Proiect

Ingineria Programelor

-Drop the number-

Coordonator, Student,

Ş.l. dr. inf. Sabina Bogdan-Iulian

COSTACHE ANDREI

ANUL III CTI

GRUPA 22C32A

AN 2021-2022

**Introducere**

În ingineria software, un model de proiectare este o soluție generală repetabilă la o problemă frecventă. Un model de design nu este un design finit care poate fi transformat direct în cod, ci este o descriere sau un șablon pentru rezolvarea unei probleme care poate fi folosită în multe situații diferite.

Modelele de proiectare pot accelera procesul de dezvoltare prin furnizarea de paradigme de dezvoltare testate și dovedite. Proiectarea eficientă a software-ului necesită luarea în considerare a problemelor care pot să nu devină vizibile decât mai târziu în implementare. Reutilizarea modelelor de proiectare ajută la prevenirea problemelor subtile care pot cauza probleme majore și îmbunătățește lizibilitatea codului pentru programatori și arhitecții familiarizați cu modelele.

**Etapele Proiectului:**

- Analiza

- Design

- Implementare

- Testare

**1)Analiza jocului**

Drop the number - [Drop the Number | Math Playground](https://www.mathplayground.com/logic_drop_the_number.html)

Drop the number este un joc al cărui scop este obținerea unui pătrat cu scor cât mai mare, scorul pătratului crescând unindu-l cu un pătrat de aceeasi valoare. Jocul se termină atunci când una din cele cinci coloane nu mai are spațiu liber pentru a intra un alt pătrat. Fiind un joc logic, este nevoie de o strategie bine pusă la punct, cât și de răbdare.

Regulile jocului:

-folosind săgețile, pătratul se mută de pe un rând pe altul

-pătrațelele își dublează valoarea atunci când două de aceeași valoare sunt puse una lângă alta, transformându-se într-una singura.

-dacă o coloană se umple, jocul o ia de la început.

**2) Ierarhia pachetelor**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**3) Prezentarea șabloanelor**

În acest proiect s-au folosit următoarele șabloane:

**1)Singleton**

**Singleton** este un model de proiectare creațional care vă permite să vă asigurați că o clasa are o singură instanță, oferind în același timp un punct de acces global la această instanță.

Toate implementările Singleton au acești doi pași în comun:

* Faceți constructorul implicit privat, pentru a împiedica alte obiecte să folosească operatorul new cu clasa Singleton.
* Creați o metodă statică care să acționeze ca un constructor. În interior această metodă apelează construcorul privat pentru a crea un obiect și a-l salva ca un câmp static.

Singleton trebuie folosit când o clasă din program ar trebui să aibă doar o singură instanță disponibilă pentru toți clienții.

Table

Description automatically generated

În acest caz, ne asigurăm că “appUI” are o singură instanță disponibilă pentru toți clienții.

**2)Observer**

**Observer** este un model de proiectare comportamental care ne permite să definim un mecanism de abonament pentru a notifica mai multe obiecte despre orice evenimente care se întâmplă cu obiectul pe care îl observă.

Publisher-ul emite evenimente de interes pentru alte obiecte. Aceste evenimente apar atunci când publisher-ul își schimbă starea sau execută unele comportamente. Publisher-ul conține o infrastructură prin care abonații noi se pot alătura iar cei actuali să părăsească lista.

Subscriber-ul declară interfața de notificare. În majoritatea cazurilor, constă dintr-o singură metodă update.

Diagram

Description automatically generated

În cazul nostru, scorul se updatează în clasa “appUI”, atunci când in clasa “board” se fac modificări.

**3)Command**

În programarea orientată pe obiecte, **command** este un model de design comportamental în care un obiect este folosit pentru a încapsula toate informațiile necesare pentru a efectua o acțiune sau pentru a declanșa un eveniment la un moment ulterior. Aceste informații includ numele metodei, obiectul care deține metoda și valorile pentru parametrii metodei.

Text

Description automatically generated

În cazul de față, clasa “arrowButton” are un actionListener care este făcut doar de aceasta.

**4)Memento**

**Memento** este un model de design comportamental care vă permite să salvați și să restaurați starea anterioară a unui obiect fără a dezvălui detaliile implementării acestuia.

.

Text

Description automatically generated

În cazul nostru, clasa “Memento” ne ajută să revenim la un stadiu anterior al clasei “board”, prin apăsarea butonului de “undo”.

**4) Plan de testare**

JUnit este un cadru de testare unitară pentru limbajul de programare Java. JUnit a fost important în dezvoltarea dezvoltării bazate pe testare și face parte dintr-o familie de cadre de testare unitară, cunoscută în mod colectiv sub numele de xUnit, care a provenit din SUnit.

Graphical user interface, text, application

Description automatically generated

**5) Capturi de ecran**

Application

Description automatically generated with low confidence**Începutul jocului**

**A screenshot of a cell phone

Description automatically generated with medium confidenceÎn timpul jocului**

**Finalul jocului.**

**A screenshot of a cell phone

Description automatically generated with medium confidence**