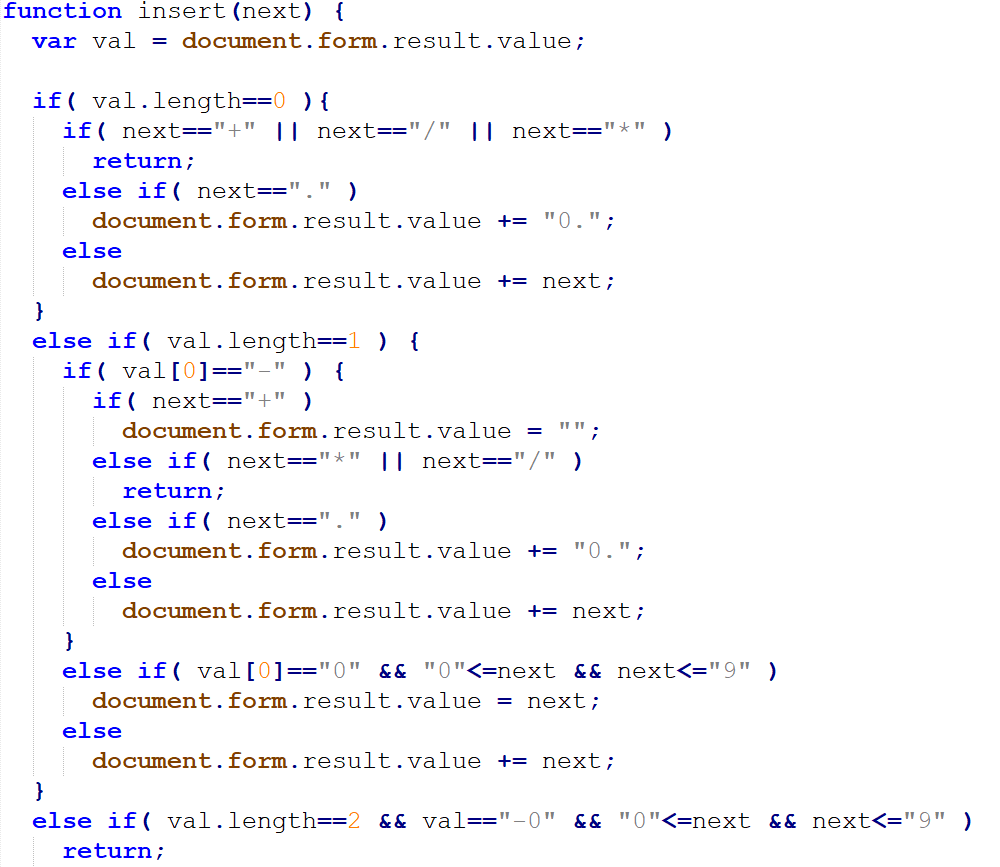
Butonul “” are scopul de a sterge ultimul caracter (numar, operatie, pananteza) introdus de utilizator, fiind usor pentru acesta sa remedieze erorile de scriere.

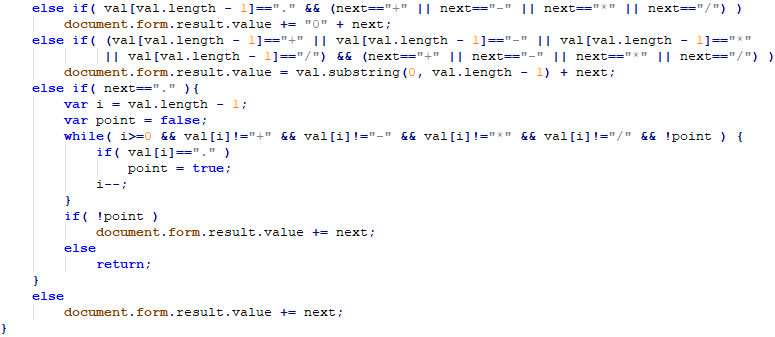
**Prezentarea programului**

Pentru a incepe discutia programului vom observa intai partea de cod HTML.

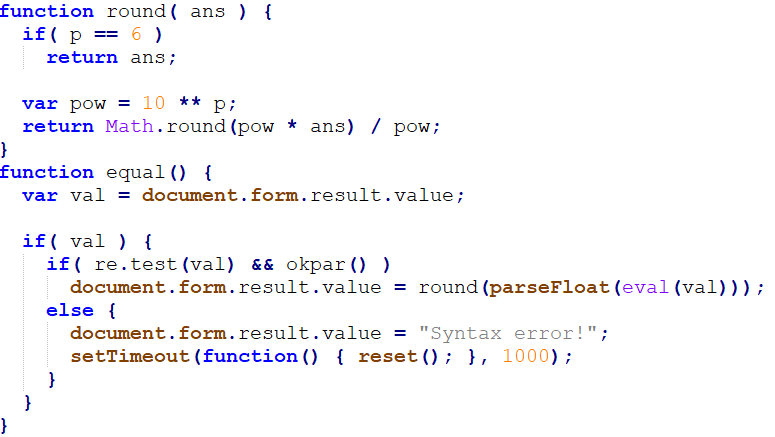


Campul in care vor aparea caracterele introduse de utilizator, cu alte cuvinte ecranul calculatorului, este un *form*. Butoanele calculatorului sunt grupate intr-un *table*, astfel incat fiecare cifra / simbol matematic / paranteza apeleaza functia *insert()* care va adauga respectivul caracter in form prin *document.form.result.value* .

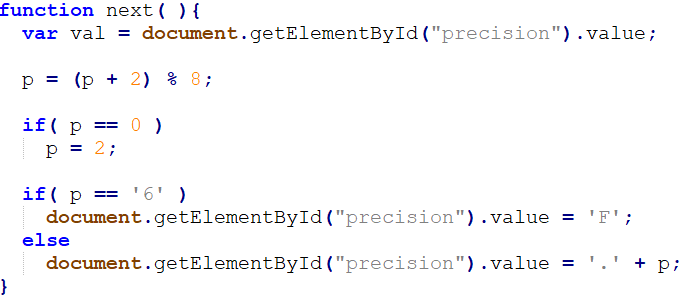




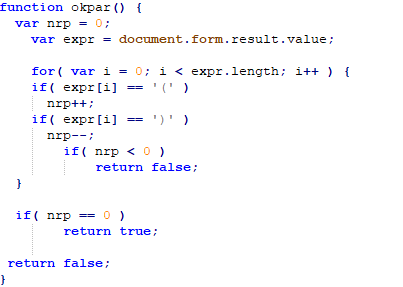
Evaluarea propriu-zisa a expresiei se va realiza apeland functia *equal()* , care preia string-ul din *form* si utilizeaza functia predefinita din JavaScript *eval()*. Un lucru important de mentionat este ca desi functia eval() poate usura atat timpul de lucru in scrierea unui cod cat si lizibilitatea, este o functie periculoasa, intrucat daca nu este folosita cu atentie, poate rula parti periculoase de cod care va pot afecta programul.



In interiorul functiei *equal()* folosim si functia *round()* care da un rezultat ce respecta cerintele de precizie ale utilizatorului. Precizia se poate schimba circular de la 2, la 4 sau la toate zecimalele unui numar, prin apelarea repetata a functiei *next()*. Aceasta verifica valoarea variabilei globale *p* care poate fi 2, 4 sau 6:



Alternativ, pentru verificarea corectitudinii unei expresii din punct de vedere al sintaxei am creat un regex care are in vedere daca expresia contine doar numere rationale, simboluri matematice de baza si/sau paranteze; cat si functia okpar() care varifica daca ecuatia este corect parantezata.





Nu in ultimul rand, am creat o functie *rand()* pentru generarea unui numar aleatoriu. Aceasta se foloseste de functiile aditionale *Math.random()* si respectiv *Math.floor()* din libraria predefinita *Math*.



**Bibliografie**

1)<https://www.w3schools.com/>

2)<https://stackoverflow.com/questions/11009320/validate-mathematical-expressions-using-regular-expression>