**Comparatie Test Cases create de noi VS Test Cases create de catre Gemini**

**Structura Suitei de Teste**

* Ambele suite de teste sunt structurate similar, având metode pentru configurarea mediului de testare și diverse implementări de **TestMethod** pentru a testa funcționalități specifice.
* Suita de teste creata de catre Gemini, include o varietate de teste care se concentrează pe adăugarea etapelor (fazelor), scorurilor și verificarea compatibilității platformei.
* Suita de teste creata de catre noi, acoperă funcționalități similare, dar include și teste structurale și de mutație.

**Acoperirea Testelor**

* **Adăugarea Fazelor și Scorurilor:** Ambele suite testează capacitatea de a adăuga faze și scoruri jocului, tratează cazuri limită precum intrări nule sau goale și diferite tipuri de scoruri (pozitive, zero și negative).
* **Testarea Output-ului:** Ambele suite implică testarea output-ului, asigurându-se că mesajele corecte sunt afișate pentru platformele compatibile și incompatibile. De asemenea, verifică afișarea corectă a fazelor și a informațiilor despre joc utilizând o clasă personalizată **ConsoleOutput** pentru capturarea output-urilor consolei.
* **Tratarea Excepțiilor:** Testarea excepțiilor este proeminentă în ambele suite. Ele testează pentru excepții așteptate atunci când sunt furnizate intrări invalide, cum ar fi valori nule pentru numele fazelor sau verificări ale platformei.
* **Compatibilitatea Platformei:** Ambele suite abordează verificarea compatibilității platformei, inclusiv sensibilitatea la variații de caz în numele platformelor. Acest lucru asigură că jocul recunoaște "PC" ca echivalent cu "pc" și alte variații de caz.
* **Teste de Mutație:** Suita noastra prezinta explicit unele teste ca fiind de mutație, indicând un focus pe modificarea intrărilor pentru a testa diverse ramuri în logica codului.

**Diferențe în Detaliu**

* **Teste de Mutație și Structurale Explicite:** Testele noastre prezinta testarea mutației și testarea structurala, reflectând o abordare mai detaliată și concentrată pe asigurarea robusteții codului față de schimbări și pentru a acoperi mai multe scenarii.
* **Utilizare Extinsă a Utilităților:** Am folosit repetat **ConsoleOutput** pentru a captura și analiza output-urile consolei în multiple teste, indicând o abordare ușor mai rafinată pentru testarea output-ului consolei în diferite cazuri de test.
* **Verificarea Mesajelor de Eroare:** Testele noastre acopera conținutul mesajelor de excepție, adăugând un strat suplimentar de detaliu la procesul de testare pentru a asigura nu numai că sunt aruncate excepții, dar și că acestea sunt informative și corecte.

**În Concluzie**

* Ambele suite de teste sunt bine proiectate pentru a asigura functionalitatea clasei **JocVideo** așa cum este așteptat pentru diverse actiuni. Totuși, testele create de catre noi includ unele rafinamente suplimentare în verificările mesajelor de eroare și în testele de mutație care oferă o acoperire mai larga in mai scenario diferite și robustețe.