

UTCN
Computer Science Department
Proiectare Software
Nume: Galis Laurentiu
Grupa: 30236

Proiect

1. Obiectiv

Obiectivul acestei teme este de a aprofunda elementele de baza a modelului de programare Model-View-Presenter si limbajul de programare ales.

2. Cerinta

Dezvoltați (analiză, proiectare, implementare) o aplicație desktop care poate fi utilizată ca soft educațional pentru studiul poliedrelor. Aplicația va avea un singur tip de utilizator care va putea efectua următoarele operații fără autentificare:

- ❖ desenarea interactivă a poliedrelor (prismă, paralelipiped dreptunghic, cub, piramidă, tetraedru, trunchi de piramidă) prin înlocuirea creionului și a riglei cu mouse-ul și alegerea stilului de desenare (inclusiv culoarea);
- ❖ calcularea și afișarea unor proprietăți: aria laterală, aria bazei, aria totală, volum;
- ❖ determinarea și desenarea poliedrului obținut prin simetrie față de un plan;
- ❖ salvarea / încărcare unui poliedru într-un/dintr-un fișier xml.

3. Alegerea limbajului de programare

Am ales limbajul de programare Java deoarece este unul dintre cele mai folosite limbaje de programare bazate pe OOP(Oriented Object Programming) si unul dintre limbajele pe care le stapanesc cel mai bine. In plus este limbajul folosit cel mai des in facultate si unul dintre cele mai usoare de invatat si folosit pentru dezvoltarea aplicatiilor. Pe langa acest limbaj am folosit si limbajul XML cu care am putut implementa interfata grafica si a face mai usor interactiunea utilizatorului cu aplicatia.

4. Instrumentele utilizate

Pentru dezvoltarea acestei aplicatii am folosit mediul de programare IntelliJ in care am scris codul in Java siFXML. In acest mediu de programare se pot instala extensii care fac mai usoara dezvoltarea unei aplicatii. La dezvoltarea interfetei grafice am folosit tool-ul SceneBuilder care se

foloseste de biblioteca JavaFX in care se afla componentele corespunzatoare unei interfete grafice(Label-uri, butoare, text field-uri, etc). In SceneBuilder se poate realiza o interfata grafica cu drag and drop si in functie de asezarea obiectelor in scena se modifica codul in FXML. Asadar, acest tool face mai usor vizualizarea obiectelor declarate in fisere FXML sau XML.

Am ales mediul de programare IntelliJ deoarece face mult mai usor de realizat scrierea si indexarea codului in limbajul Java. Mai mult de atat se pot vedea mult mai bine si mai clar pachetele si clasele create pentru dezvoltarea aplicatiei.

5. Sablonul de proiectare creational

Sabloanele creationale privesc modul de creare al obiectelor. Am ales sablonul de proiectare de tipul **singleton** deoarece este un model simplu de proiectare deoarece uneori avem nevoie de o singura instanta dintr-o clasa. Acest lucru se poate realiza in mai multe feluri dar eu am ales sa declar o metoda statica din intermediul clasei respective si sa o apelez atunci cand declar un obiect.

6. Sablonul de proiectare comportamental

Sabloanele comportamentale caracterizeaza modul in care obiectele si clasele interactioneaza si isi distribuie responsabilitatile. Am ales sablonul de proiectare **observer**. Sablonul Observer defineste o relatie de dependenta 1..* intre obiecte astfel incat cand un obiect isi schimba starea, toti dependentii lui sunt notificati si actualizati automat. Am ales acest sablon deoarece la fiecare apasare de buton care schimba limba se notifica si se va actualiza interfata grafica.

7. Etapele de dezvoltare

a. Etapa de analiza si proiectare

Pentru a respecta structura de Client-Server se vor face doua aplicatii diferite si fiecare aplicatie va avea diagrame si implementari diferite. Mai jos sunt prezentate diagramele use-case, diagramele de clase, diagramele de activitate si diagramele de secventa.

i. Client

Clientul este cel care utilizeaza aplicatia si o data ce se deschide aplicatia se face conexiunea la server. Daca serverul este inchis, aplicatia se inchide automat.

Diagrama use-case:

In diagrama use-case se prezinta actiunile ce le vor face clientii sau utilizatorii aplicatiei si relatiile dintre actiuni(Figura 1).

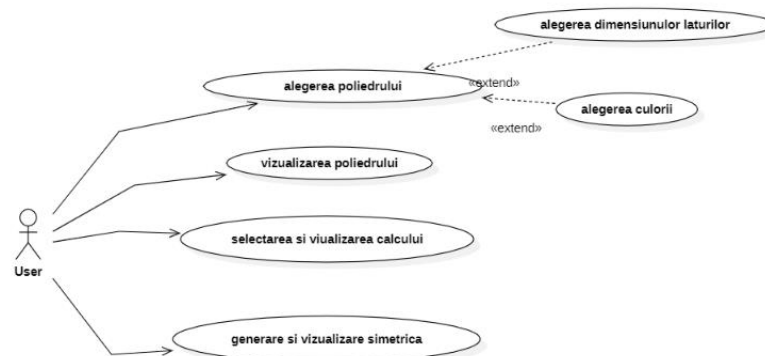


Figura 1

Diagrama de clase

In figura de mai jos este prezentata diagrama de clase a aplicatiei in care am inclus toate pachetele si clasele din interiorul acestora care realizeaza aplicatia. In diagrama de clase sunt prezentate si relatii dintre clase cum ar fi mostenire, agregare, etc.(Figura 2)

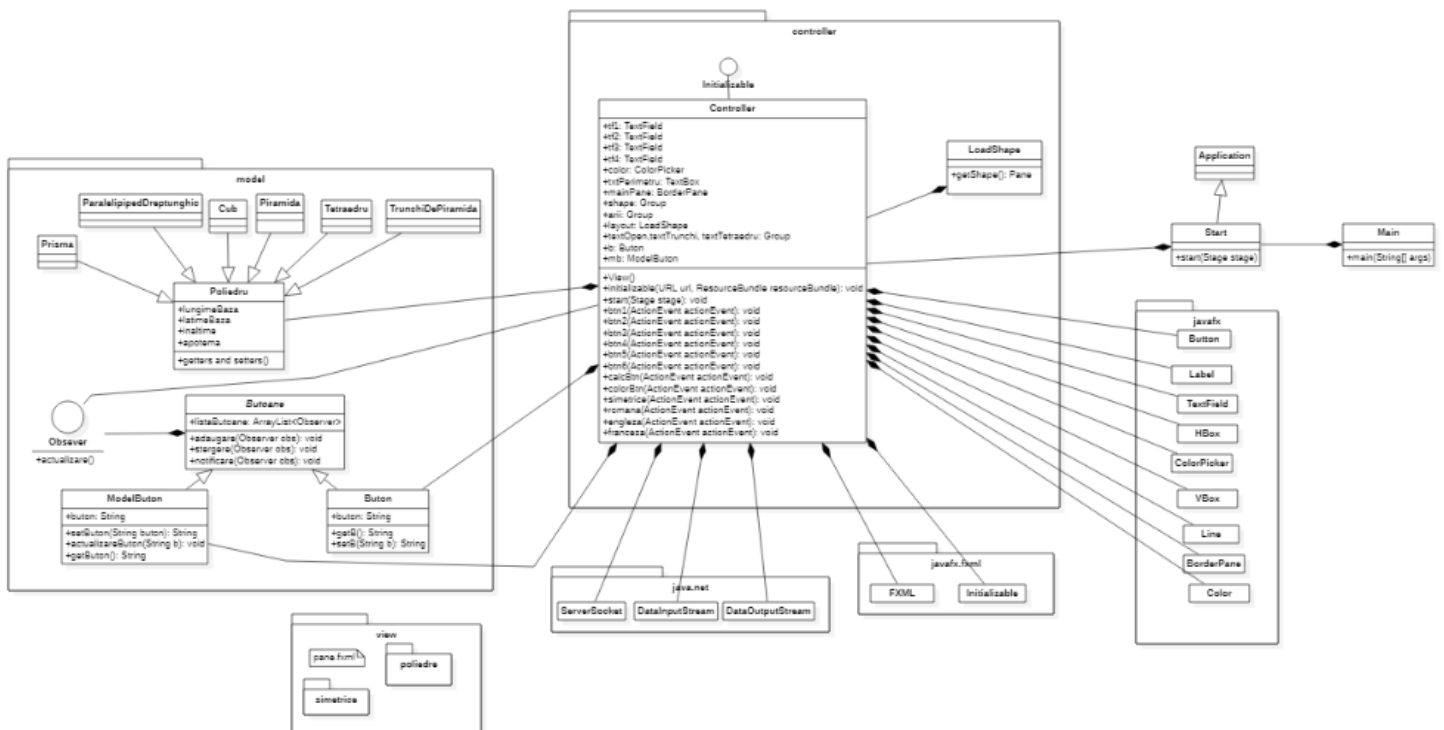


Figura 2

Diagrame de activitate

În diagramele de activitate descriu un flux procedural. Diagramele de activități se pot reprezenta pentru a descrie un caz de utilizare, un pachet sau implementarea unei operații. De exemplu Figura 3 reprezintă alegerea poliedrului

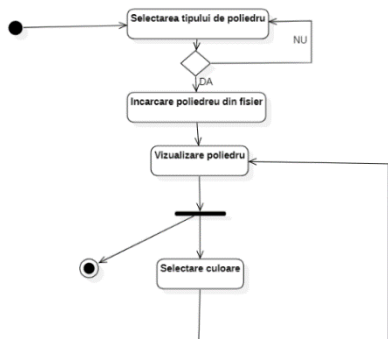


Figura 3(alegere poliedrului)

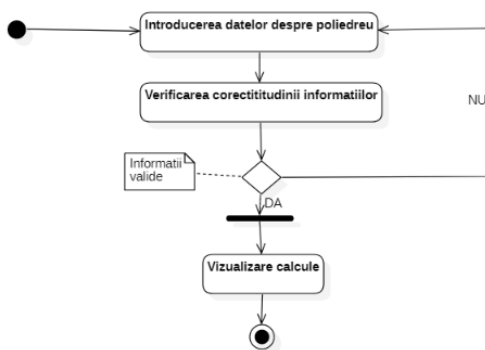


Figura 4(Selectarea calculului)

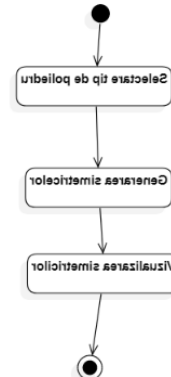


Figura 5(Generarea simetrice)

Diagrame de secvență

Diagramele de interacțiuni sunt folosite pentru a modela comportamentul unei mulțimi de obiecte dintr-un anumit context care interacționează în vederea îndeplinirii unui anumit scop. Scopul specifică modul în care se realizează o operație sau un caz de utilizare.

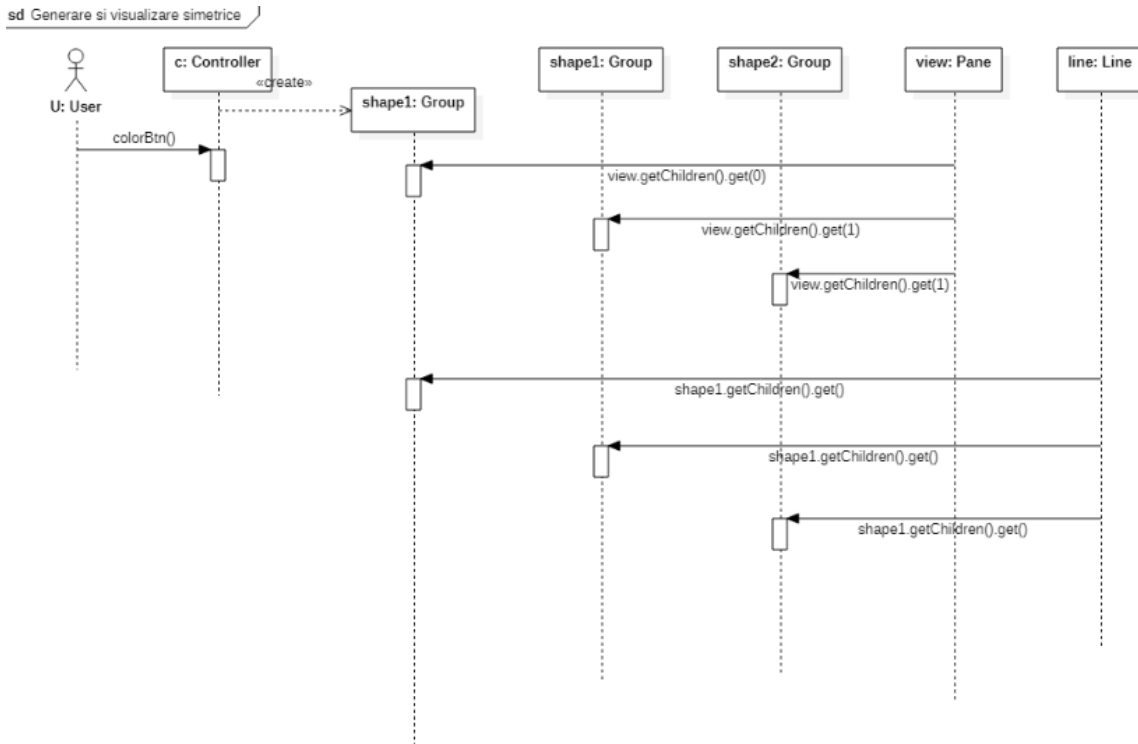


Figura 6(Generarea simetrice)

sd Selectarea si vizualizarea calcul

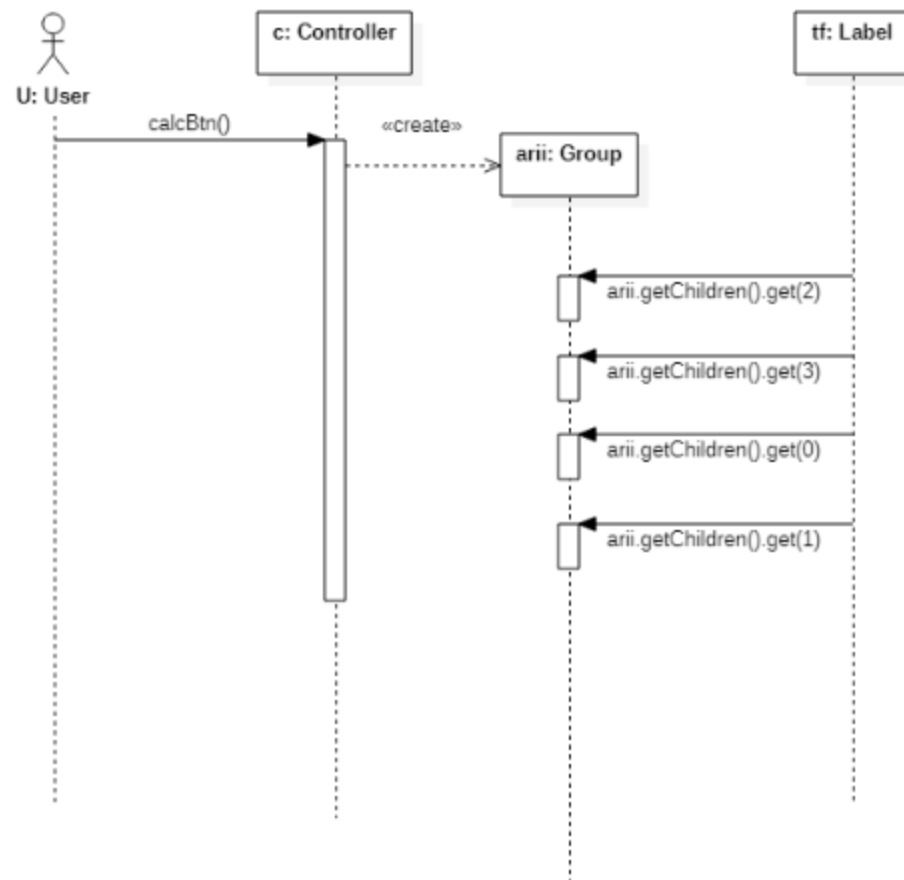


Figura 7(Selectarea calcului)

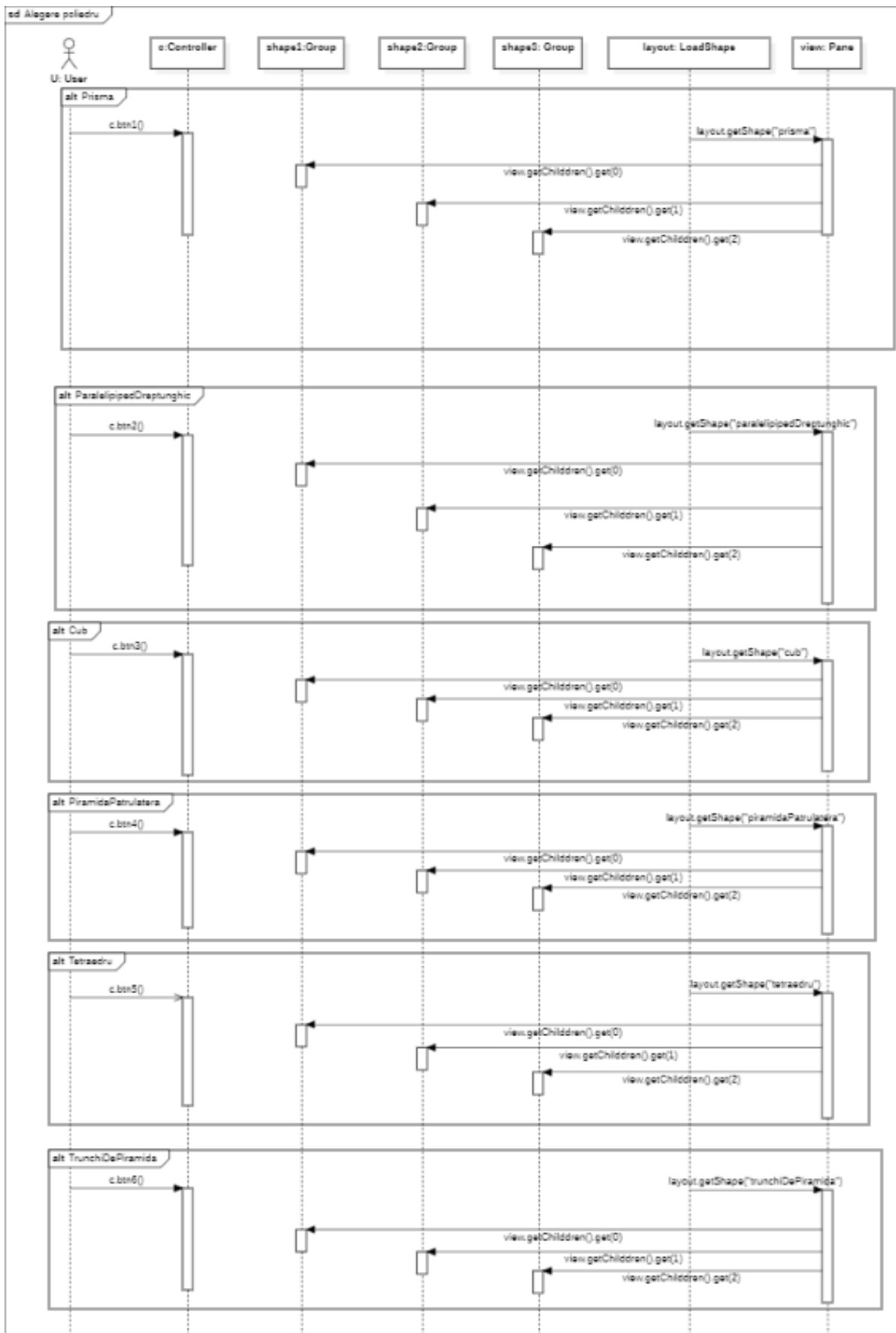


Figura 7(alegerea poliedrului)

ii. Server

Serverul este o aplicatie separata care creaza un host pentru a se putea conecta mai multi clienti pe baza acestuia. In server se primesc datele de la clienti si se fac operatii cu acestea pentru calculul ariilor si volumului poliedrului ales. Dupa calcularea acestor date se trimite rezultatele inapoi la client.

Diagrama use-case:

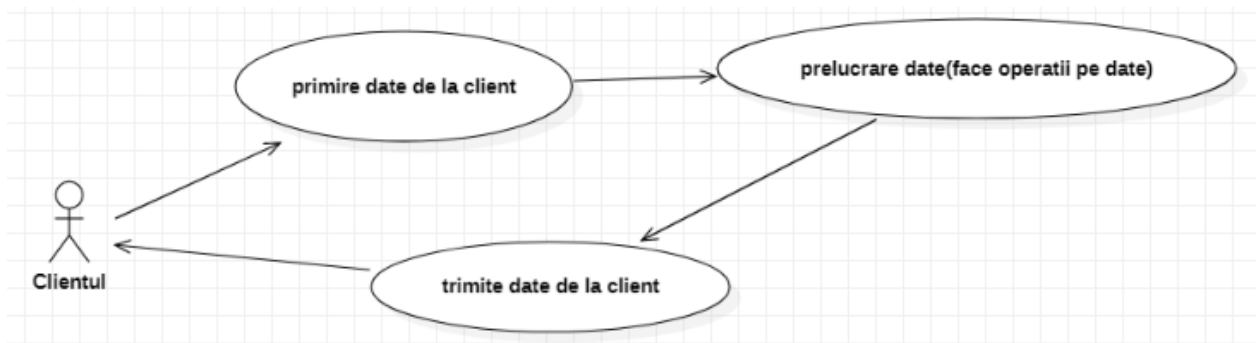


Figura 8

Diagrama de clase

In figura de mai jos este prezentata diagrama de clase a aplicatiei in care am inclus toate pachetele si clasele din interiorul acestora care realizeaza aplicatia.

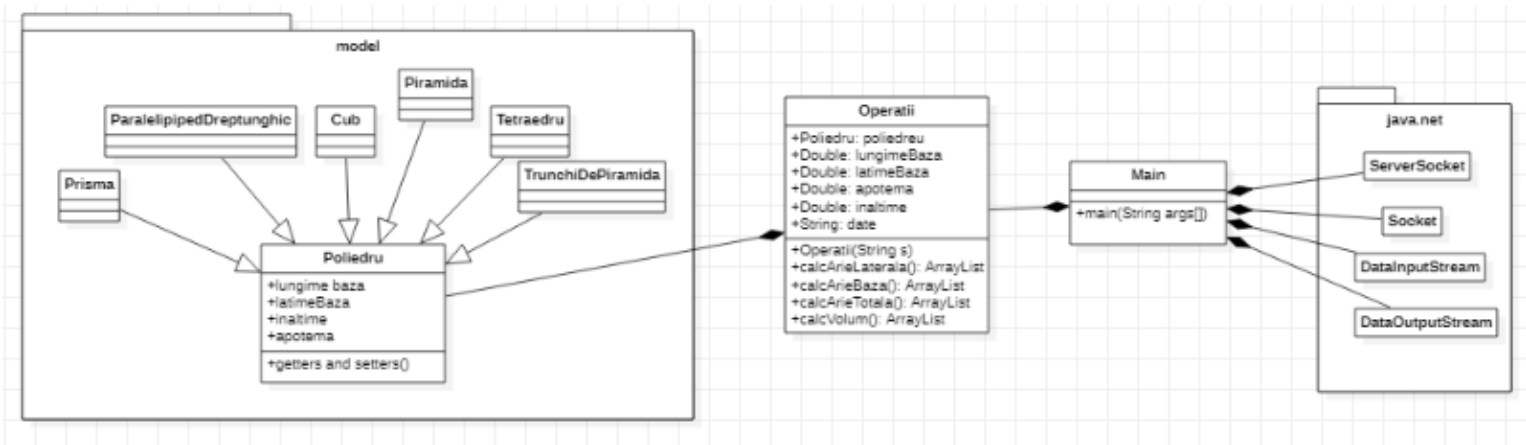


Figura 9

Diagrame de activitate

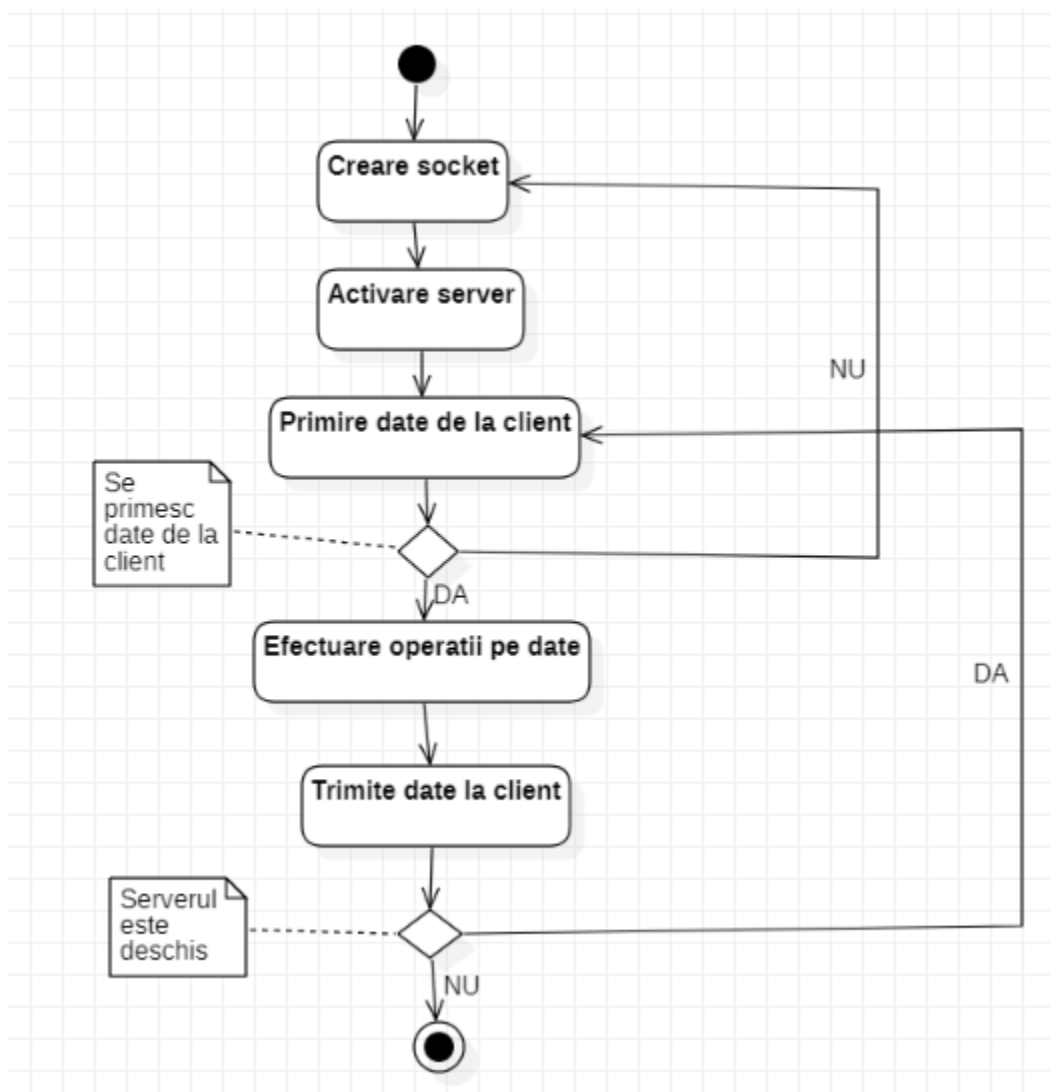


Figura 10

Diagrame de secventa

Diagramele de interactiuni sunt folosite pentru a modela comportamentul unei multimi de obiecte dintr-un anumit context care interactioneaza in vederea indeplinirii unui anumit scop.

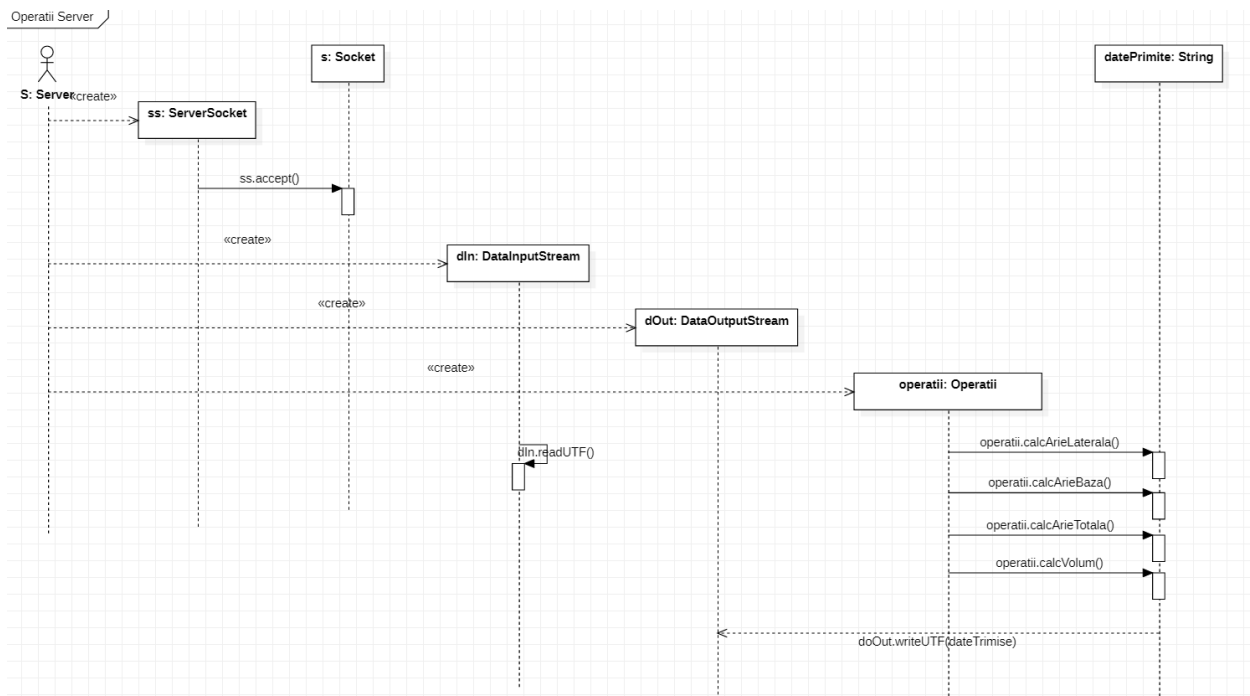


Figura 11

b. Etapa de implementare

La deschiderea aplicatiei se va vedea interfata grafica in care se poate observa informatii teoretice despre poliedre regulate. In partea de sus se vor afla butoanele care selecteaza ce tip de poliedru sa se afiseze. In partea stanga sunt campurile in care se vor introduce informatiile despre poliedru si butoanele care executa diferite operatii. La deschiderea aplicatiei, clientul se conecteaza automat la Server si in caz in care nu este deschis serverul, va aparea un mesaj respectiv.

```

E:\Java\JDK\bin\java.exe --module-path D:\1facultatie\AN3\sem2\PS\javafx-sdk-16\lib --add-modules javafx.controls,javafx.fxml "
Serverul nu este deschis

Process finished with exit code 0
  
```

Figura 12

In cazul in care serverul este deschis, va apararea mesajul de mai jos in consola si se deschide aplicatia(Figura 14)

```

E:\Java\JDK\bin\java.exe --module-path D:\1facultatie\AN3\sem2\PS\javafx-sdk-16\lib --add-modules javafx.controls,javafx.fxml
Clientul s-a conectat la server
  
```

Figura 13



Figura 14

Dupa ce se introduc datele despre poliedre(lungime baza, latimea bazei, apotema si inaltime) si se apasa butonu „Calculeaza”, se va afisa in consola ce date se primesc de la server(Figura 15)dupa calculele facute de acesta, iar in consola de la server se vor afisa datele ce s-au primit se la client(Figura 16). Aceste date se vor afisa in interfata grafica de la aplicatie in dreptul fiecarui parametru(Figura17).

```
E:\Java\JDK\bin\java.exe --module-path D:\1facultatie\AN3\sem2\PS\javafx-sdk-10\lib --add-modules javafx.controls,javafx.fxml
Clientul s-a conectat la server
Datele primite sunt: 24.0 32.0 48.0 1.73 4.0 10.379999999999999 27.46 40.0 68.75999999999999 6.92 16.0 41.519999999999996
```

Figura 15

```
E:\Java\JDK\bin\java.exe "-javaagent:E:\IntelliJ IDEA Community Edition 2020.2.3\lib\idea_rt.jar=65135:E:\IntelliJ IDEA Community Edition 2020.2.3\bin" -jar D:\1facultatie\AN3\sem2\PS\javafx-sdk-10\lib\javafx-10.jar
S-a deschis serverul
Client conectat
Datele primite sunt: Prisma 2 1 3 4
```

Figura 16

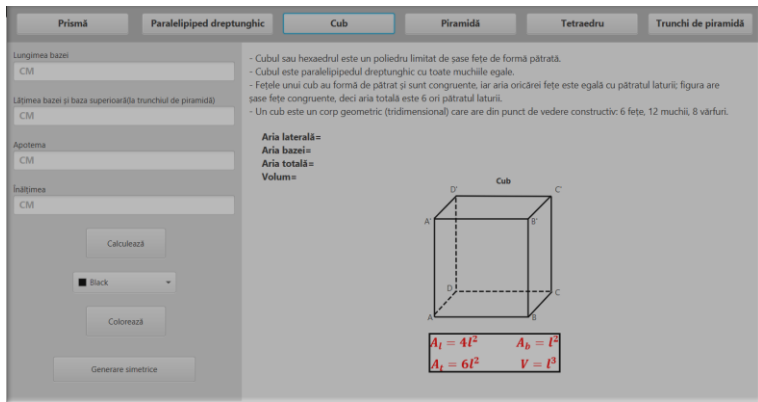


Figura 12

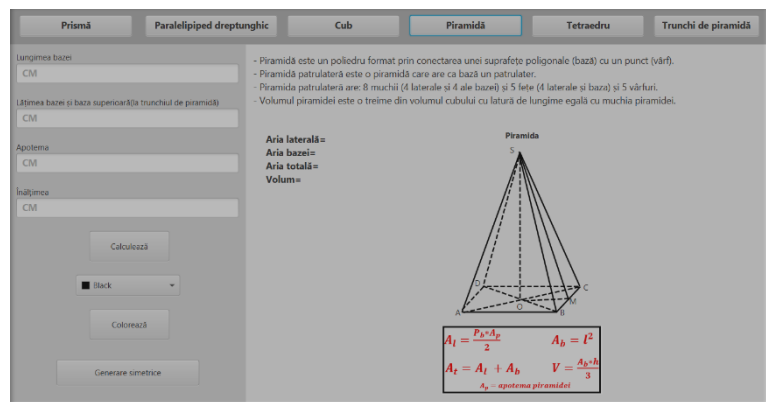


Figura 13

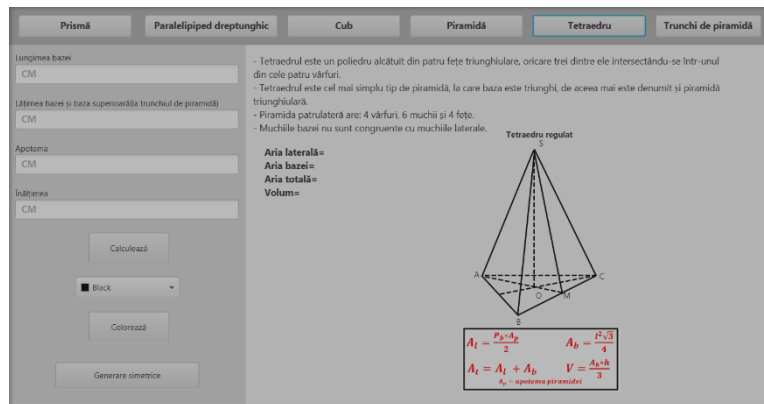


Figura 14

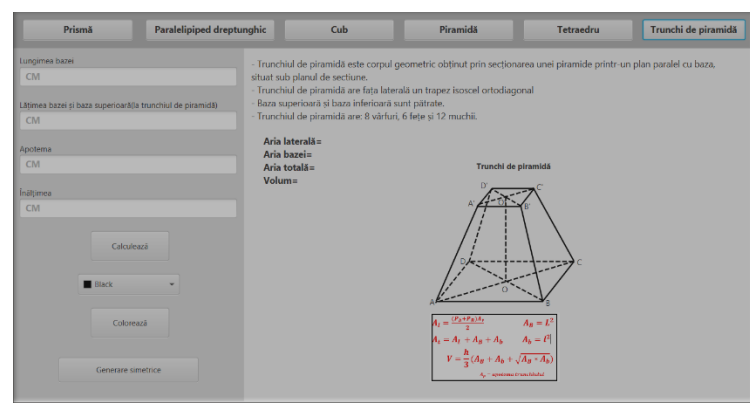


Figura 15

Dupa ce se vor introduce datele datele pentru poliedre, odata la apasarea butonului se vor afisa calculele facute pe baza informatiilor introduse(aria laterala, aria bazei, aria totala si volumul). Pe langa acest lucru, se poate selecta si culoarea pe care o poate lua poliedrul. Acesta se coloreaza o data la apasarea butonului

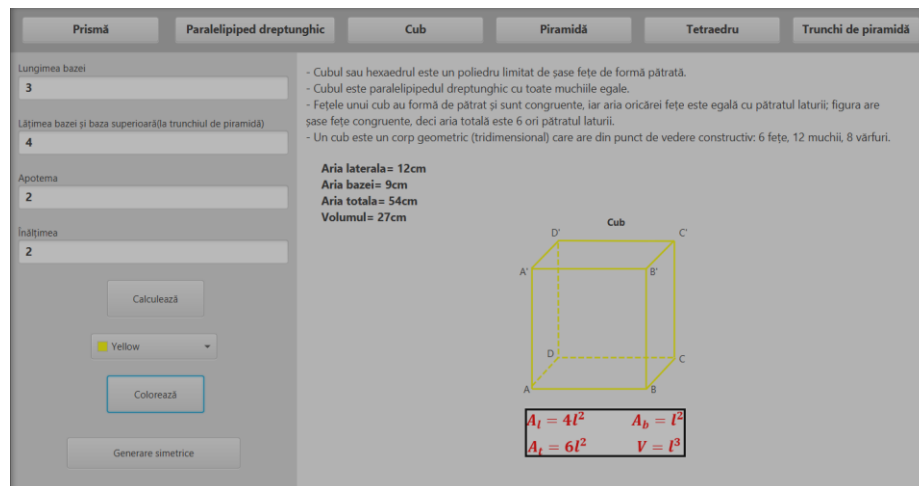


Figura 16

În funcție de poliedrul selectat, o dată ce se apasă butonul de „Generare simetrică” se va deschide un Panel nou în care se va vedea simetrica.

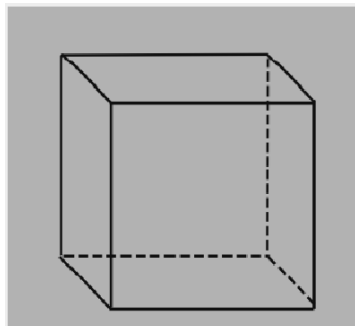


Figura 17

8. Webografie

<https://www.youtube.com/>

<https://openjfx.io/>

<https://www.w3schools.com/>