**PROIECT IP:**

**2048 GAME**

**COORDONATOR:**

**Prof. Cristian Sandu**

**STUDENT:**

**Dinu Neculai**

**Grupa: 22c32a**

**An universitar: 2022-2023**

**Anul 3 CTI**

Contents

[***INTRODUCERE*** 3](#_Toc125644673)

[***Analiza Jocului*** 3](#_Toc125644674)

[***Ce sunt design patternurile?*** 3](#_Toc125644675)

[***LIMBAJUL JAVA*** 4](#_Toc125644676)

[***Utilizarea limbajului in cadrul proiectului:*** 4](#_Toc125644677)

[***Alte utilitati folosite in crearea acestui proiect:*** 4](#_Toc125644678)

[***Cum se calculeaza indicele de stabilitate?*** 5](#_Toc125644679)

[***CODUL SURSA AL PROIECTULUI*** 6](#_Toc125644680)

[***STRUCTURA DE CLASE/PACHETE*** 6](#_Toc125644681)

[***SABLOANE UTILIZATE:*** 7](#_Toc125644682)

[***Singleton:*** 7](#_Toc125644683)

[***Command:*** 7](#_Toc125644684)

[***Iterator*** 8](#_Toc125644685)

[***Memento*** 8](#_Toc125644686)

[***Adapter*** 8](#_Toc125644687)

[***PLAN DE TESTARE*** 9](#_Toc125644688)

# ***INTRODUCERE***

## ***Analiza Jocului***

2048 este un joc creat de Gabriele Cirulli in 2014, acest joc are ca scop imbinarea a 2 casute cu numere la fel pentru a le multiplica si astfel sa se ajunga la numarul de 2048, pornind de la numarul 2, acestea apoi inmultindu-se intre ele si ajungand la rezultatul de 2048. Jocul poate continua si peste acest numar.

Jocul se joaca pe o grila de 4x4, jocul va incepe cu 2 numere pe tabla, la fiecare apasare de tasta(stanga,dreapta, sus, jos) se vor mai adauga alte numere, cu sansa de 50% sa fie 2 sau 4.

Jocul e unul simplu, poate fi jucat offline cu usurinta, e un joc ce foloseste de obicei putine resurse.

## ***Ce sunt design patternurile?***

*Un Design Pattern este o soluție generală la o problemă comună care poate apărea în proiectarea de software. Acestea sunt bazate pe experiența dezvoltatorilor și au ca scop să ofere o abordare organizată și eficientă pentru rezolvarea problemei. Avantajele utilizării unui Design Pattern sunt:*

* *Asigură o structură bine definită pentru cod, ceea ce face mai ușor de înțeles și de întreținut.*
* *Îmbunătățește comunicarea între dezvoltatori, deoarece există un limbaj comun pentru a descrie soluțiile.*
* *Oferă o soluție testată și dovedită pentru o problemă specifică, ceea ce poate reduce riscul de a introduce erori în cod.*
* *Poate fi adaptat pentru a se potrivi cu cerințele specifice ale proiectului.*

*Design Pattern-urile sunt utilizate în diverse limbaje de programare și sunt un instrument valoros pentru dezvoltatorii care doresc să creeze software de calitate și scalabil.*

## ***LIMBAJUL JAVA***

Java este un limbaj de programare orientat pe obiect, conceput de către James Gosling la Sun Microsystems (acum filială Oracle) la începutul anilor ʼ90, fiind lansat în 1995. Cele mai multe aplicații distribuite sunt scrise în Java, iar noile evoluții tehnologice permit utilizarea sa și pe dispozitive mobile, spre exemplu telefonul. În felul acesta se creează o platformă unică, la nivelul programatorului, deasupra unui mediu eterogen extrem de diversificat. Acesta este utilizat în prezent cu succes și pentru programarea aplicațiilor destinate intranet-urilor.

Avantajele unui limbaj orientat pe obiect ca si Java sunt:

* Cod usor de reutilizat
* O structura a codului mai transparenta
* Siguranta(Incapsulare)
* Dezvoltare rapida datorita sistemului de clase, acestea permitandu-ne sa lucram pe diferite clase in acelasi timp.

## ***Utilizarea limbajului in cadrul proiectului:***

Pentru crearea jocului 2048 a fost folosit Java ca si limbaj de programare, interfata acestuia a fost facuta prin Java Swing iar testele acestuia au fost facute cu ajutorul JUnit

## ***Alte utilitati folosite in crearea acestui proiect:***

IDE: IntelliJ

Version Control: Git

Hosting: GitHub

# ***Cum se calculeaza indicele de stabilitate?***

Indicele de stabilitate este o măsură a gradului în care o clasă poate fi modificată fără a afecta alte clase din proiect. În proiectele Java, acest indice poate fi calculat utilizând metoda LCOM (Lack of Cohesion of Methods). Există mai multe variante de calcul pentru LCOM, dar cele mai utilizate sunt LCOM2 si LCOMHS.

LCOM2:

* Se contorizează câte metode din clasă accesează fiecare variabilă membru.
* Se calculează numărul de metode care accesează aceeași variabilă membru.
* Se calculează LCOM2 prin împărțirea numărului de metode care accesează aceeași variabilă membru la numărul total de metode din clasă.

LCOMHS:

* Se contorizează câte metode din clasă accesează fiecare variabilă membru.
* Se calculează numărul de metode care accesează aceeași variabilă membru.
* Se calculează numărul de metode care nu accesează aceeași variabilă membru.
* Se calculează LCOMHS prin împărțirea numărului de metode care accesează aceeași variabilă membru la numărul total de metode care nu accesează aceeași variabilă membru.

Valorile obținute de LCOM2 și LCOMHS pot fi cuprinse intre 0 si 1. Valorile mai mari indica o clasa mai putin stabila si mai putin usor de modificat.

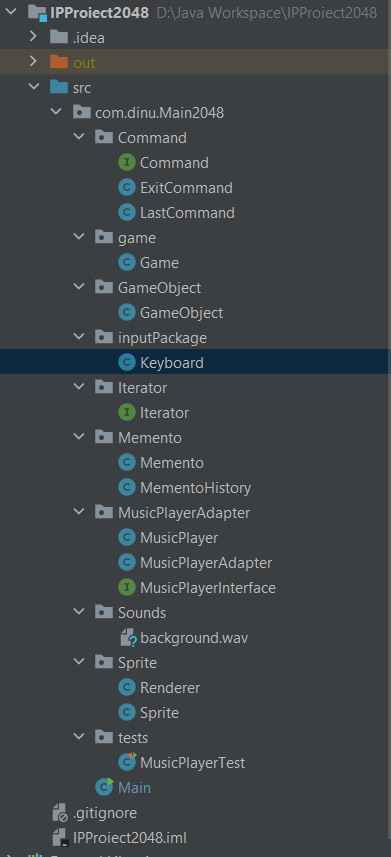
Există mai multe tool-uri de analiza de cod, cum ar fi sonarqube, PMD sau FindBugs, care pot fi utilizate pentru a calcula LCOM într-un proiect Java.

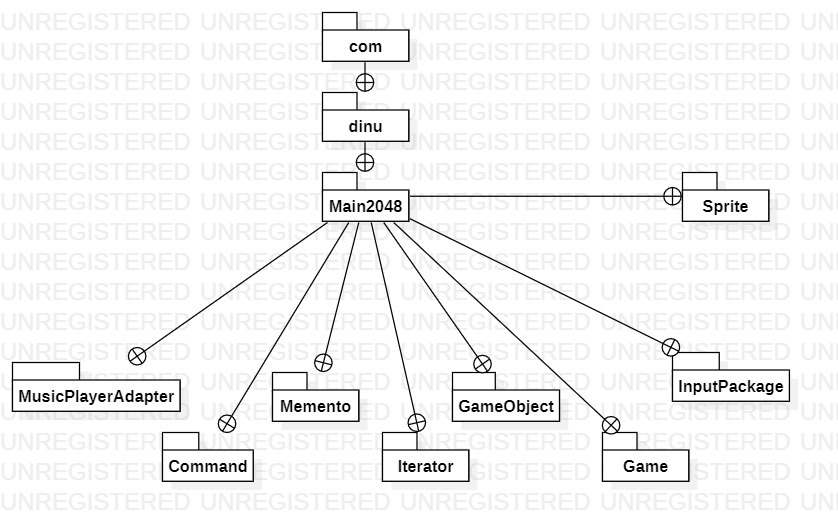
# ***CODUL SURSA AL PROIECTULUI***

Codul sursa al proiectului se gaseste pe GitHub, acesta a fost creeat pas cu pas si folosind commituri pe parcursul dezvoltarii jocului se poate tine un istoric al modificarilor/implementarilor

Link catre github: <https://github.com/N1cuDinu/IPProiect2048>

# ***STRUCTURA DE CLASE/PACHETE***

****

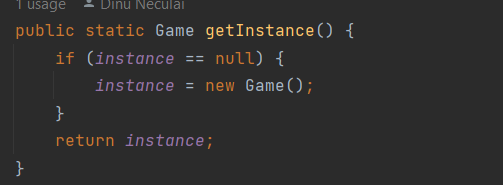
****

# ***SABLOANE UTILIZATE:***

## ***Singleton:***

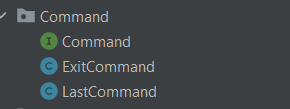
-In cadrul acestui joc singleton a fost folosit pentru a ne asigura ca exista o singura instanta de tip „Game” creeata la momentul rularii jocului, astfel acesta ne face aplicatia sa fie mai rapida.

-Tip Sablon: Creational

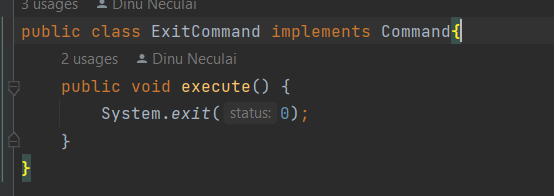


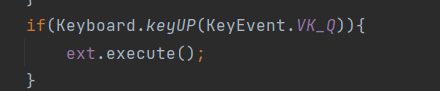


## ***Command:***

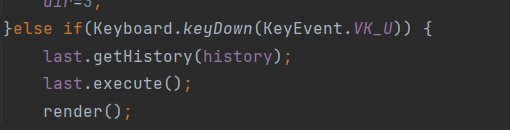
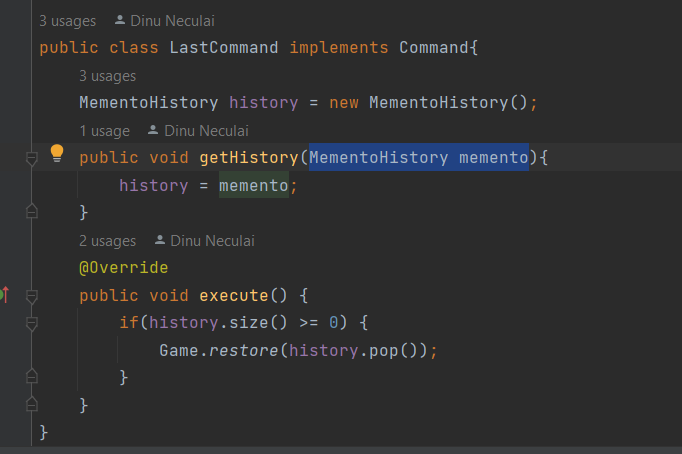


-Command a fost folosit pentru crearea unui buton de exit, la actionarea butonului Q de pe tastatura jocul nostru se va inchide.





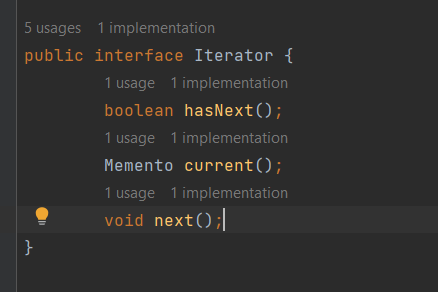
-De asemenea a fost folosita interfata Command pentru a facilita utilizarea patternului Memento, la actionarea butonului U, jocul se va relua cu miscarea anterioara.

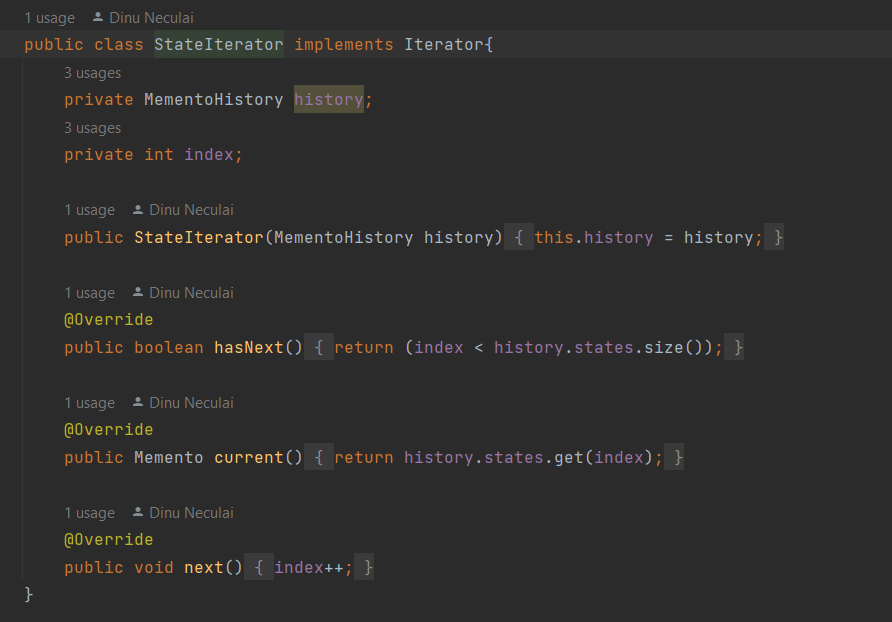


-Tip Sablon: Behavioral

## ***Iterator***

-Iteratorul a fost utilizat pentru parcurgerea numerelor create si stocarea lor pentru a lucra si cu sablonul memento.



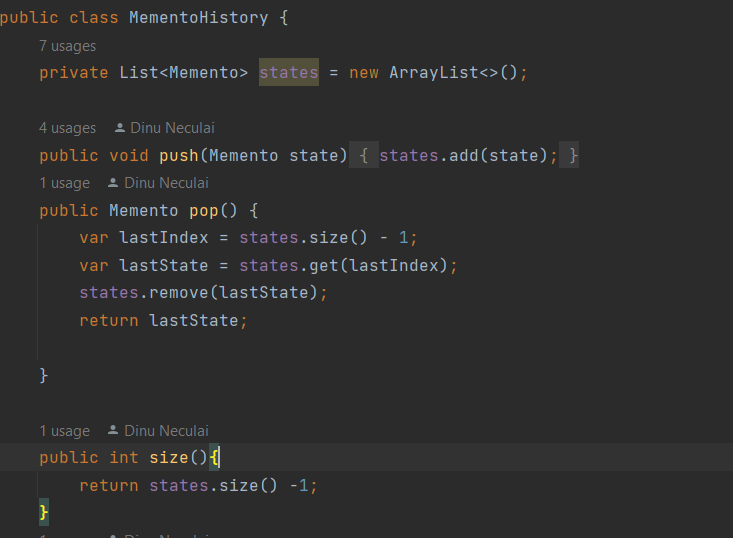


-Tip Sablon: Behavioral

## ***Memento***

-Memento este folosit pentru a tine evidenta miscarilor in cazul in care utilizatorul doreste sa se intoarca inapoi cu o miscare, acesta cu ajutorul iteratorului parcurge fiecare miscare in parte si astfel ne permite sa mutam inapoi la fiecare actiune.



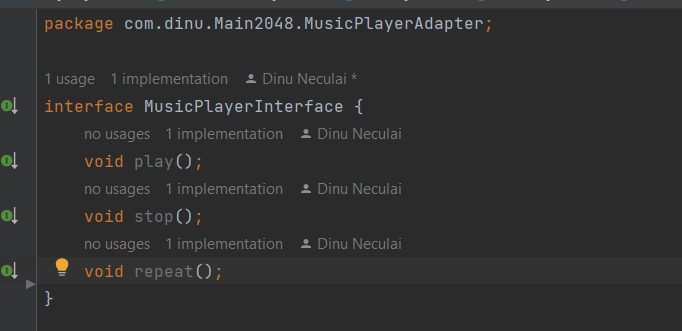


-Tip Sablon: Behavioral

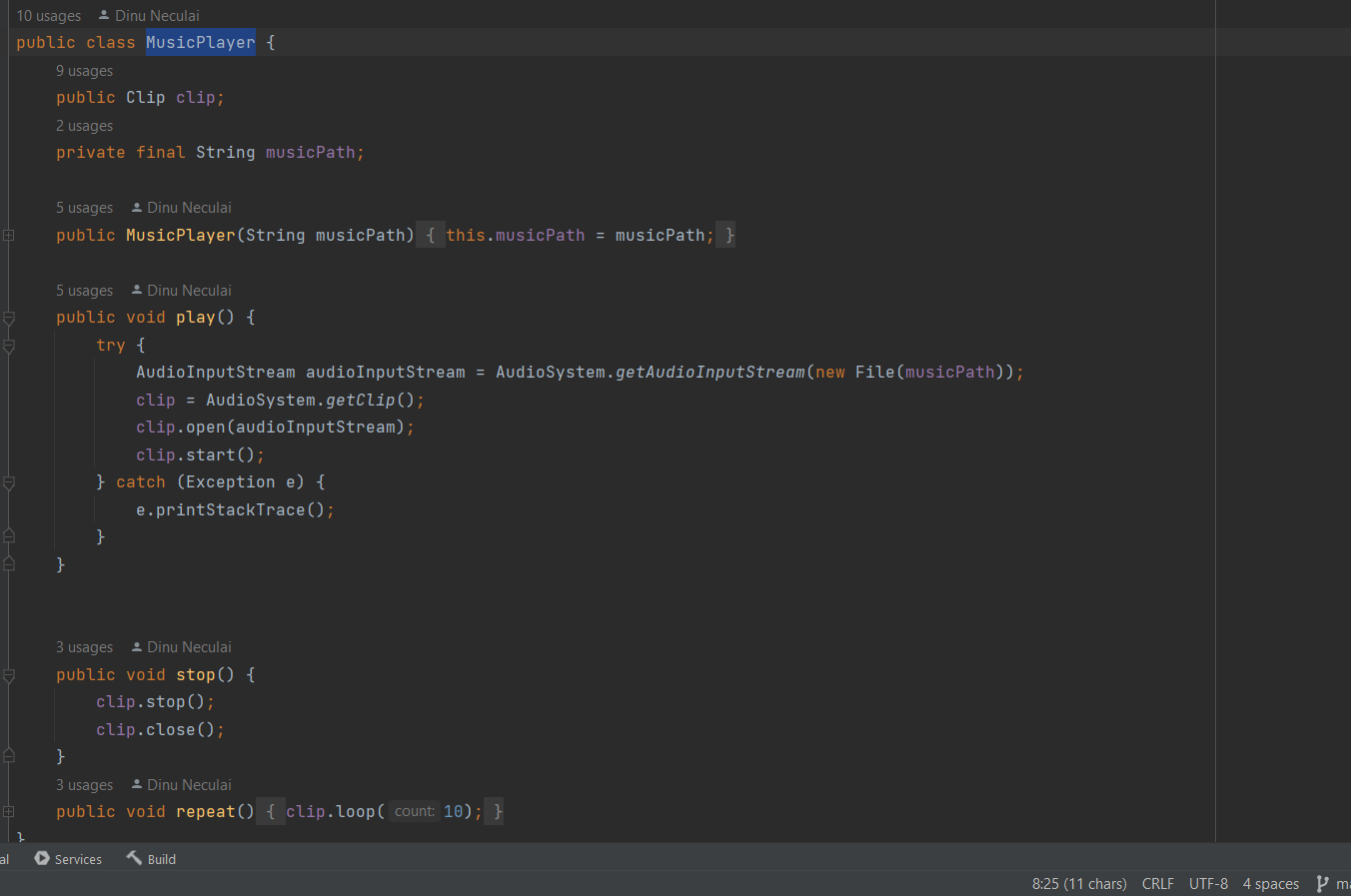
## ***Adapter***

-Adapter a fost utilizat pentru adaugarea muzicii de background, acesta permitandu-ne sa combinam 2 obiecte cu interfete incompatibile.

-Tip Sablon: Structural







# ***PLAN DE TESTARE***

Pentru testare am ales sa ma folosesc de JUnit, testand sablonul Adapter si functionalitatile implementate cu ajutorul sau.



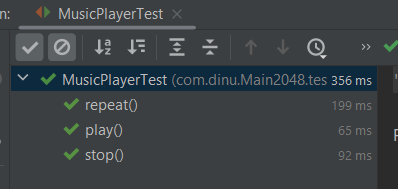
Cu acest test verificam astfel ca cele 3 functii ale Music playerului:

-Play

-Stop

-Repeat

In acesta vom incarca fisierul ce il folosim drept background music, si verificam astfel ca aceste functii sa lucreze asa cum ne dorim.



# ***SCREENSHOT DIN JOC***

