

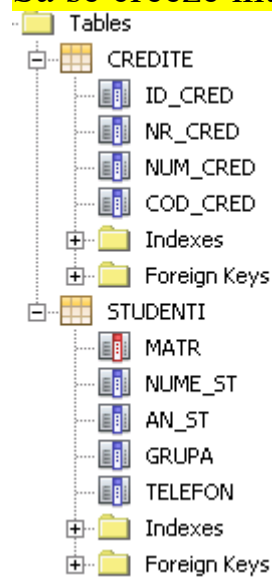
Sa se lucreze tutorialele

1. <https://netbeans.apache.org/tutorial/main/kb/docs/ide/>
2. <https://www.jetbrains.com/help/idea/relational-databases.html#first-steps>


Sa se creeze structura unui tabel in modul programat, vezi codul sursa :

```
import java.sql.*; public class apl1 {  
    public static void main(String args[]) {  
        String url = "jdbc:derby://localhost:1527/student;create=true";  
        Connection con;  
        String createString;  
        createString = "create table EXAMEN (Nume_Curs varchar(30), Ora varchar(30), Zi int, Luna  
int, An int, Tip varchar(30), Locatie varchar(30), persoana_Examinata varchar(30))";  
        Statement stmt;  
        try {  
            Class.forName("org.apache.derby.jdbc.ClientDriver");  
        } catch (java.lang.ClassNotFoundException e) {  
            System.err.print("ClassNotFoundException: ");  
            System.err.println(e.getMessage());  
        }  
        try { con = DriverManager.getConnection(url, "root", "123");  
            stmt = con.createStatement();  
            stmt.executeUpdate(createString); stmt.close(); con.close();  
        }  
        catch (SQLException ex) {  
            System.err.println("SQLException: " + ex.toString());  
        }  
    }  
}
```

Sa se creeze interfata pentru definirea structurii tabelelor

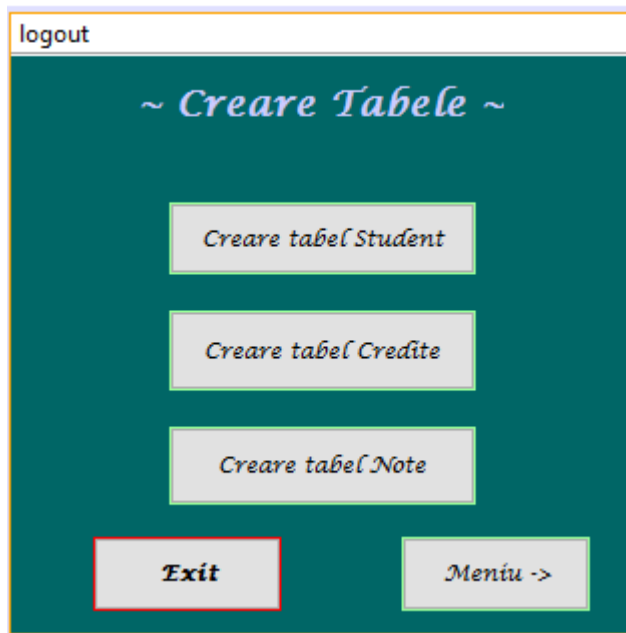


Login cu user si parola



```
private void jButton3ActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    String password;
    String user;
    user = username.getText();
    password = pass.getText();
    if(password.equals("student") && user.equals("Parola@RD!@"))
    {
        new meniu().setVisible(true);
        this.dispose();
    }
    else{
        mesaj.setText("Please try again.");
    }
}
```

Creare structura tabele



```
private void tabeStudentActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    Connection con;
    try{
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        con =
        DriverManager.getConnection("jdbc:derby://localhost:1527/Test_curs
_2020;create=true;user=student;password=Parola@RD!@");
        Statement stmt = con.createStatement();
        stmt.executeUpdate("CREATE TABLE student" +
            "(Nr_matricol INTEGER ," + "Nume_stud
            VARCHAR(50)," +
            "AN VARCHAR(4)," +
            "Grupa VARCHAR(4)," + "Telefon VARCHAR(11))");
    }
    catch(ClassNotFoundException e)
    {
        JOptionPane.showMessageDialog(rootPane, e.toString(),"Mesaj
        Error...", JOptionPane.INFORMATION_MESSAGE);
    }
    catch(SQLException e)
```

```

    {
        JOptionPane.showMessageDialog(rootPane, e.toString(), "Mesaj
Error...", JOptionPane.INFORMATION_MESSAGE);
    }
}

```

```

private void
tabelCrediteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Connection con;
    try{
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        con =
DriverManager.getConnection("jdbc:derby://localhost:1527/Test_curs
_2020;create=true;user=student;password=Parola@RD!(@");
        Statement stmt = con.createStatement();
        stmt.executeUpdate("CREATE TABLE credite" +
            "(id_cred INTEGER ,"+"num_cred VARCHAR(50)," +
"nr_cred VARCHAR(5)," +
            "cod_cred VARCHAR(50))");
    }
    catch(ClassNotFoundException e)
    {
        JOptionPane.showMessageDialog(rootPane, e.toString(), "Mesaj
Error...", JOptionPane.INFORMATION_MESSAGE);
    }
    catch(SQLException e)
    {
        JOptionPane.showMessageDialog(rootPane, e.toString(), "Mesaj
Error...", JOptionPane.INFORMATION_MESSAGE);
    }
}
}

```

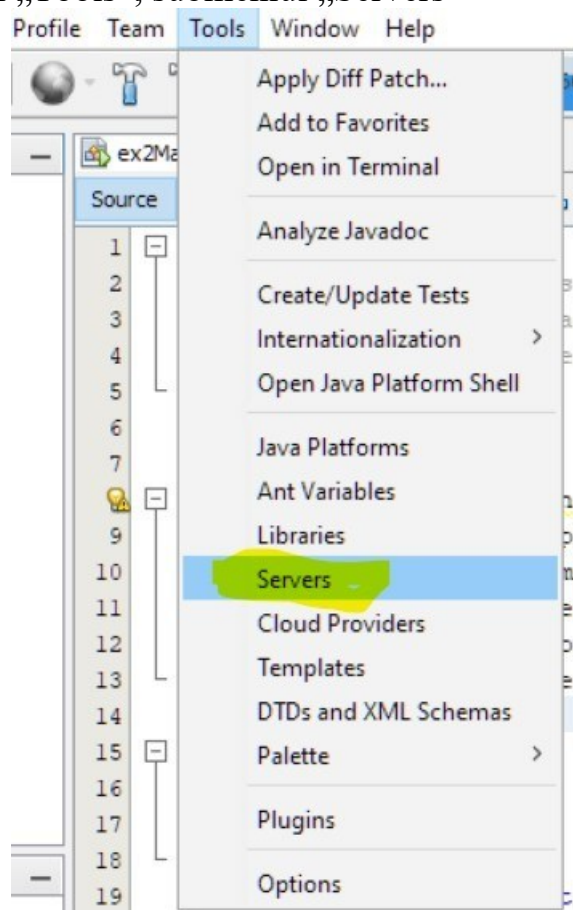
Configurarea NetBeans IDE 12 pentru JavaDB

Pentru activarea caracteristicii JavaDB pe NetBeans IDE v. 12, este necesară instalarea unei instanțe de server GlassFish. Acest lucru este destul de simplu, dacă este realizat cu ajutorul instrumentelor integrate în NetBeans.

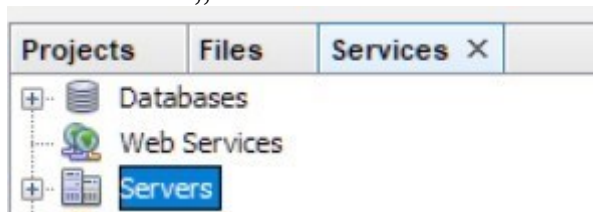
JavaDB este versiunea suportată de compania Oracle a serverului Apache GlassFish.

Instalarea acestui component este posibilă prin intermediul a două căi de acces:

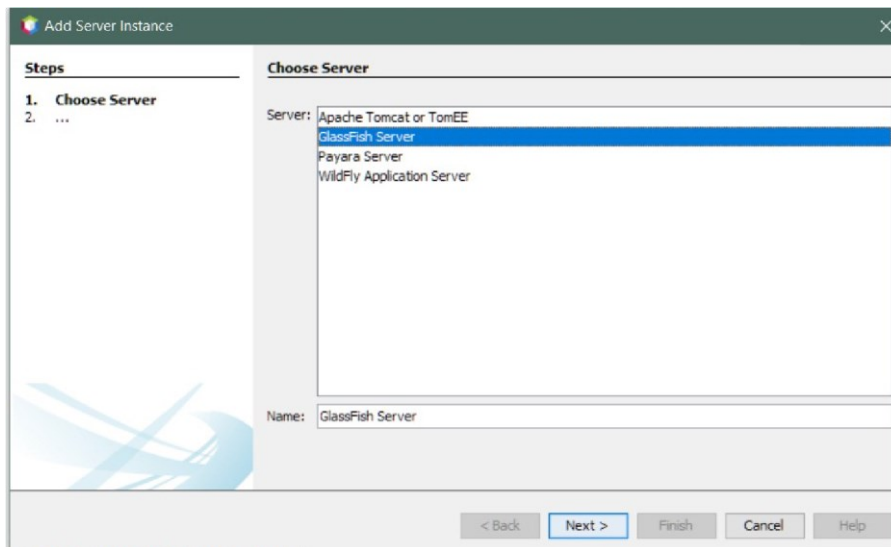
- a) Din meniul „Tools”, submeniul „Servers”



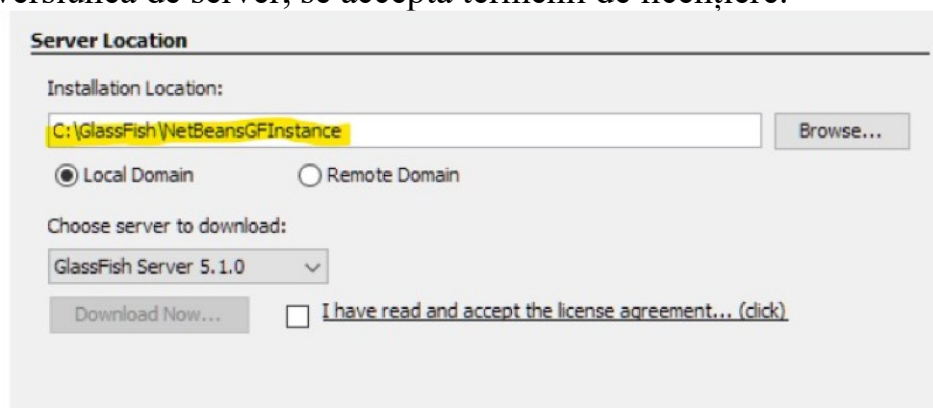
- b) Din fereastra „Servers”



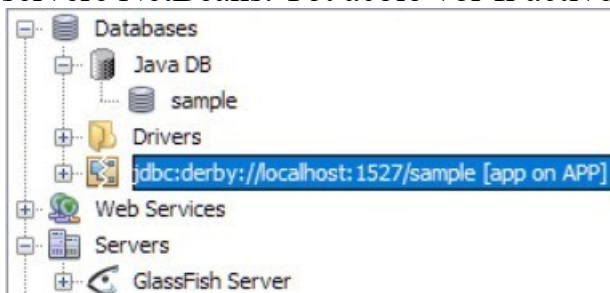
Se adaugă server de tip GlassFish. Atenție: este posibil ca NetBeans să afișeze cu întârziere opțiunile disponibile, acest pas necesită o conexiune la internet activă.



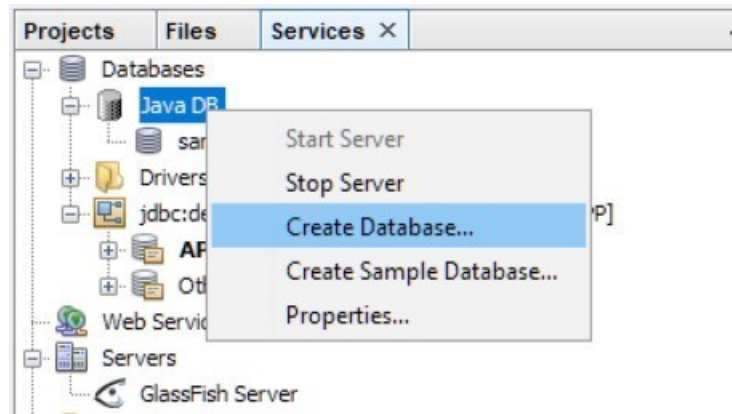
Se selectează locația unde se instalează serverul GlassFish (local domain), versiunea de server, se acceptă termenii de licențiere.



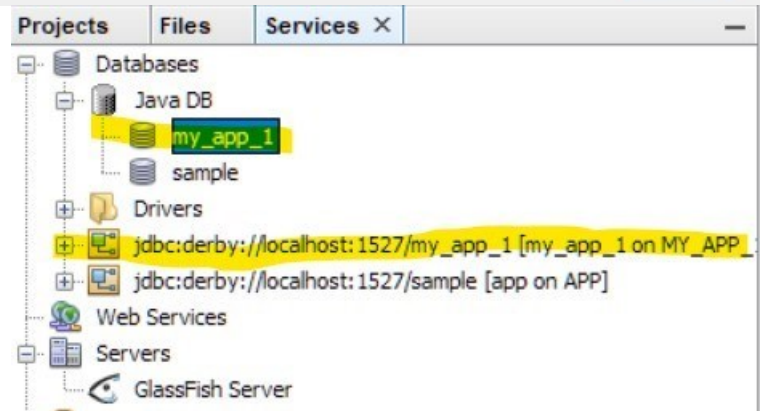
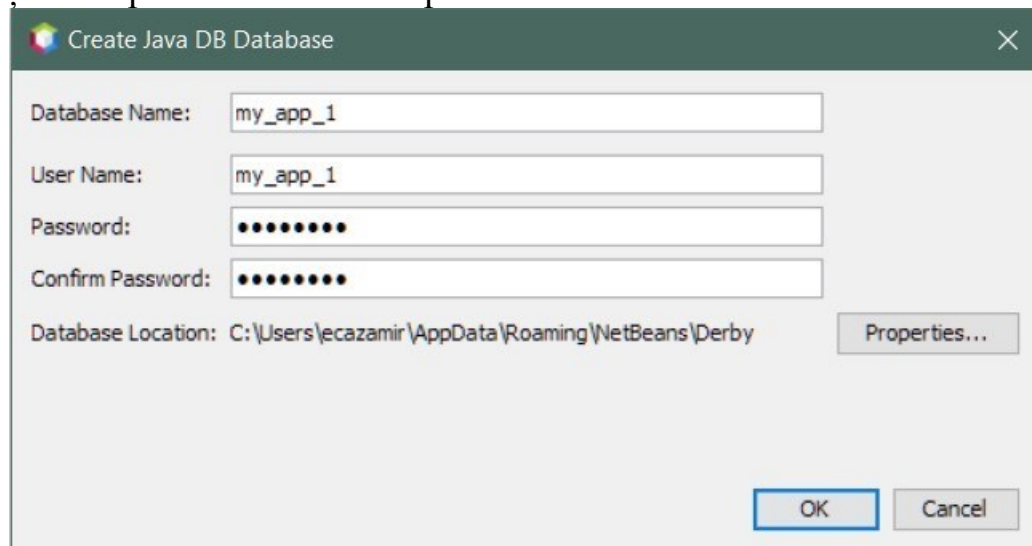
După ce se apasă pe butonul „Finish”, instanța de server va apărea în lista de servere NetBeans. Tot acolo vor fi active opțiunile JavaDB și JDBC:Derby



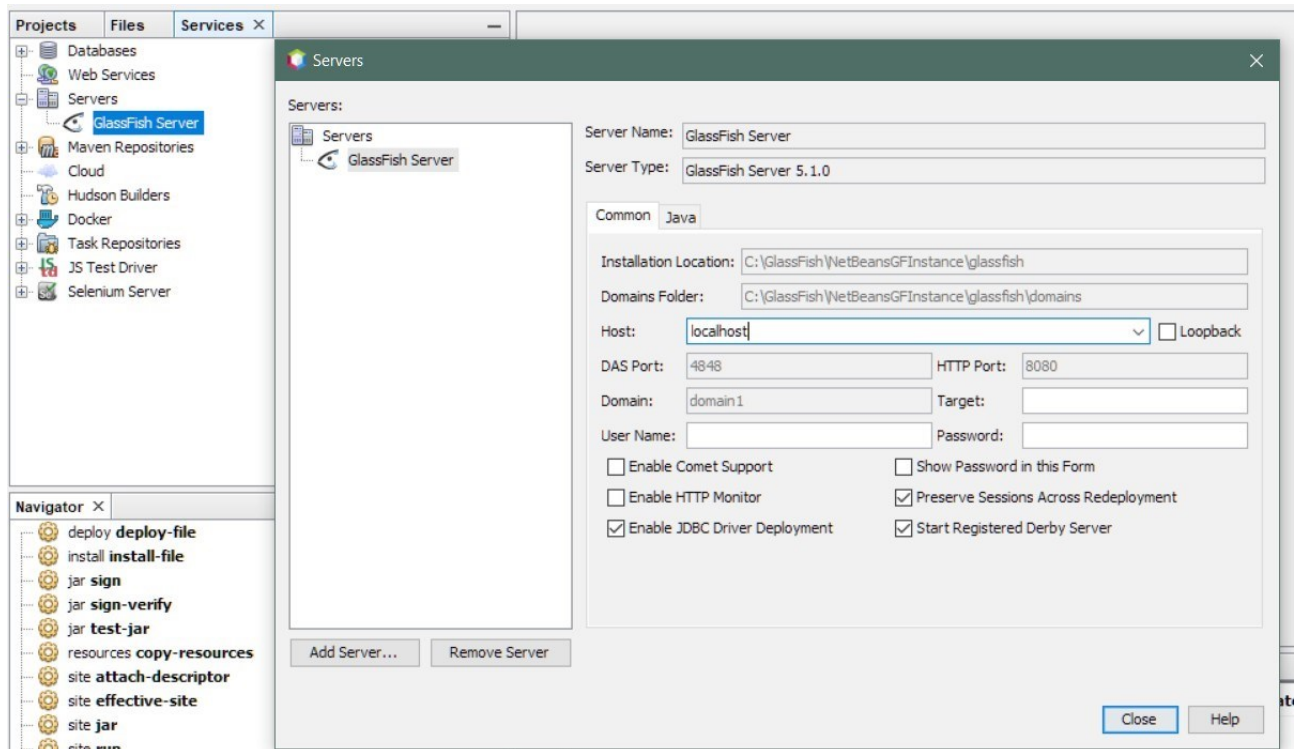
În acest moment este posibil să creați instanțe noi de baze de date derby (se va cere locația de amplasare pe disc)



Observați / setați amplasarea datelor bazei de date Derby, dacă nu sunteți mulțumiți de amplasamentul ales implicit:



Setările instanței se pot ajusta cu click dreapta pe serverul GlassFish (ex: versiune Java utilizată pentru instanța GlassFish, sau alți parametri)



Tutorial adaugare dependente JavaDB

1) Descarcam apache derby de pe site-ul :
https://db.apache.org/derby/derby_downloads.html

Recomand versiunile pentru java 8+

Apache Derby: Downloads

- ▣ [For Java 17 and Higher](#)
 - ▣ [For Java 9 and Higher](#)
 - ▣ [For Java 8 and Higher](#)
 - ▣ [For Java 6 and Higher](#)
 - ▣ [For Java 1.4 and Higher](#)
 - ▣ [For Java 1.3 and Higher](#)
 - ▣ [Deprecated Releases](#)
 - ▣ [Change History](#)
-


For Java 17 and Higher

- [10.16.1.1](#) (May 19, 2022 / SVN 1901046)

For Java 9 and Higher

- [10.15.2.0](#) (February 18, 2020 / SVN 1873585)
- [10.15.1.3](#) (March 5, 2019 / SVN 1853019)

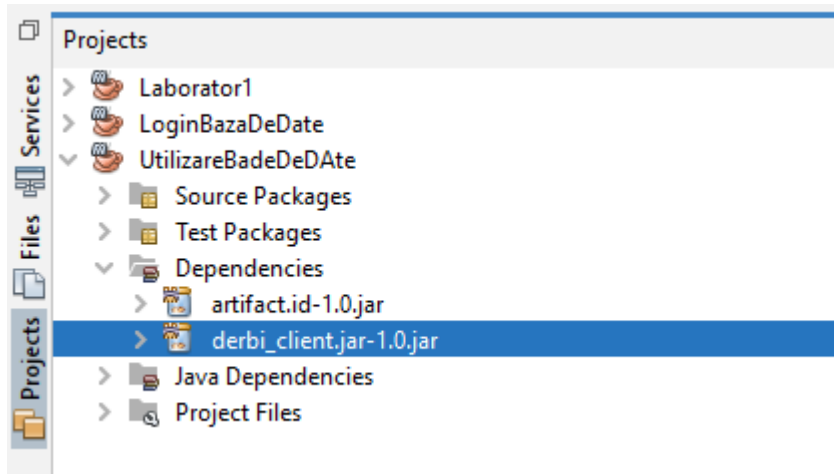
For Java 8 and Higher

- 
- [10.14.2.0](#) (May 3, 2018 / SVN 1828579)
 - [10.13.1.1](#) (October 25, 2016 / SVN 1766613)

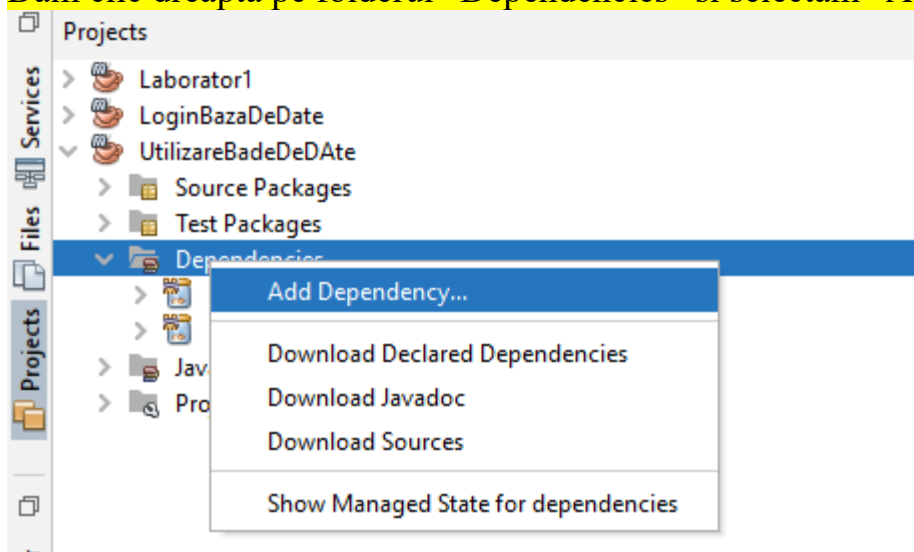
For Java 6 and Higher

- [10.12.1.1](#) (October 11, 2015 / SVN 1704137)
- [10.11.1.1](#) (August 26, 2014 / SVN 1616546)

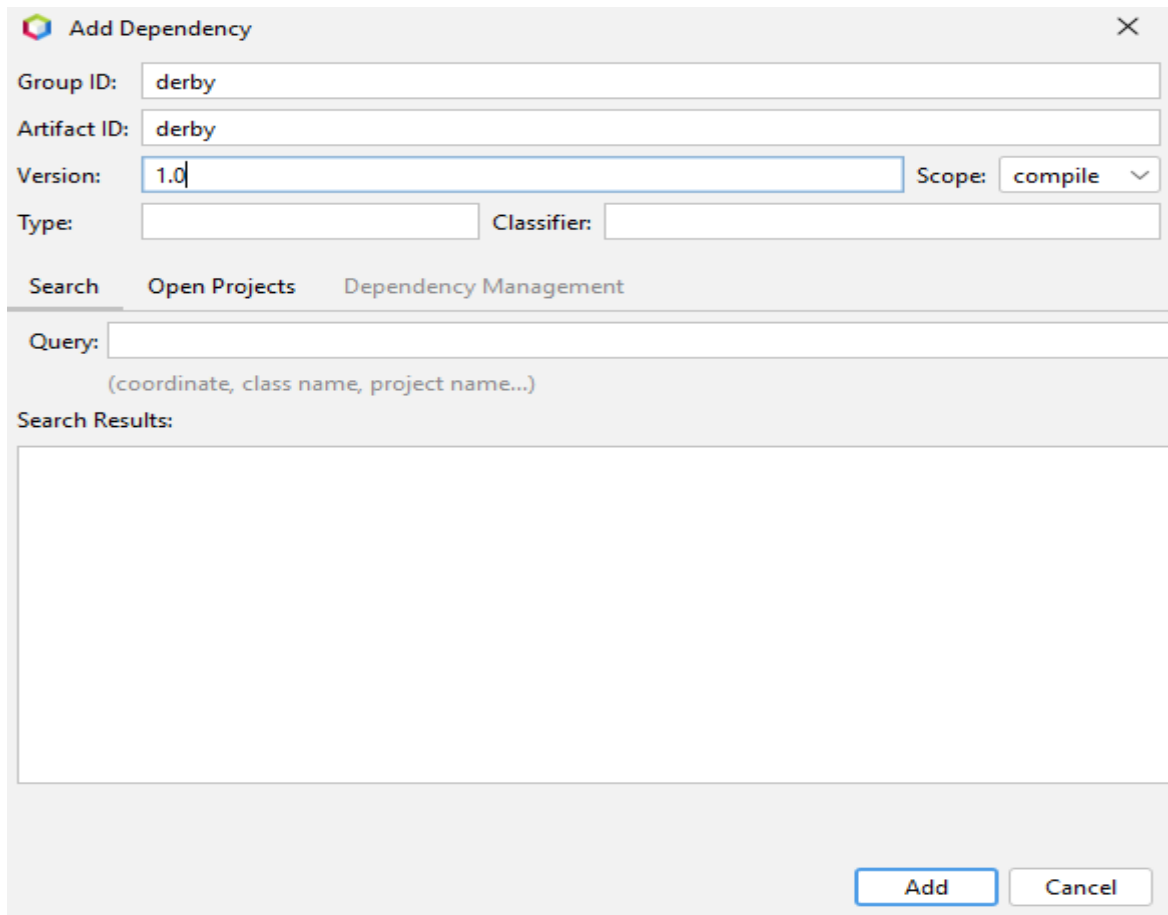
In cadrul proiectului existent se selecteaza fisierul "Projects ->Dependencies"



Dam clic dreapta pe folderul “Dependencies” si selectam “Add Dependencie...”



Completam campurile GroupID , ArtifactID si Version. Continutul nu conteaza dar recomand o denumire intuitiva. Dupa completare apasam pe “Add”, si adaugam 2 dependente, **derby.jar** si **derbyclient.jar**



The image shows a software dialog box titled "Add Dependency" with a close button (X) in the top right corner. It contains several input fields: "Group ID:" with the value "derby", "Artifact ID:" with the value "derby", "Version:" with the value "1.0", and "Scope:" with a dropdown menu set to "compile". There are also empty fields for "Type:" and "Classifier:". Below these fields is a tabbed interface with three tabs: "Search", "Open Projects", and "Dependency Management". The "Search" tab is active, showing a "Query:" input field and a hint "(coordinate, class name, project name...)". Below the query field is a section labeled "Search Results:" which is currently empty. At the bottom right of the dialog are two buttons: "Add" and "Cancel".

Add Dependency

Group ID: derby

Artifact ID: derby

Version: 1.0 Scope: compile

Type: Classifier:

Search Open Projects Dependency Management

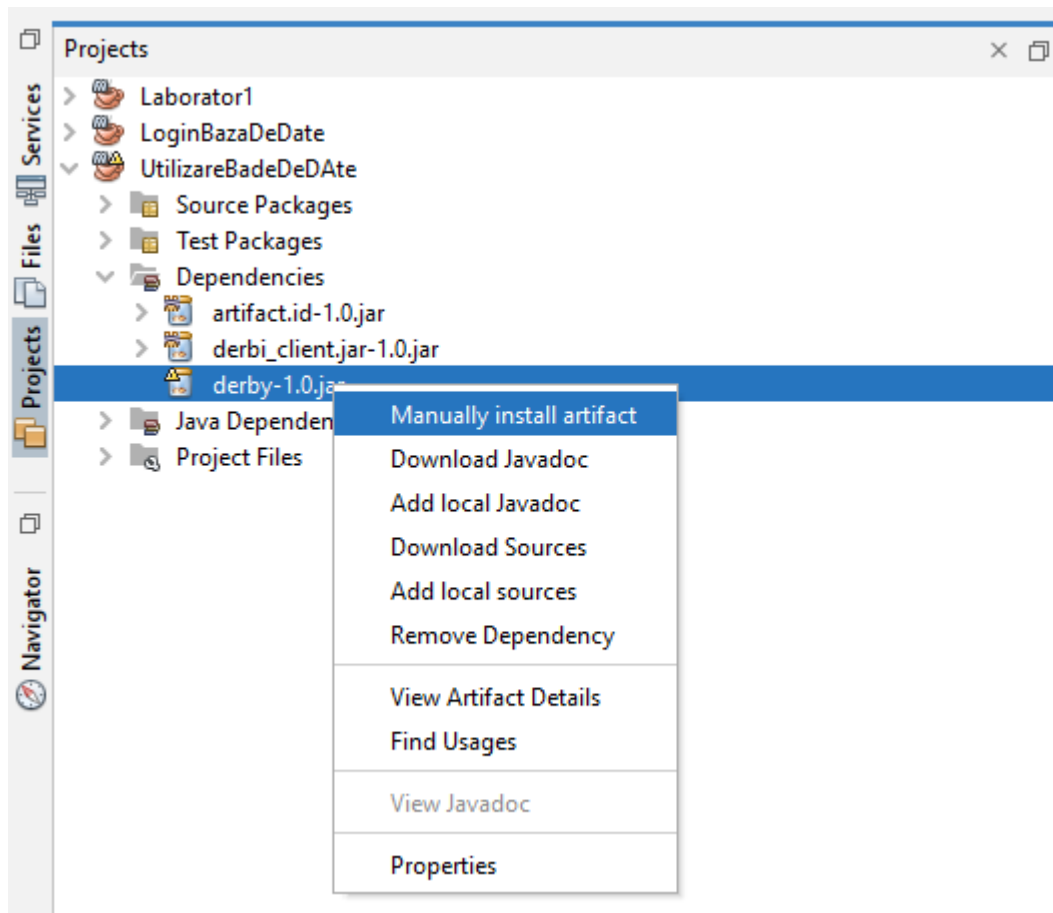
Query:

(coordinate, class name, project name...)

Search Results:

Add Cancel

Observam ca a fost creata o dependenta in cadrul fisierului. Dam click dreapta pe aceasta si selectam "Manually install artifact"



Se va deschide o fereastra de dialog in care definim calea spre fisierele care dorim sa fie introduse drept dependente. Dependentele necesare acestui proiect sunt in folderul descarcat mai sus “**Derby -> libs**” si se numesc **derbi.jar** si **derbiclient.jar**

