**Colegiul Național Costache Negri Galați**

**Lucrare pentru certificarea**

**competențelor profesionale**

Profesor Îndrumător: Elena DinuCandidați: Chiriac Robert-Cătălin

Stoian Claudiu Mihai

**Galați, 2020**

**MATH WIZARD 2.0**

**Cuprins**

[1 INTRODUCERE 4](#_Toc39958079)

[2 FUNCȚIONALITATEA APLICAȚIEI 6](#_Toc39958080)

[3 CALCULATORUL ȘI ECRANELE SUPLIMENTARE 11](#_Toc39958081)

[3.1 ALGORITMUL DIN SPATELE CALCULATORULUI 11](#_Toc39958082)

[3.2 ECRANELE SUPLIMENTARE 14](#_Toc39958083)

[3.2.1 VERIFICARE NUMĂR PRIM 14](#_Toc39958084)

[3.2.2 ÎN CÂTE CIFRE SE TERMINĂ FACTORIALUL UNUI NUMĂR 15](#_Toc39958085)

[3.3.3 SUBPROGRAMUL EGAL 16](#_Toc39958086)

[4 QUIZ 17](#_Toc39958087)

[5 CONCLUZIE 21](#_Toc39958088)

[6 BIBLIOGRAFIE 22](#_Toc39958089)

[7 ANEXE 23](#_Toc39958090)

# 1 INTRODUCERE

La început, calculele dificile sau complexe erau efectuate de oameni al căror unic rol era acesta. Acești *calculatori* se ocupau de regulă de efectuarea calculelor unei expresii matematice, de exemplu pentru efemeride în astronomie, pentru navigație sau pentru tabelele de tragere în artilerie. Calculele acelor vremuri erau foarte specializate și costisitoare și necesitau ani de pregătire [matematică](https://ro.wikipedia.org/wiki/Matematic%C4%83).

De mii de ani s-a apelat la diverse invenții, pentru a face calculele mai simple, începând prin corespondența unu-la-unu cu degetele. Primul dispozitiv a fost probabil sub forma bețelor de numărat. Alte dispozitive pentru înregistrarea numerelor folosite în [Cornul Abundenței](https://ro.wikipedia.org/wiki/Cornul_Abunden%C8%9Bei) erau formele de lut, care reprezentau numărul unor lucruri, probabil animale sau produse agricole, ținute în vase.

[Abacul](https://ro.wikipedia.org/wiki/Abac) era folosit pentru calcule aritmetice încă din 2400 î.e.n.

 In anii 1620 a fost inventată [rigla de calcul](https://ro.wikipedia.org/wiki/Rigl%C4%83_de_calcul), pentru a mări semnificativ viteza de efectuare a operațiilor de înmulțire și împărțire. Riglele de calcul au fost utilizate de generații întregi de ingineri și de profesioniști în domeniile științelor exacte, până la inventarea calculatorului de buzunar.

Pe la 1820, [Charles Xavier Thomas](https://ro.wikipedia.org/w/index.php?title=Charles_Xavier_Thomas&action=edit&redlink=1) a creat primul calculator mecanic produs în serie, aritmometrul Thomas, care putea efectua adunări, scăderi, înmulțiri și împărțiri. Calculatoarele mecanice, ca Addiator (zecimal), [comptometrul](https://ro.wikipedia.org/w/index.php?title=Comptometru&action=edit&redlink=1), calculatoarele Monroe, Curta și Addo-X au continuat să fie folosite până în anii 1970.

Era modernă a maşinilor de calcul a început cu o frenezie a dezvoltării în perioada dinainte de şi după al doilea război mondial, când componentele electronice (la acea vreme, relee, rezistoare, condensatoare, bobine, şi tuburi electronice) au înlocuit echivalentele lor mecanice, şi calculul digital a înlocuit calculul analogic.

Lucrarea lui Alan Turing din 1936 s-a dovedit extrem de influentă în domeniile informaticii şi ştiinţei calculatoarelor. Scopul principal a fost cel de a demonstra că există probleme (şi anume problema opririi) care nu pot fi rezolvate de niciun proces secvenţial. Prin aceasta, Turing a dat o definiţie a calculatorului universal care execută un program stocat pe o bandă. Această construcţie a ajuns să fie denumită maşina Turing; ea a înlocuit limbajul universal, mult mai complex şi bazat pe aritmetică, al lui Kurt Gödel. În afara limitarilor impuse de spaţiul finit de stocare, calculatoarele moderne sunt denumite Turing-complete, adică au o capabilitate de a executa algoritmi echivalentă cu cea a maşinii Turing universale.

[ENIAC](https://ro.wikipedia.org/wiki/ENIAC) (Electronic Numerical Integrator and Computer–Calculator Și Integrator Electronic Numeric), construit în SUA, a fost primul calculator electronic generic. El combina, pentru prima dată, viteza mare a componentelor electronice cu posibilitatea programării pentru probleme mai complexe. Putea efectua 5000 de operații de adunare și scădere pe secundă, fiind de o mie de ori mai rapid decât alte mașini care efectuau aceste operații. Avea și module pentru înmulțire, împărțire și rădăcină pătrată. Memoria de mare viteză era limitată la 20 de cuvinte (aproximativ 80 de octeți. Mașina era uriașă, cântărea 30 de tone, și conținea peste 18.000 de tuburi. Una dintre marile realizări inginerești ale mașinii era minimizarea arderii tuburilor, problemă comună la acea vreme. Mașina a fost utilizată aproape permanent de-a lungul următorilor zece ani.

În  ziua de astăzi, cu ajutorul tehnologiei, putem crea, folosind diverse limbaje de programare, propriile calculatoare cu ajutorul cărora putem efectua mult mai ușor și mai eficient calcule.

    Prin această lucrare de atestat dorim să demonstrăm faptul că noi, oamenii, suntem capabili de a ne construi singuri programe software pentru a ne ușura munca și, astfel, prin intermediul lor, de a minimiza marjele de eroare. Scopul acestui program este de a introduce cât mai multe metode de analiză și calcul a diverselor variabile într-o singură aplicație, astfel încât  utilizatorul  să nu întâmpine dificultăți.

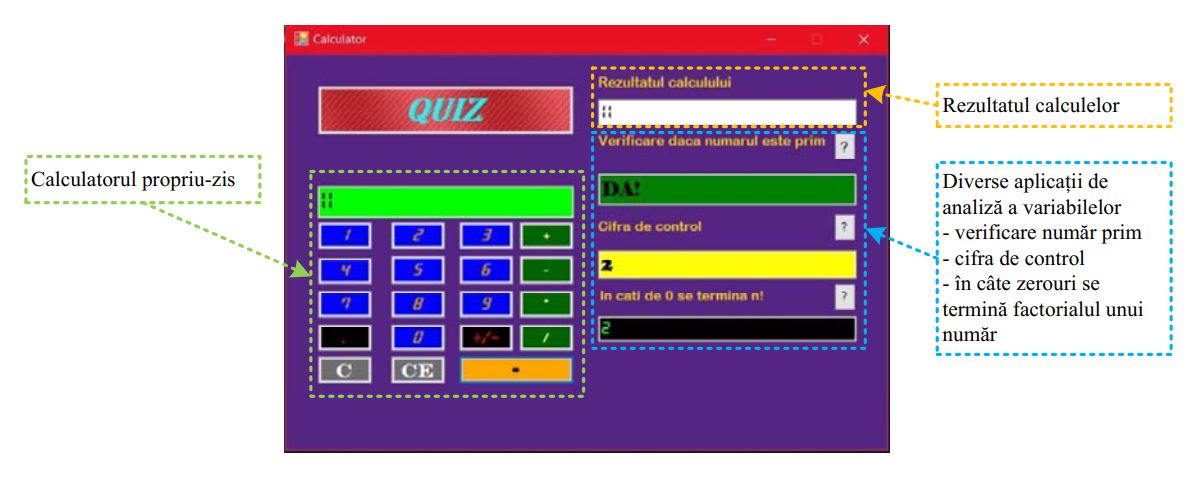


Figura 1

# 2 FUNCȚIONALITATEA APLICAȚIEI

Aplicația a fost implementată folosind următorul mediu de dezvoltare: Windows 10, Microsoft Visual Studio 2013 community, folosind limbajul de programare C#. Aplicația noastră este formata din 2 părți:

1. Calculatorul propriu-zis, cu ecrane suplimentare
2. Un Quiz de matematică

Aceasta aplicație are functionalitățile de bază ale unui calculator plus implementarea unor algoritmi ce pot fi aplicați rezultatelor operațiilor. Aplicația poate fi extinsă ușor cu alți algoritmi folosiți în mediul didactic (se mai pot adăuga ușor alte operații, proprietăți, etc.).

După cum se poate vedea din Figura 1, aplicația va preciza dacă rezultatul este un număr prim sau nu, va calcula cifra de control a acestuia dar și în câți de 0 se va termina factorialul rezultatului.

În plus am creat câteva MessageBox-uri pentru a detalia ce va afișa fiecare ecran suplimentar din aplicația noastră.

|  |  |
| --- | --- |
| Explicatie 2Figura 2 | Explicatie 3Figura 3 |
| https://scontent.fclj2-1.fna.fbcdn.net/v/t1.15752-9/96693205_623566294894842_83351678694195200_n.jpg?_nc_cat=109&_nc_sid=b96e70&_nc_ohc=CwICBfSGa5UAX_ei_Uk&_nc_ht=scontent.fclj2-1.fna&oh=a49f85e23a32c3f3e2f76cd70831fa34&oe=5EDB6F04  Figura 4 | |

De asemenea butoanele C/CE au o funcționalitate diferită față de alte calculatoare:

* Butonul C șterge rezultatul doar din ecranul principal (cel de deasupra butoanelor cu cifre / operații );
* Butonul CE șterge toate rezultatele, inclusiv cele din ecranele suplimentare.

Să luăm ca exemplu ecuația ”5+15\*1.25” . Trebuie precizat faptul ca această aplicație va efectua operațiile în ordinea în care au fost introduse deci va executa mai întâi operația de adunare și apoi cea de înmulțire. Această implementare o au și calculatoarele obișnuite daca nu sunt introduse paranteze.

Pasul 1: mai întâi vom introduce valoarea ”5”.

|  |
| --- |
| part 1  Figura 5 |

Pasul 2: apoi vom apăsa pe butonul ”+” și vom scrie introduce valoarea ”15”.

|  |
| --- |
| part 2  Figura 6 |

Calculatorul a efectuat adunarea celor două variabile, asa cum se poate vedea mai jos.

|  |
| --- |
| part 3  Figura 7 |

Pasul 3: În continuare înmulțim cu ”1.25” și apăsăm butonul ”=”.

|  |
| --- |
| part 4  Figura 8 |

Rezultat: aplicația a afișat rezultatul ecuației plus proprietățile acestui număr.

|  |
| --- |
| rezultat 1  Figura 9 |

Observatie: este posibilă utilizarea operației de împărțire dar și transformarea dintr-un număr pozitiv într-unul negativ.

A două parte a aplicației noastre este Quiz-ul format din patru întrebări (o adunare, o scădere, o înmulțire și o împărțire) ce trebuie rezolvate într-un interval de timp dat.

|  |
| --- |
| Ușor: 30  secunde  quiz usor  Figura 10 |
| Mediu: 20  secunde  quiz mediu  Figura 11 |

|  |
| --- |
| Greu: 10 secunde  quiz greu  Figura 12 |

Ca să motivăm utilizatorul  să completeze toate Quiz-urile fără să închidă aplicația am decis să introducem MessageBox-uri de “Felicitari” și să afișăm un text care informează utilizatorul că a câștigat diferite prăjituri (ne era poftă de prajituri când ne-a venit această idee).

|  |  |  |
| --- | --- | --- |
| Win 1  Figura 13 | Win 2  Figura 14 | Win 3  Figura 15 |
| Total win  Figura 16 | | |

Însă, dacă utilizatorul nu reușește să termine la timp Quiz-ul, aplicația va afișa un MessageBox care va spune utilizatorului că timpul s-a terminat (MessageBox-ul va avea mesaj diferit în funcție de dificultatea aleasă).

|  |  |  |
| --- | --- | --- |
| Fail 3  Figura 17 | Fail 2  Figura 18 | Fail 1  Figura 19 |

Pentru ca utilizatorul trebuie să fie atent la timpul rămas am decis să schimbăm culoarea căsuței timpului rămas când sunt mai puțin de 10 secunde pentru a atenționa jucătorul.

|  |
| --- |
| Schimba culoarea  Figura 20 |

În capitolele următoare vom prezenta algoritmii din spatele acestei aplicații, atât în interfața Calculatorului cât și în cea a Quiz-ului.

# 3 CALCULATORUL ȘI ECRANELE SUPLIMENTARE

## 3.1 ALGORITMUL DIN SPATELE CALCULATORULUI

În această secțiune vă vom prezenta algoritmul din spatele calculatorului, respectiv cum se introduc variabilele și cum se află rezultatul.

Variabilele principale folosite în program sunt următoarele:

* variabila **num**, de tip double, reprezintă numărul creat prin apăsarea tastelor 0,1,2,...9
* variabila **R**, de tip double, este rezultatul operațiilor
* variabila **ok**, de tip int, este folosită pentru punerea punctului de zecimală
* variabila **k**, de tipul int, este un contor pentru operații, folosita pentru ca algoritmul să funcționeze corespunzător
* variabila **change**, de tipul int, este folosită pentru transformarea numerului din pozitiv în negativ și vice-versa
* variabilele **input** și **input2**, de tip string, sunt folosite pentru a afișa cifrele apăsate în căsuța de deasupra butoanelor cu cifre (**input2** va fi folosit  când vom face schimbul dintre 5 și -5 și vice versa)
* variabila **operation**, de tip char, este folosită în subprogramul-mamă (mai multe detalli pe acest subprogram mai târziu).

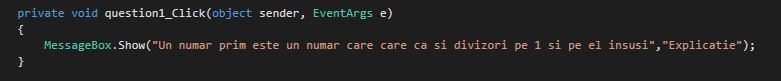
|  |
| --- |
| Figura 21 |

Subprogramele principale folosite pentru fiecare buton al calculatorului de bază (Figura 22) sunt următoarele:

* subprogramele **zero\_Click**(), **one\_Click**(), **two\_Click**(), **three\_Click**(), **four\_Click**(), **five\_Click**(), **six\_Click**(), **seven\_Click**(), **eight\_Click**(), **nine\_Click**() convertesc butoanele numerice de pe calculator în variabila **num** și  afișează acest număr în căsuța de deasupra butoanelor cu cifre.
* subprogramul **digit\_Click**() permite folosirea numerelor cu zecimală
* subprogramul **switcheroo\_Click**() permite folosirea numerelor negative; prin variabilele **input** și **input2** am permis apăsarea tastei +/- , care transformă numărul pozitiv în negativ și vice versa
* subprogramele **plus\_Click**(), **minus\_Click**(), **times\_Click**(), **div\_Click**() detectează ce taste cu rol de operații au fost apăsate și transmit informația către subprogramul-mamă.
* subprogramul **Motherboard**() primește informațiile de la celelalte subprograme, calculează rezultatul parțial și îl afișează în căsuța de deasupra cifrelor așteptând alte variable și alte operații.
* subprogramul **equal\_Click**() folosește pentru ultima oară subprogramul-mamă și va afișa rezultatele pe ecranele suplimentare.
* subprogramul **Clear\_Click**() va șterge rezultatul doar din căsuța de deasupra cifrelor
* subprogramul **ClearEverything\_Click**() va șterge și rezultatele și proprietățile din ecranele suplimentare.

|  |
| --- |
| Figura 22 |

De asemenea am adăugat câteva butoane cu semnul întrebării în caz că utilizatorul nu cunoaște terminologia. Subprogramele pentru aceste butoane se regăsesc mai jos iar ferestrele aferente acestor butoane, cu textul complet, se regăsesc in Figurile 2, 3, 4.



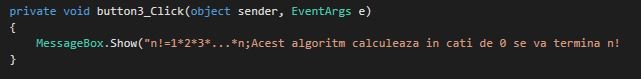
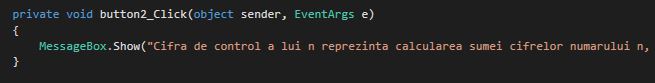
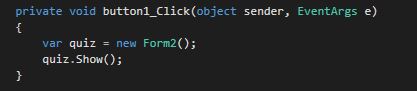


Figura 23

Singurul buton rămas este cel care va deschide fereastra Quiz-ului. Subprogramul pentru acest buton se găsește mai jos iar ferestrele aferente acestui buton se regăsesc in Figurile 10, 11, 12.



## 3.2 ECRANELE SUPLIMENTARE

În această parte vă vom prezenta algoritmii din spatele ecranelor suplimentare și cum subprogramul **equal\_Click**() afișează proprietățile în ecranul respectiv.

### 3.2.1 VERIFICARE NUMĂR PRIM

Prin folosirea subprogramului **prim()** putem calcula într-un mod eficient dacă numărul este prim sau nu.

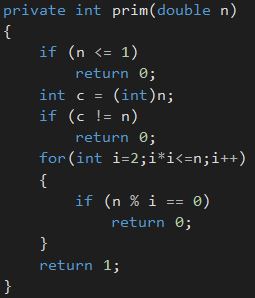


Figura 24

### 3.2.2 ÎN CÂTE CIFRE SE TERMINĂ FACTORIALUL UNUI NUMĂR

Subprogramele **putere()** și **rezolvare()** folosite împreună creează probabil cel mai eficient algoritm pentru a determina în câte cifre se termină factorialul unui număr, prin folosirea recursivității în ambele subprograme.

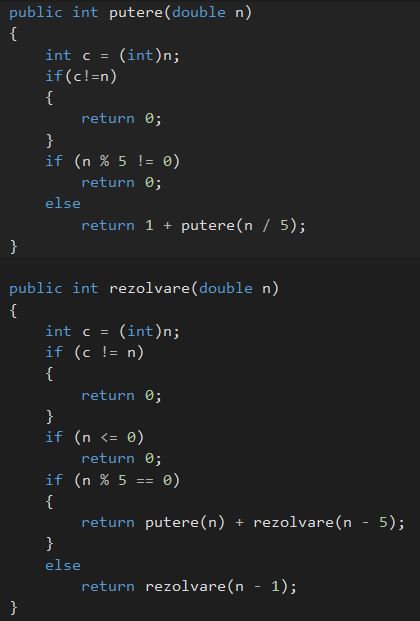


Figura 25

De exemplu, 25! se va termina în 6 zerouri: 2+1+1+1+1=6.

putere(25) returnează 2;

putere(20) returnează 1;

putere(15) returnează 1;

putere(10) returnează 1;

putere(5) returnează 1;

### 3.3.3 SUBPROGRAMUL EGAL

Subprogramul **equal\_Click**() primește rezultatul ultimei operațiuni a subprogramului-mamă - **Motherboard**() - și afișează rezultatul obținut sau proprietatea obținută în căsuța corespunzătoare, calculând totodată și cifra de control a rezultatului.

Algoritmul funcționează fără erori deoarece introducerea variabilelor se face prin apăsarea butoanelor de pe ecran deci este imposibilă adăugarea din greșeală a unor variabile care nu sunt cifre sau operații matematice (de exemplu litere sau caractere @#$& etc.).

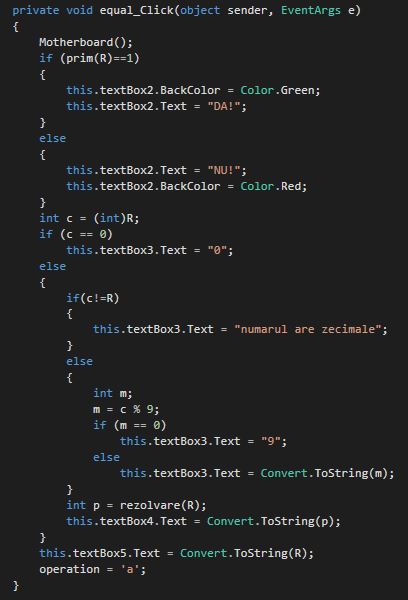


Figura 26

# 4 QUIZ

În această parte vă vom prezenta algoritmul și variabilele utilizate pentru crearea celor 4 întrebări, pentru măsurarea timpului rămas, verificarea rezultatelor și afișarea recompensei.

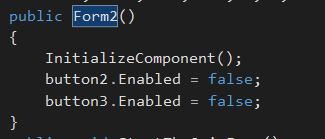


Figura 26

Pentru a nu exista posibilitatea de a face cel mai greu quiz din start (și deci să se sară peste procese ale progresului) am dezactivat butoanele care ar fi generat quiz-urile de mai mare dificultate. Acestea vor fi repornite la momentul potrivit, adică când se rezolva quiz-ul cu un nivel mai mic.

Variabilele principale folosite în program sunt următoarele:

* variabilele **AddL** și **AddR** sunt folosite pentru operația de adunare.
* variabilele **DifL** și **DifR** sunt folosite pentru operația de scădere.
* variabilele **ML** și **MR** sunt folosite pentru operația de înmulțire.
* variabilele **CutL** și **CutR** sunt folosite pentru operația de împărțire.
* variabila **TL** este folosita pentru măsurarea timpului rămas.



Figura 27

Subprogramele **button1\_Click(), button2\_Click()** și **button3\_Click()** dezactivează butoanele cât timp jucătorul completează quizul și de asemenea determină gradul de dificultate al Quiz-ului. În plus resetează căsuța care afișează timpul rămas la culorile inițiale.

|  |  |  |
| --- | --- | --- |
| button1  Figura 28 | button2  Figura 29 | button3  Figura 30 |

Subprogramele **StartTheQuizEasy(), StartTheQuizMedium()** și **StartTheQuizHard()** creează Quiz-uri de dificultate diferită și pornesc timer-ul corespunzător care va determina (dacă quiz-ul este completat corect) recompensa aferentă nivelului de dificultate.

|  |  |  |
| --- | --- | --- |
| quiz ez  Figura 31 | quiz me  Figura 32 | quiz ha  Figura 33 |

Subprogramele **timer1\_Tick(), timer2\_Tick ()** și **timer3\_Tick()** suntfolosite pe rând, în funcție de dificultatea aleasă. A fost nevoie să folosim 3 cronometre pentru a putea afișa fereastra de recompensă (sau eșec) și textul creat pentru gradul de dificultate respectiv.

Aceste subprograme au urmatoarele funcționalități:

* verifică dacă întrebările au fost răspunse corect, folosind subprogramul **CTA()**
  + dacă au fost răspunse corect va opri cronometrarea și va afișa fereastra de recompensă
* verifică dacă timpul a expirat:
  + dacă timpul nu a expirat va continua cronometrarea, cu avertizare vizuală când timpul este sub 10 secunde
  + dacă timpul a expirat și întrebările nu a fost răspunse corect va opri cronometrarea și va afișa fereastra de eșec

|  |  |
| --- | --- |
| timer1  Figura 34 | timer2  Figura 35 |
| timer3  Figura 36 | |

Subprogramul **CTA()** verifică dacă au fost răspunse corect întrebările. CTA este prescurtarea de la CheckTheAnswer și este folosit în toate timerele pentru a le opri dacă a fost rezolvat Quiz-ul sau pentru a lasa să treacă o secundă în caz că nu s-au rezolvat corect toate cele 4 întrebări.

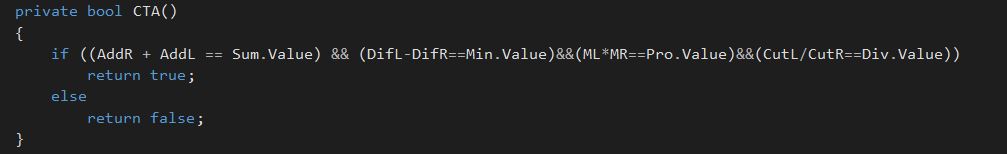


Figura 37

Subprogramul **answer\_Enter()** a fost introdus pentru cazurile în care utilizatorul introduce un răspuns care începe cu cifra 0, de exemplu 026 în loc de 26.

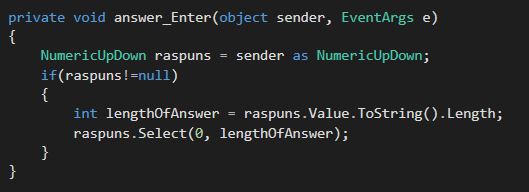


Figura 38

# 5 CONCLUZIE

Sperăm ca această lucrare să demonstreze utilitatea aplicației prezentate, care, credem noi, poate ajuta cu succes elevii pentru a rezolva mai rapid problemele și pentru a le înțelege mai în amănunt. De asemenea, aceasta aplicație ajută și la reducerea timpului petrecut în efectuarea temelor.

Prin crearea acestui calculator extins, am intenționat să le venim în ajutor nu doar elevilor, ci și altor persoane care lucrează în domenii ce necesită un calculul amănunțit al unor variabile ce pot servi ulterior în dezvoltarea unor proiecte mai complexe, cum ar fi construcția unui pod sau crearea unui alte aplicații software.

De-a lungul timpului, erorile umane au condus la dezastre ce au avut ca efect atât pierderi materiale și financiare, cât și pierderi de vieți omenești. Probabilitatea acestor erori nu va dispărea niciodată, pentru că oamenii, prin natura lor, sunt supuși greșelilor. Însă, cu ajutorul și prin intermediul inteligenței artificiale aceste greșeli pot fi cu succes evitate. Mai mult decât atât, aplicațiile pot imita inteligența umană, în acest mod sporindu-ne semnificativ productivitatea și performanța și preluând cu succes o parte din activitățile umane, cu scopul de a ne ușura viața.

# 6 BIBLIOGRAFIE

1. <https://docs.microsoft.com/en-us/visualstudio/ide/tutorial-2-create-a-timed-math-quiz?view=vs-2019>
2. <https://ro.wikipedia.org/wiki/Istoria_ma%C8%99inilor_de_calcul>

# 7 ANEXE

Aplicația s-ar putea să afișeze eronat sau să nu raspundă la comenzi în mod corespunzător dacă se folosește un calculator cu un sistem de operare mai vechi decât Windows 10, probabil din cauza problemelor de incompatibilitate dintre DOT.NET în care s-a făcut aplicația și cel de pe calculatorul pe care se testează.

Deoarece nu este sigură prezența elevilor la testarea aplicației am decis să includem un video care prezintă funcțiile aplicației, atât fereastra quiz-ului cât și cea a calculatorului cu ecranele suplimentare. În plus acest video poate fi vizionat în caz că apar problemele de compatibilitate menționate anterior. Aplicația este eficientă, rapidă însă pe video există posibilitatea ca tranziția dintre culori să se execute parțial: această întârziere se remediază în cadrul următor al filmării și este pricinuită de aplicația cu care s-a înregistrat demo-ul.