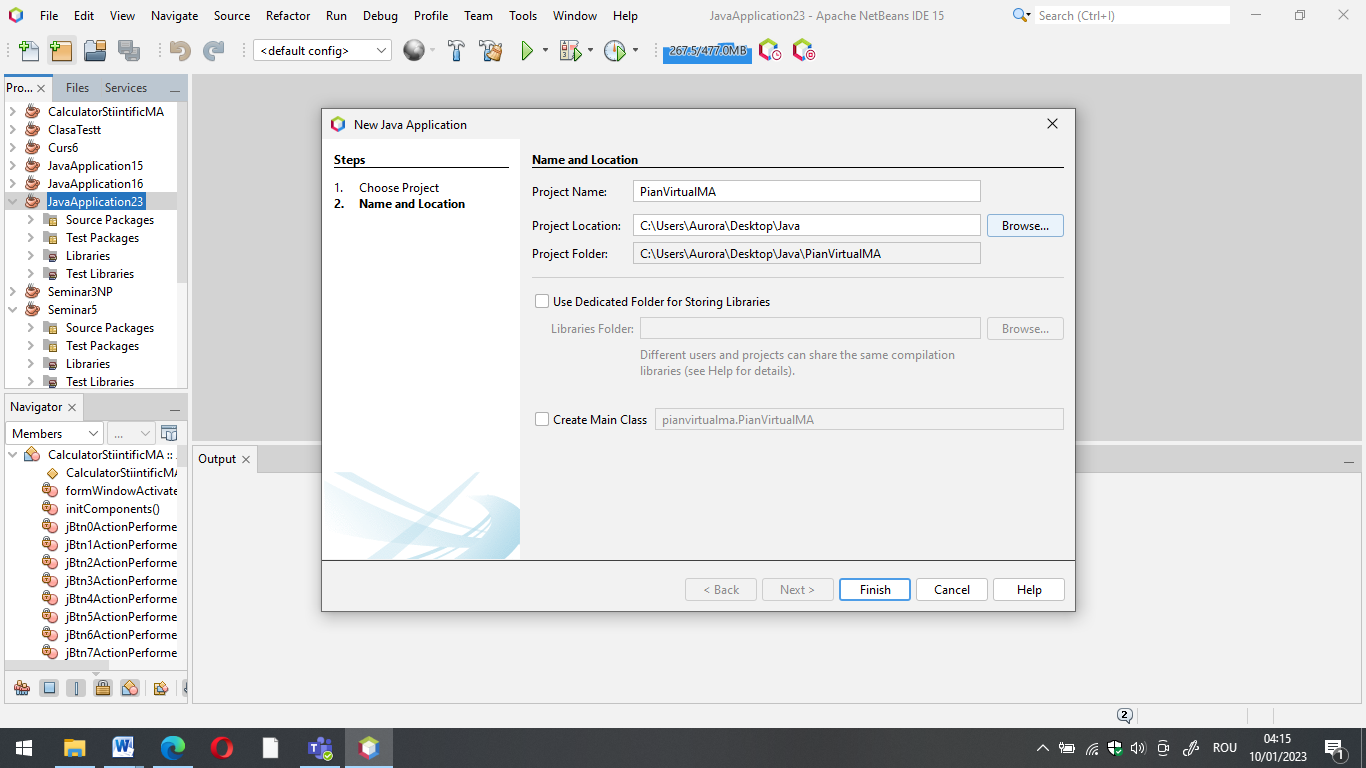
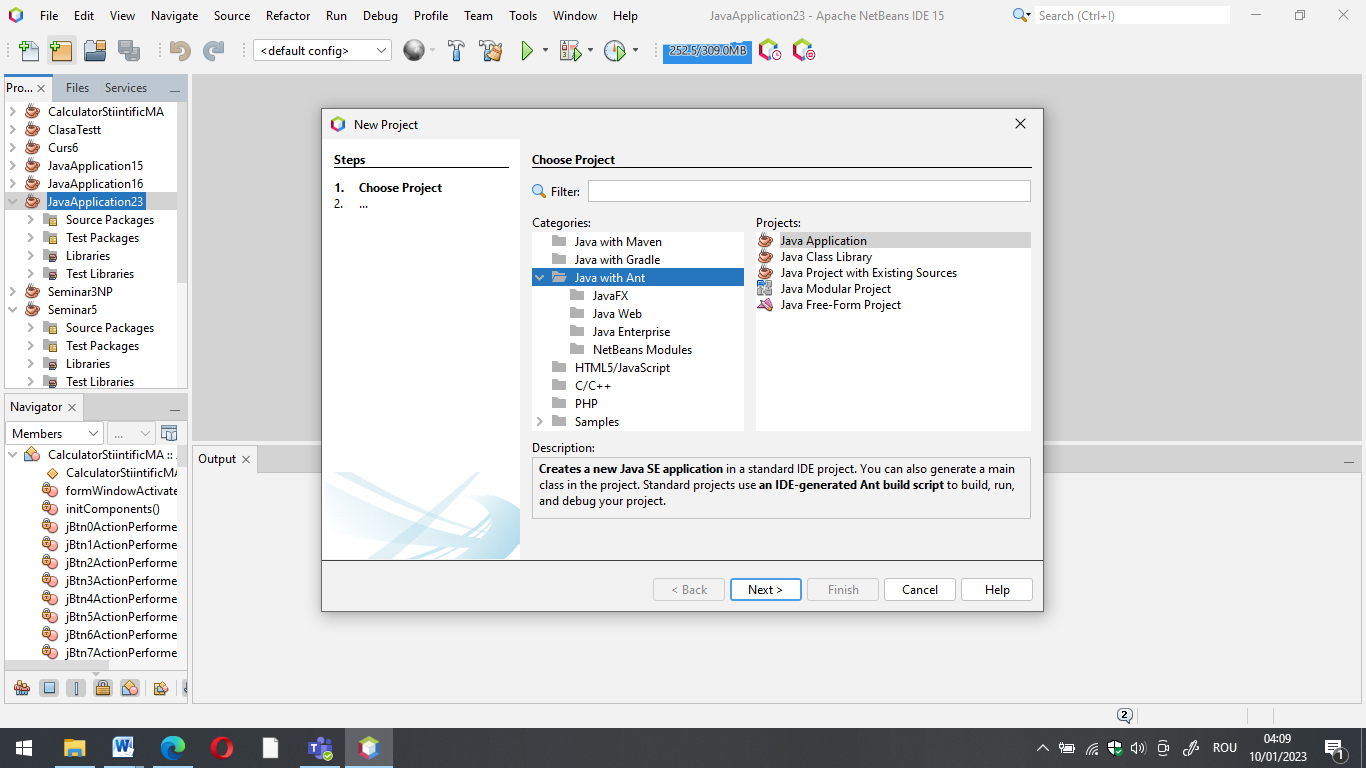
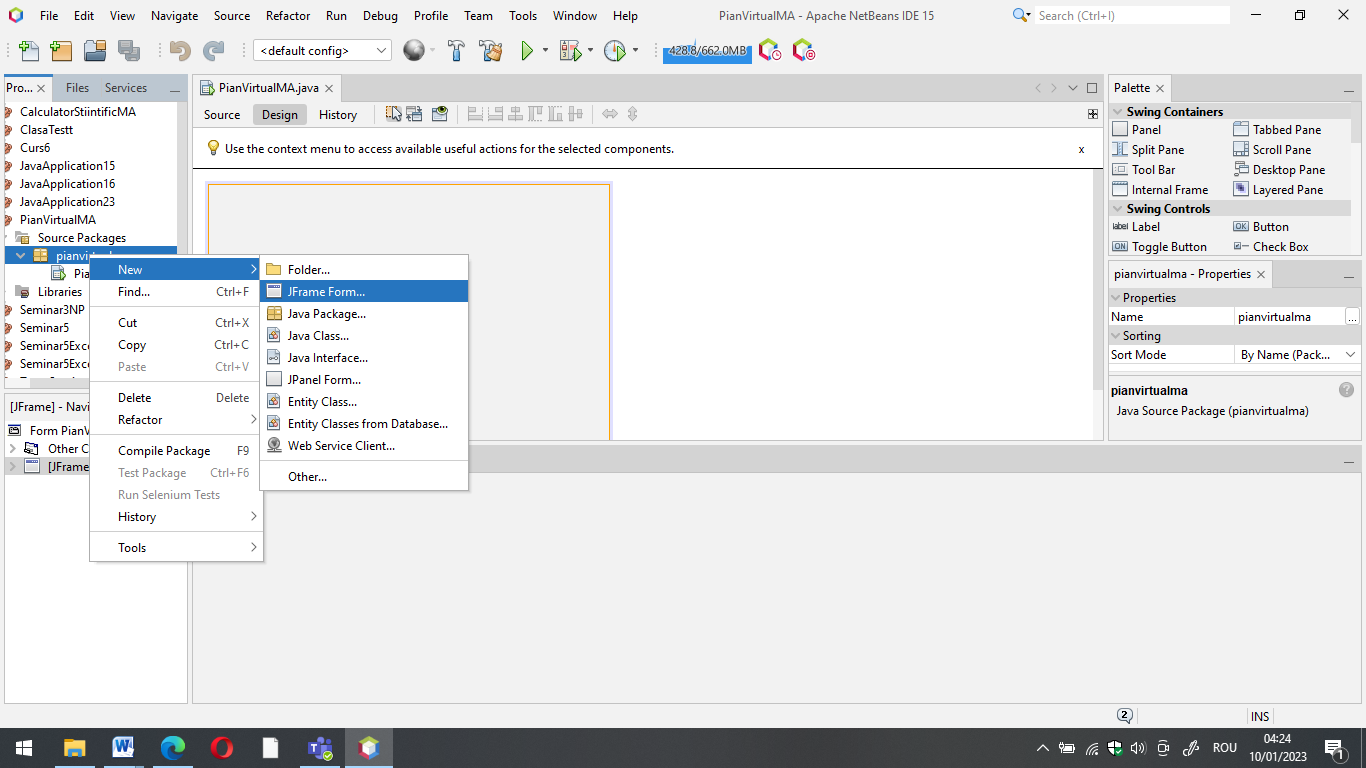
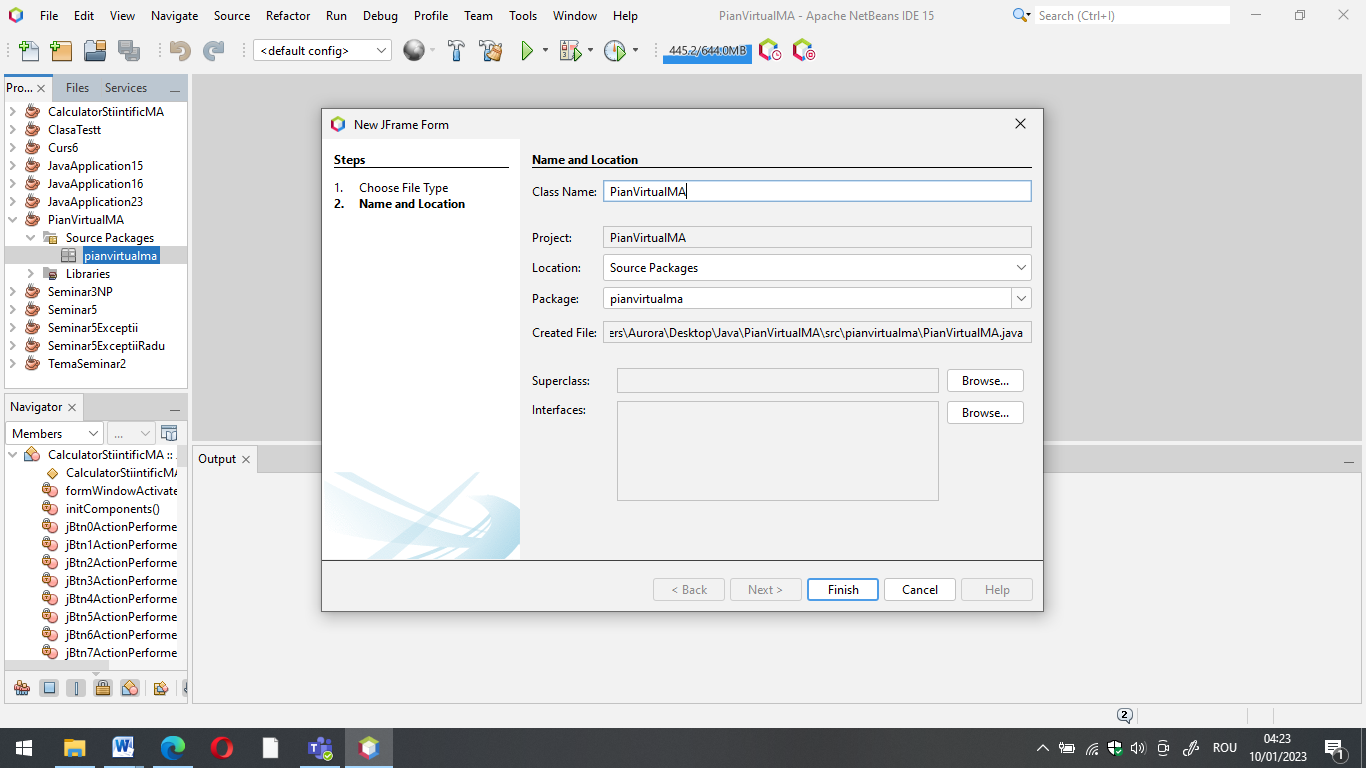
**SEMINAR 7 – PIAN VIRTUAL**

1. **DESIGN**

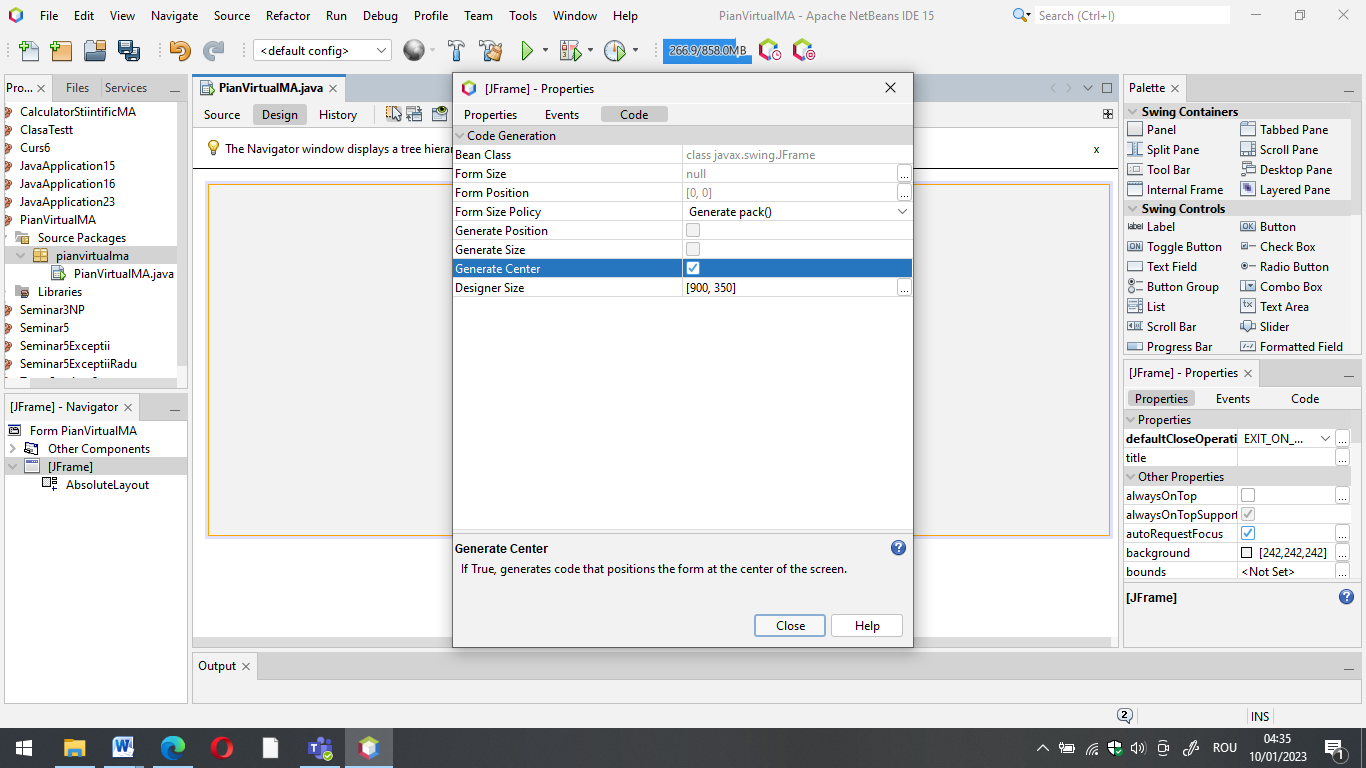
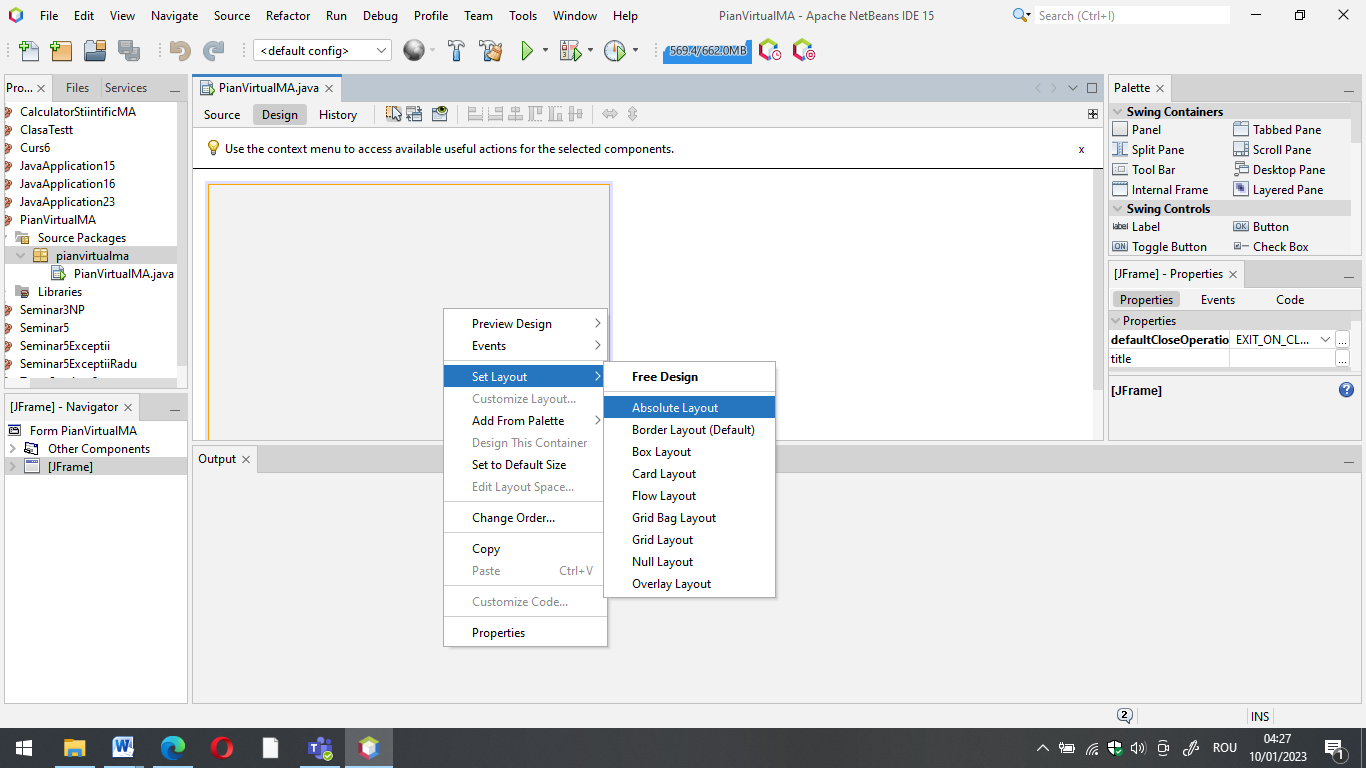
* Vom crea acest proiect folosind Apache Netbeans ca IDE
* Creăm un nou proiect PianVirtualN(ume)P(renume) cu setările din stânga la Choose Project și cele din dreapta la Name and Location. Atenție! Create Main Class debifat!



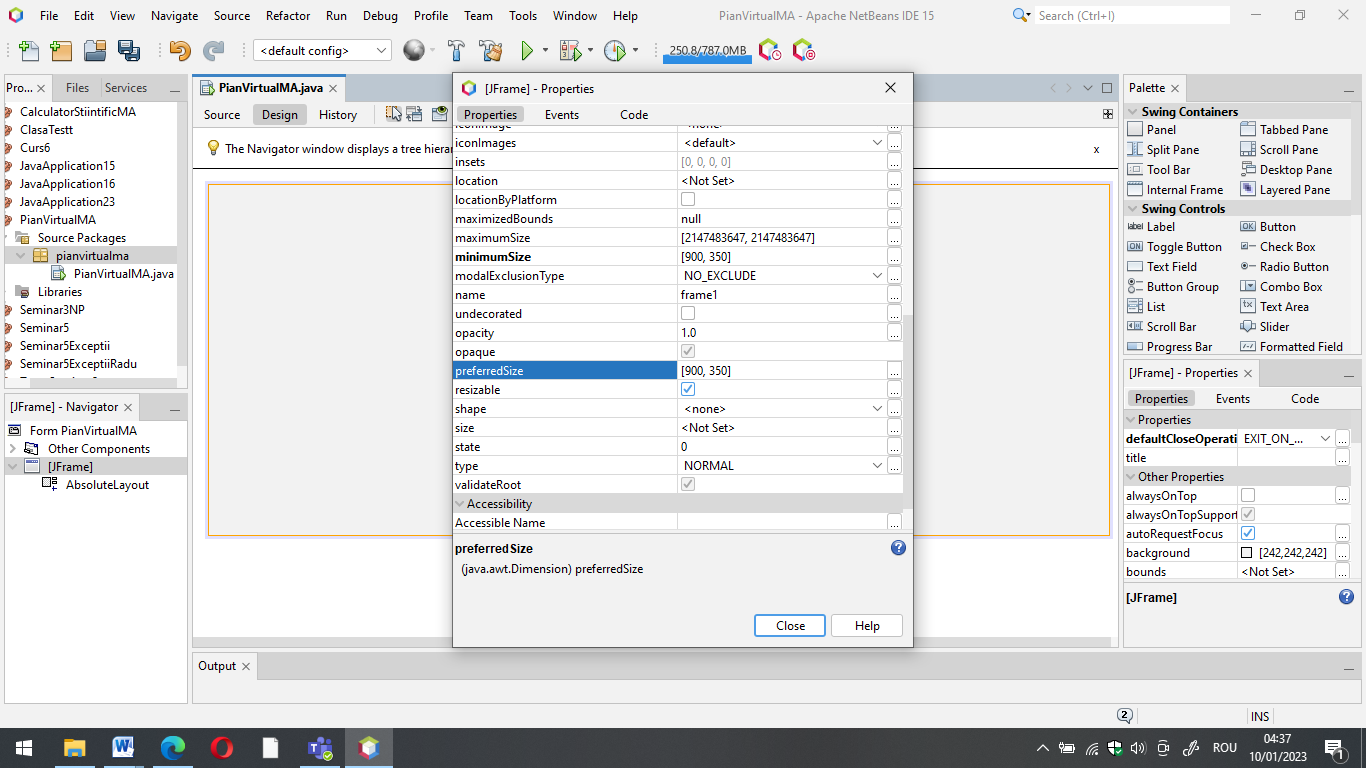
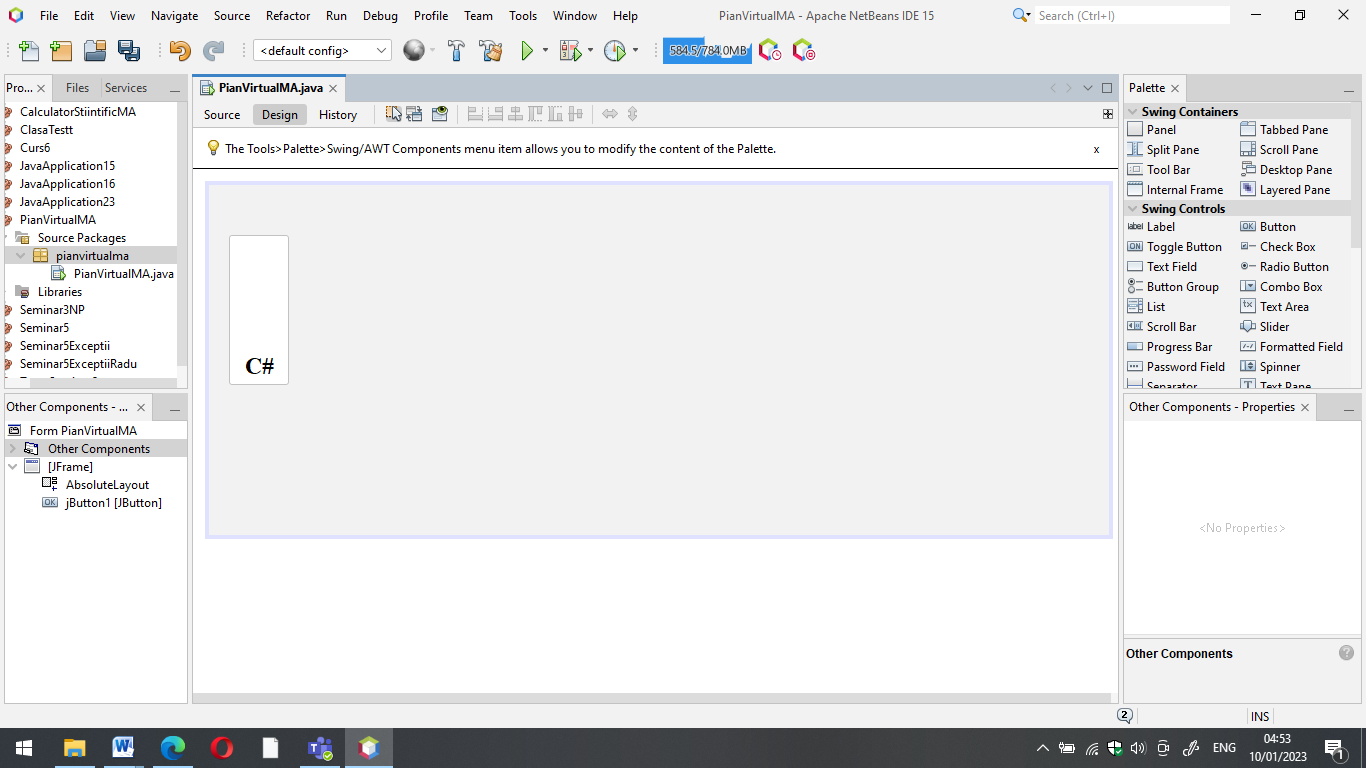
* Click-dreapta pe numele proiectului în browserul din stânga și alegem New → Java Package cu numele pianvirtualn(ume)p(renume)
* Click-dreapta pe numele pachetului în browserul din stânga și alegem New → JFrame Form cu numele clasei PianVirtualN(ume)P(renume)

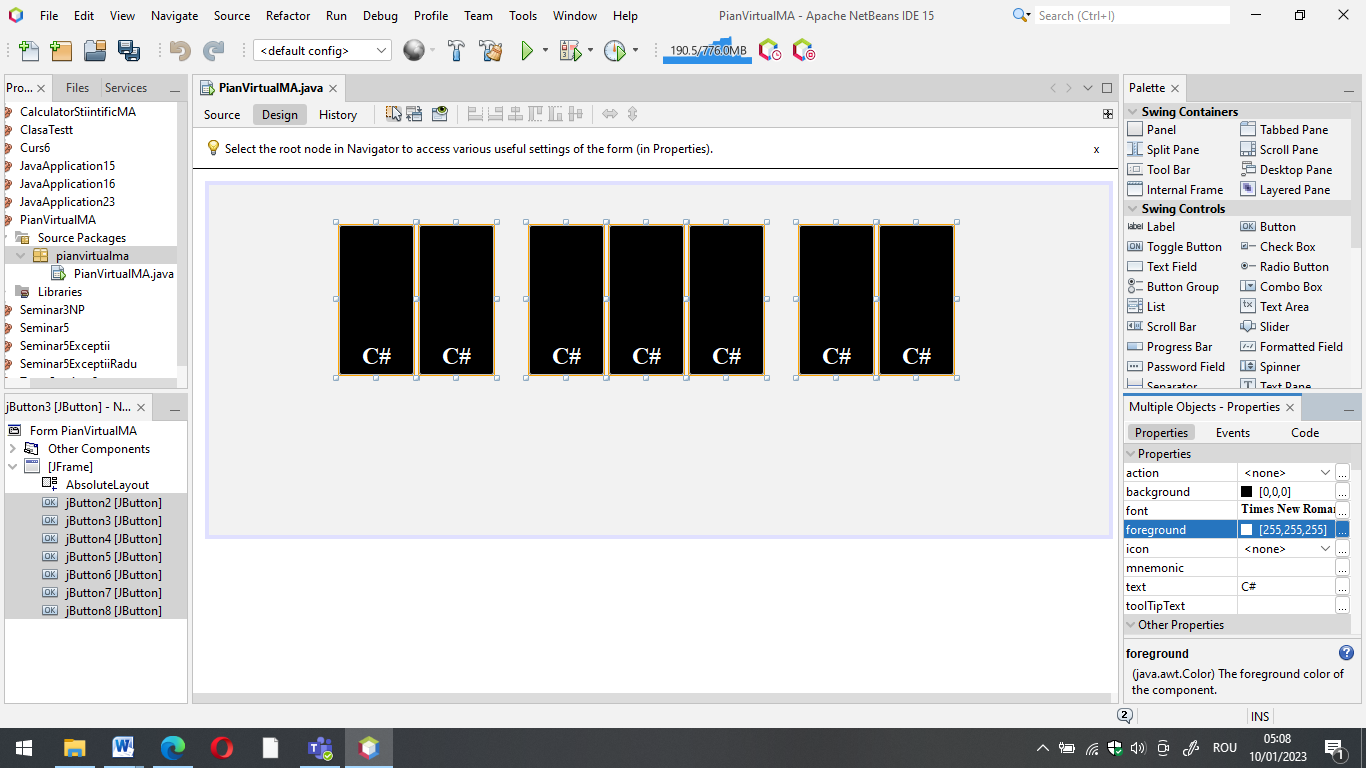
* Click-dreapta pe dreptunghiul formei → Set Layout → Absolute Layout. Tot click-dreapta pe dreptunghiul formei → Properties → Code → Designer Size 900x350 și Generate Center bifat:



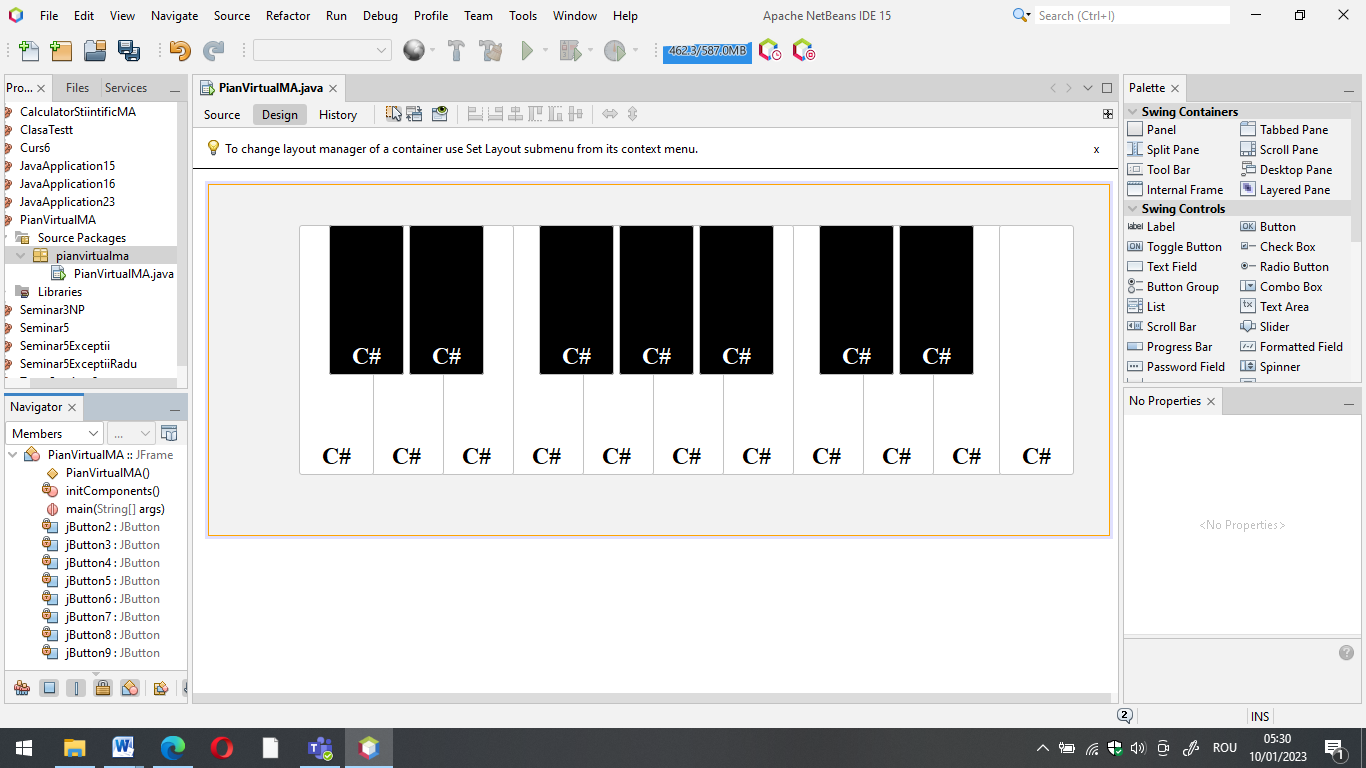
* Tot click-dreapta pe dreptunghiul formei → Properties → Properties și setăm minimumSize și preferredSize tot la 900x350

* Din Palette din dreapta introducem un Button. Formatăm fontul de pe el la Times New Roman – bold, de mărime 24. Click-dreapta pe buton → Edit text și introducem C# (adică Do diez, deci va fi o clapă neagră a pianului). Click-dreapta pe buton → Properties → verticalAllignement = BUTTOM și width=75, height=150. După aceste modificări, butonul arată ca în figura de sus – dreapta
* Mai facem încă 6 copii ale butonului pe care le organizăm în felul următor. Selectăm toate cele 7 butoane și schimbăm culoarea fontului foreground=255,255,255 (alb) și background=0,0,0 (negru).

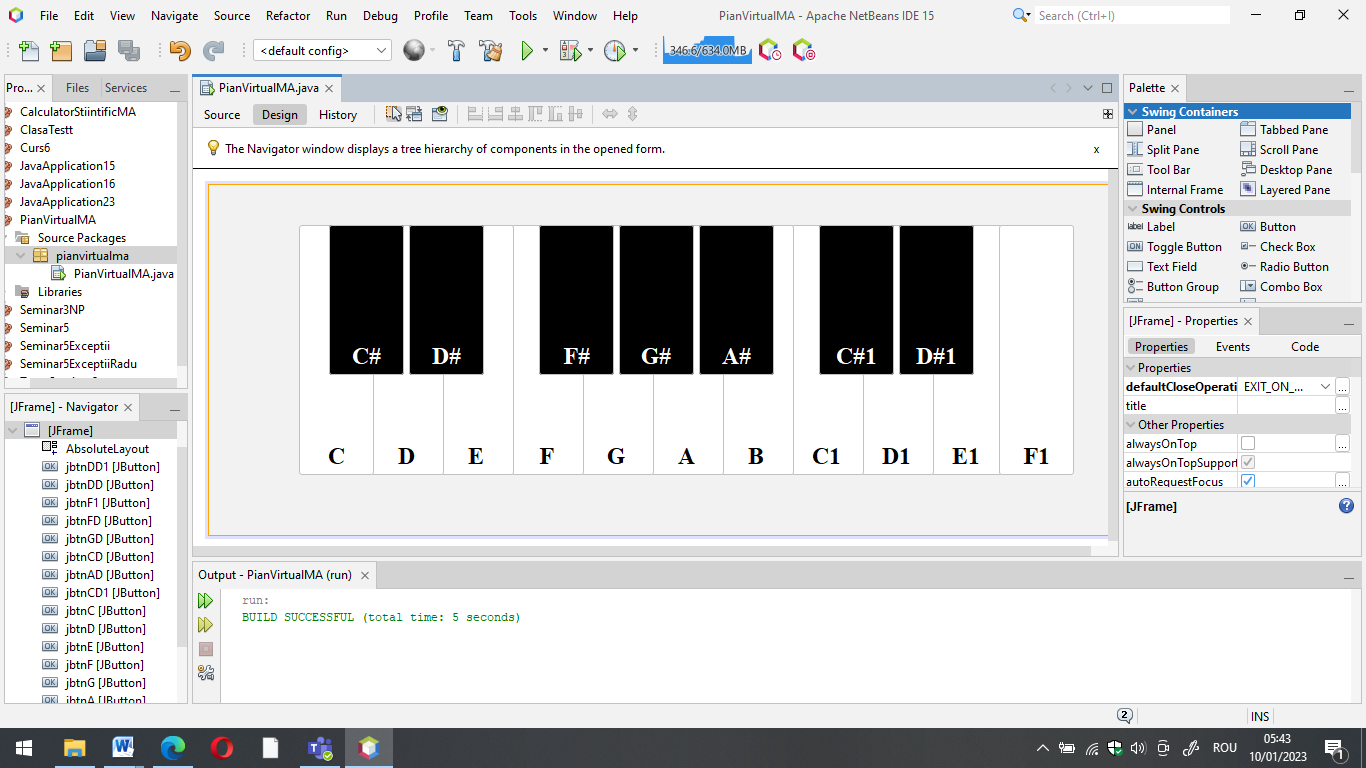


* Mai facem încă o copie a butonului, schimbăm height=250, culoarea fontului foreground=0,0,0 (negru) și background=255,255,255 (alb). Aceasta va fi una dintre clapele albe. Multiplicăm clapa alba astfel încât să avem în total 11 clape albe puse una lângă alta. Configurația finală:



* Click-dreapta pe fiecare buton și Edit Text astfel încât să avem următoarele denumiri vizibile pe fiecare clapă. De asemenea, click-dreapta și Change Variable Name pe fiecare clapă astfel încât să avem următoarele asocieri de clape și variabile:

|  |  |  |  |
| --- | --- | --- | --- |
| **CLAPĂ** | **NUME VARIABILĂ** | **CLAPĂ** | **NUME VARIABILĂ** |
| **C** | **jbtnC** | **A** | **jbtnA** |
| **C#** | **jbtnCD** | **A#** | **jbtnAD** |
| **D** | **jbtnD** | **B** | **jbtnB** |
| **D#** | **jbtnDD** | **C1** | **jbtnC1** |
| **E** | **jbtnE** | **C#1** | **jbtnCD1** |
| **F** | **jbtnF** | **D1** | **jbtnD1** |
| **F#** | **jbtnFD** | **D#1** | **jbtnDD1** |
| **G** | **jbtnG** | **E1** | **jbtnE1** |
| **G#** | **jbtnGD** | **F1** | **jbtnF1** |



1. **COD**

* Mergem în cod/Source și imediat după prima linie package pianvirtualnp; punem următoarele importări de pachete/clase:

|  |
| --- |
| import java.io.File;  import java.io.IOException;  import java.util.logging.Level;  import java.util.logging.Logger;  import javax.sound.sampled.AudioInputStream;  import javax.sound.sampled.AudioSystem;  import javax.sound.sampled.Clip;  import javax.sound.sampled.LineUnavailableException;  import javax.sound.sampled.UnsupportedAudioFileException; |

* Vom pune două tipuri de instrumente/tonuri pe pian: de pian și de vioară. Punem în folderul proiectului folderele Note-pian și Note-vioară conținând nótele Do-Si și diezii asociați pentru fiecare din cele două instrumente.
* În folderul Note-pian, click pe fișierul Pian-Cdiez.wav, ținem Shift apăsat și apoi click-dreapta și alegem Copy as Path.
* Imediat după constructorul fără parametri al clasei PianVirtualNP,

|  |
| --- |
| public PianVirtualNP() {  initComponents();  } |

adăugăm cod pentru următoarea metodă:

|  |
| --- |
| private void notePianoCD() throws UnsupportedAudioFileException, IOException, LineUnavailableException{  //incercati fara throws si observati ca obtineti erori de compilare in fata liniilor de cod din mijloc  //una dintre recomandarile de pe bulina rosie de eroare din fata liniei fiind Add throws clause pentru  //anumite pachete/clase…bifati pe aceste recomandari si observati ca se adauga automat declaratia throws  //dupa semnatura metodei si se rezolva progresiv erorile  File file = new File("C:\\Users\\Aurora\\Desktop\\Java\\PianVirtualMA\\Note-pian\\Pian-Cdiez.wav");  //punem calea copiata intre ghilimele si avem grija ca toate separatoarele sa fie \\  AudioInputStream audioStream = AudioSystem.getAudioInputStream(file);  Clip clip = AudioSystem.getClip();  clip.open(audioStream);  clip.start();  } |

* Ne mutăm în Design și cu click-dreapta pe clapa C#, alegem Events → Action → actionPerformed și în interiorul metodei jbtnCDActionPerformed introduceți linia de cod notePianoCD(); și observați că generează eroare de compilare. Selectați linia de cod problematică și observați că una dintre recomandări este Surround with try-catch. Dublu-click pe această recomandare, observați că s-a adăugat câte o ramură catch pentru fiecare excepție declarată după throws și s-a rezolvat eroarea de compilare.
* Run, apăsați pe clapa C# și ar trebui să se audă nota respectivă cântată.
* Construiți câte o metodă notePiano… pentru fiecare notă a unei octave, atât naturală, cât și cu diez. Copiați blocul try-catch construit pentru C#, mergeți în Design pe fiecare dintre clapele din prima octavă, click-dreapta → Events → Action → actionPerformed și lipiți blocul try-catch copiat și modificați corespunzător doar numele metodei din blocul try.

Bibliografie:

* [How to Make a Piano with Java NetBeans using Functions - YouTube](https://www.youtube.com/watch?v=nDDbr17vLgw)
* [How to Create Virtual Piano in Java NetBeans - YouTube](https://www.youtube.com/watch?v=l8dfU7SnjcM)