

Structura Sistemelor de Calcul

Program de testare a parametrilor de performanta ai unui PC

Autori: Rus Ionel

Grupa: 30234

# Cuprins

[1. Cuprins 2](#_Toc156303243)

[2. Introducere 3](#_Toc156303244)

[3. Studiu bibliografic 3](#_Toc156303245)

[4. Analiza si design 3](#_Toc156303246)

[5. Concluzii 7](#_Toc156303247)

[6. Bibliografie 7](#_Toc156303248)



# Introducere

Acest benchmark ofera o modalitate de a masura timpul necesar pentru a efectua hash pe un set de stringuri generate random, analizand performanta lor într-un context specific, respectiv de a afisa date despre componentele hardware ale sistemului.

Obiective:

* ***Evaluarea performantei***: Benchmark-ul are scopul de a masura performanta algoritmului de hashing, oferind o perspectiva asupra timpului necesar pentru procesarea unui set de 1000 de șiruri aleatoare.
* ***Hardware***: Prin utilizarea bibliotecii **oshi**, benchmark-ul da specificatiile hardware ale sistemului, precum CPU, GPU, RAM.
* **Îmbunătățirea performanței**: Rezultatele obtinute vor putea fi utilizate pentru a identifica posibile puncte de optimizare sau imbunatatire a performantei in cadrul operaiiilor de hashing.

# Studiu bibliografic

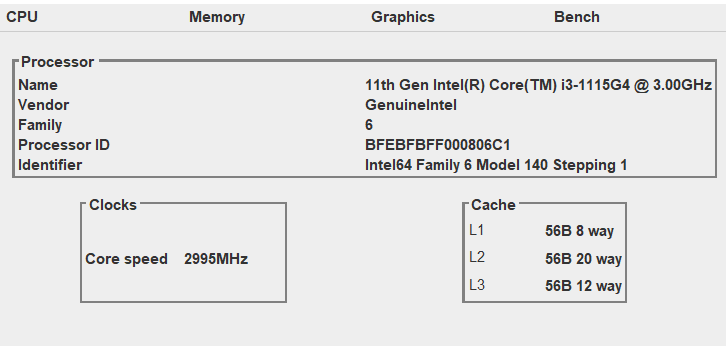
In dezvoltarea acestui benchmark, am explorat domeniul benchmark-urilor de hashing, cu accent pe implementarile in limbajul Java. Am analizat diversele abordari si tehnologii utilizate pentru a evalua eficienta algoritmilor de hashing si pentru a obtine informatii despre hardware.

# Analiza si design

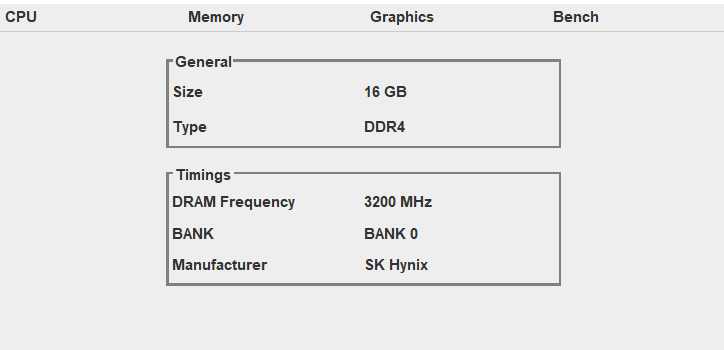
**Interfata Grafica:**

Interfata grafica este structurata intr-un meniu principal cu trei optiuni distincte: Processor (CPU), Memory (RAM) si Graphic (GPU). Aceasta organizare permite utilizatorului sa exploreze detaliile specifice fiecarei componente hardware a sistemului.

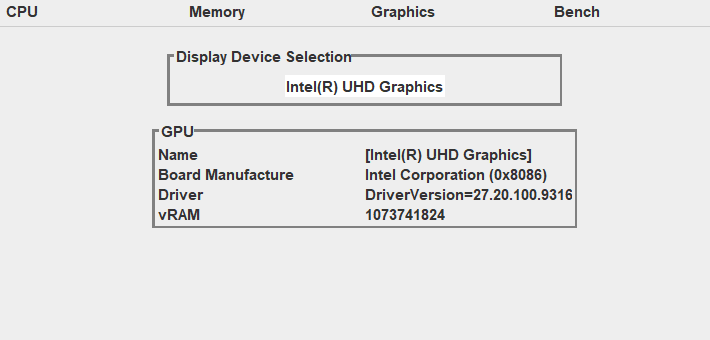
* **Panoul CPU**: La deschiderea aplicatiei, se afiseaza automat panoul CPU, care furnizeaza detalii esentiale despre procesorul sistemului. Aceste informatii includ vendorul, modelul, familia, frecventa si alte caracteristici relevante ale procesorului.



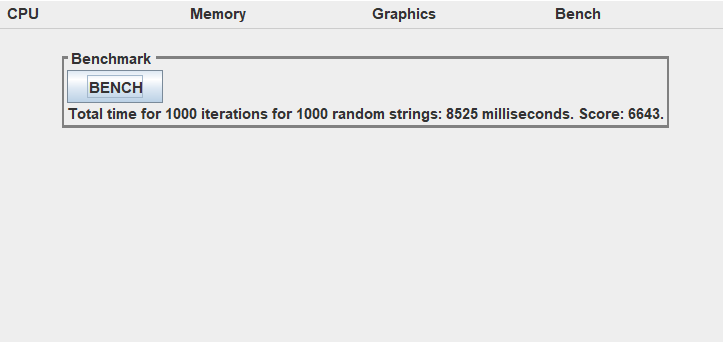
* **Panoul RAM**: Al doilea tab din meniu este dedicat memoriei RAM. Similar cu panoul CPU, acesta prezinta utilizatorului informatii despre dimensiunea, frecventa, tipul si producatorul memoriei RAM, contribuind la o intelegere comprehensiva a performantelor acestei componente.



* **Panoul GPU**: Al treilea panou este destinat unitatii de procesare grafica (GPU). Aici, utilizatorul poate examina detalii precum numele GPU-ului, producatorul placii grafice, versiunea driver-ului si cantitatea de memorie video (VRAM).

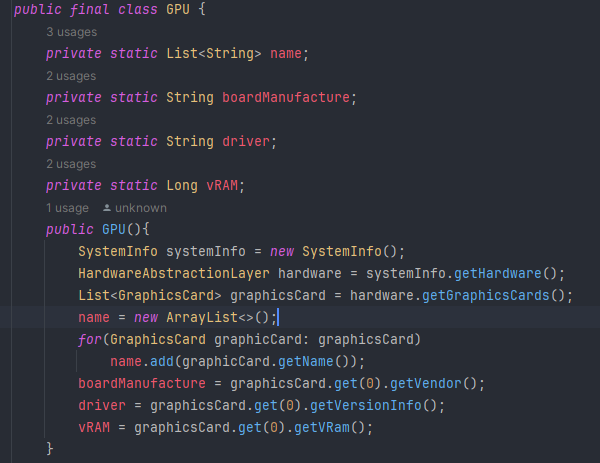


* **Panoul de Benchmark**: Programul ruleaza un hash de 1000 de ori pe un set de 1000 de siruri generate aleatoriu. Aceasta operatie ofera o masura a performantei sistemului in ceea ce priveste algoritmii de hashing, contribuind la evaluarea eficientei acestora intr-un mediu Java.

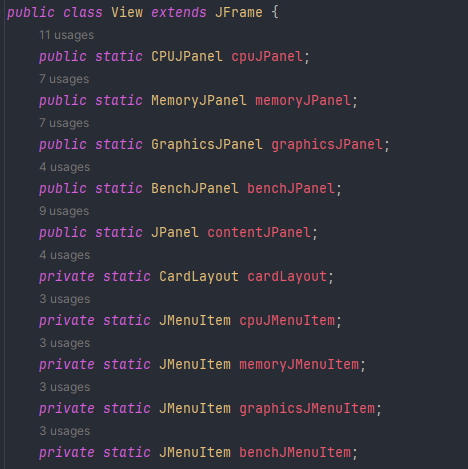


**Implementare:**

* Arhitectura proiectului este de tip MVC, modelul fiind realizat cu ajutorul bibliotecii OSHI, iar view-ul este realizat in Java Swing.



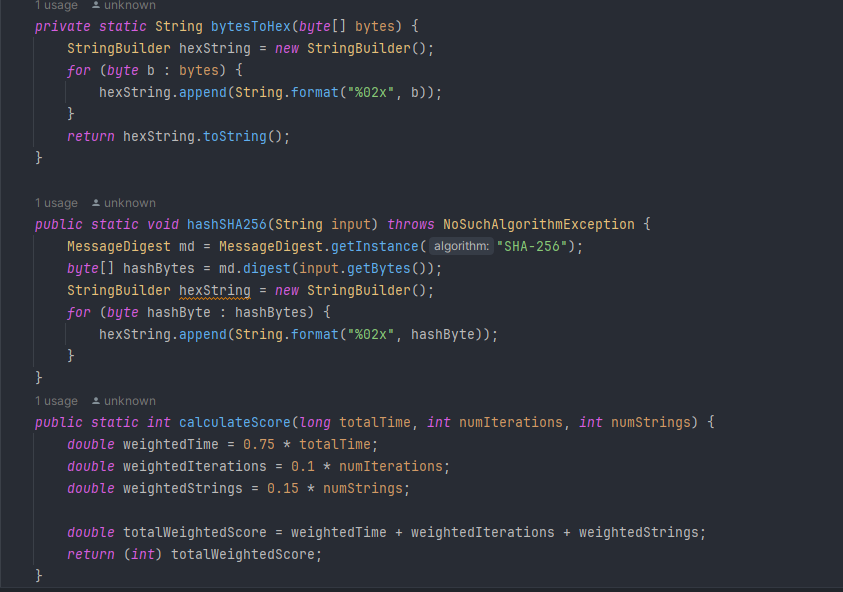
* Clasa GPU primeste de la variabila systemInfo detaliile necesare pentru a le putea afisa. Clasele RAM, respective Processor sunt implementate la fel.



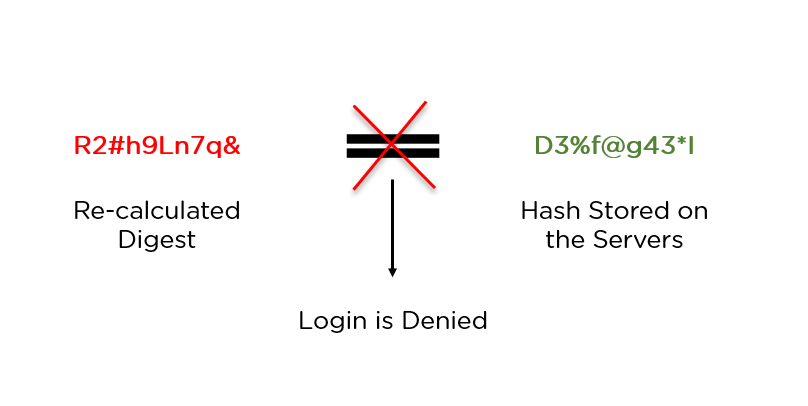
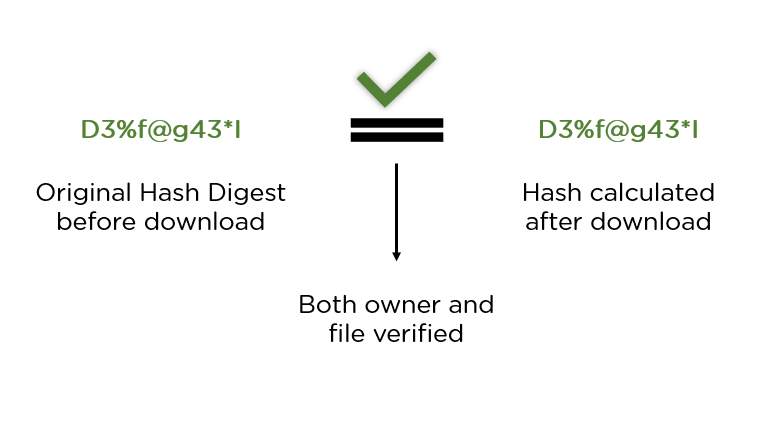
* Pentru partea de View ne folosim de implementare pentru fiecare panou separat unde le initializam si adaugam logica pentru butoane.



* Clasa Bench contine mai multe metode printre care metoda run cu ajutorul careia facem operatia si primim rezultatul in urma operatiilor facute.



* Metoda de hashSHA256 a fost aleasa pentru ca e una dintre cele mai bune metode de tip SHA oferind rezultate bune intr-un timp relativ mic. Metoda de calculateScore este implementata folosind o formula de suma ponderata in functie de numarul de iteratii, numarul de stringuri si de timpul total cat a durat operatia.

# Concluzii

Prin integrarea unei interfete grafice, programul nu doar furnizeaza informatii utile despre componentele hardware ale sistemului, ci si ofera un instrument practic de testare a performantelor acestuia.

# Bibliografie

1. <https://en.wikipedia.org/wiki/SHA-2>
2. <https://simplesolution.dev/java-get-cpu-information-oshi-library/>
3. <https://www.javatpoint.com/java-swing>
4. <https://www.geeksforgeeks.org/introduction-to-java-swing/>
5. https://www.simplilearn.com/tutorials/cyber-security-tutorial/sha-256-algorithm