



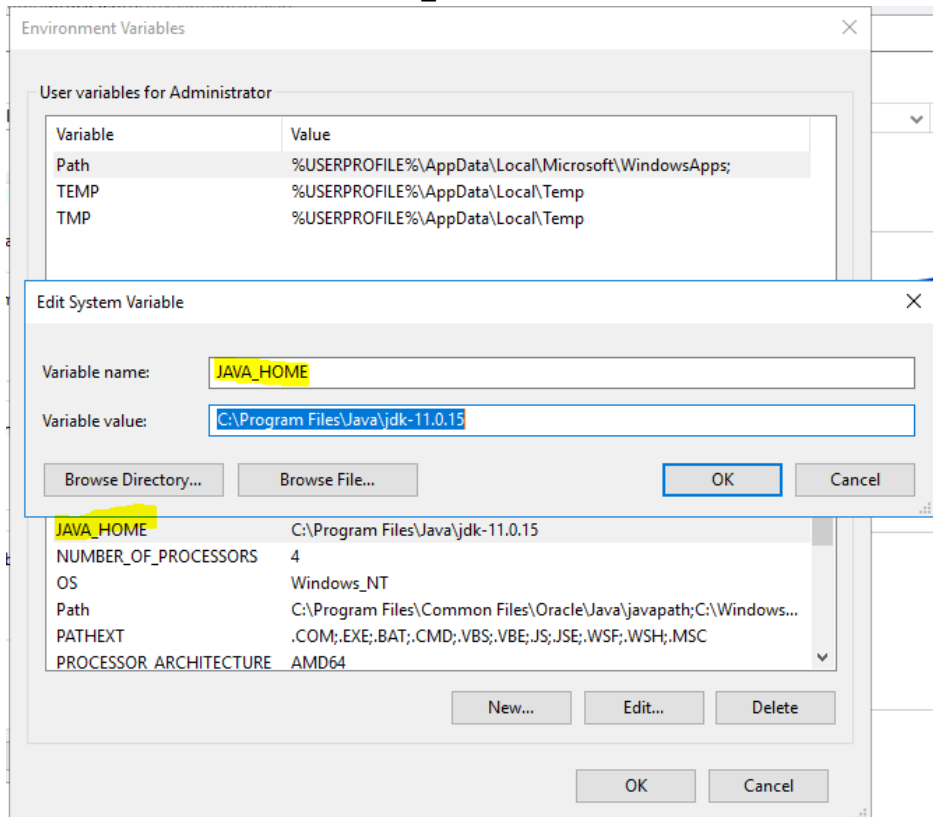
Paradigme de programare (în Java)

Lab 1/Curs 1- Introducere

Col(r) Traian Nicula

Instalare și configurare mediu de dezvoltare

1. Se descarcă și instalează Visual Studio Code, ultima versiune
2. Se descarcă și instalează Java SE 11 și Java SE 17
3. Se setează variabila de sistem JAVA_HOME la calea de instalare a Java SE 11



4. Se instalează [Extension Pack for Java](#)




Visual Studio | Marketplace

Visual Studio Code > Programming Languages > Extension Pack for Java

Sign in

New to Visual Studio Code? [Get it now.](#)



Extension Pack for Java

Microsoft | 13,108,501 installs | ★★★★★ (49) | Free

Popular extensions for Java development that provides Java IntelliSense, debugging, testing, Maven/Gradle support, project management and more.

[Install](#) [Trouble Installing?](#)

[Overview](#) [Version History](#) [Q & A](#) [Rating & Review](#)

Extension Pack for Java

Extension Pack for Java is a collection of popular extensions that can help write, test and debug Java applications in Visual Studio Code. Check out [Java in VS Code](#) to get started.


Extensions Included


Categories
[Programming Languages](#) [Snippets](#) [Linters](#)
[Debuggers](#) [Formatters](#) [Extension Packs](#)

Tags
[debugger](#) [java](#) [junit](#) [keybindings](#) [maven](#)

Details **Feature Contributions** **Changelog**

Extension Pack (6)

**IntelliCode**
AI-assisted development
Microsoft [Install](#)

**Language Support for Java(TM) by Red Hat**
Java Linting, Intellisense, formatting, refactor...
Red Hat [Install](#)

Extension Pack for Java

Extension Pack for Java is a collection of popular extensions that can help write, test and debug Java applications in Visual Studio Code. Check out [Java in VS Code](#) to get started.

Extensions Included

By installing Extension Pack for Java, the following extensions are installed:

Categories
[Programming Languages](#) [Snippets](#) [Linters](#)
[Debuggers](#) [Formatters](#) [Extension Packs](#)

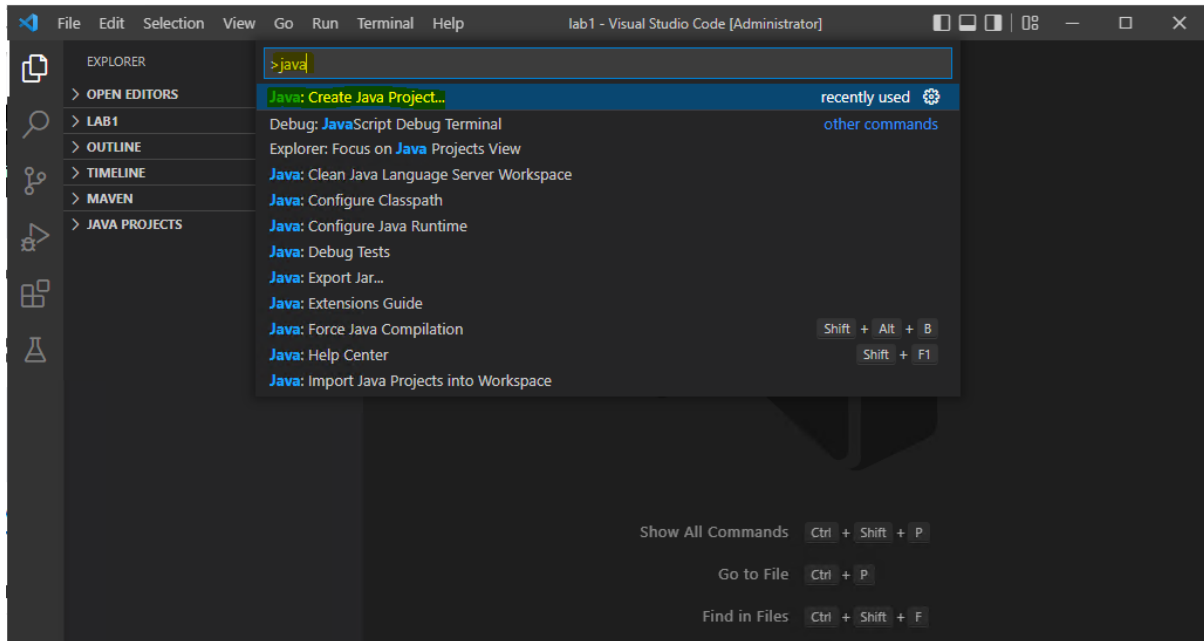
Resources
[Marketplace](#)
[Repository](#)
[License](#)
[microsoft.com](#)

Marketplace Info
Released on 9/27/2017, 12:38:52
Last updated 6/2/2022, 06:34:57
Identifier vscjava.vscode-java-pack

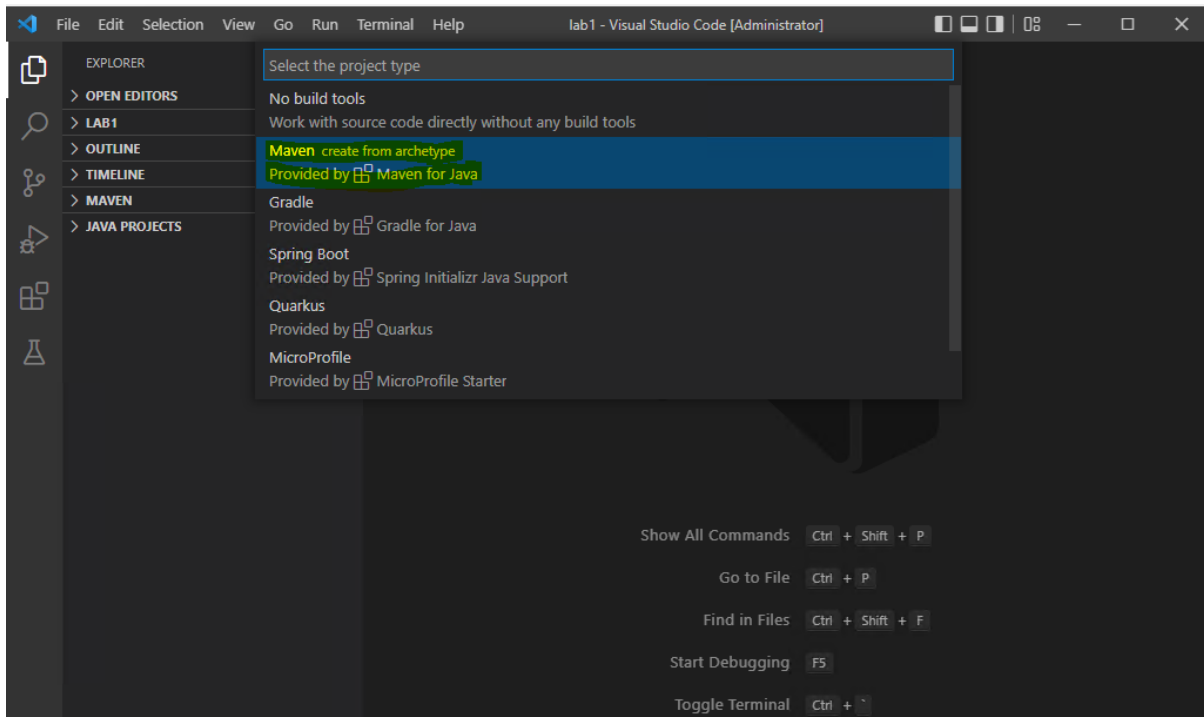


Creare proiect JAVA cu MAVEN

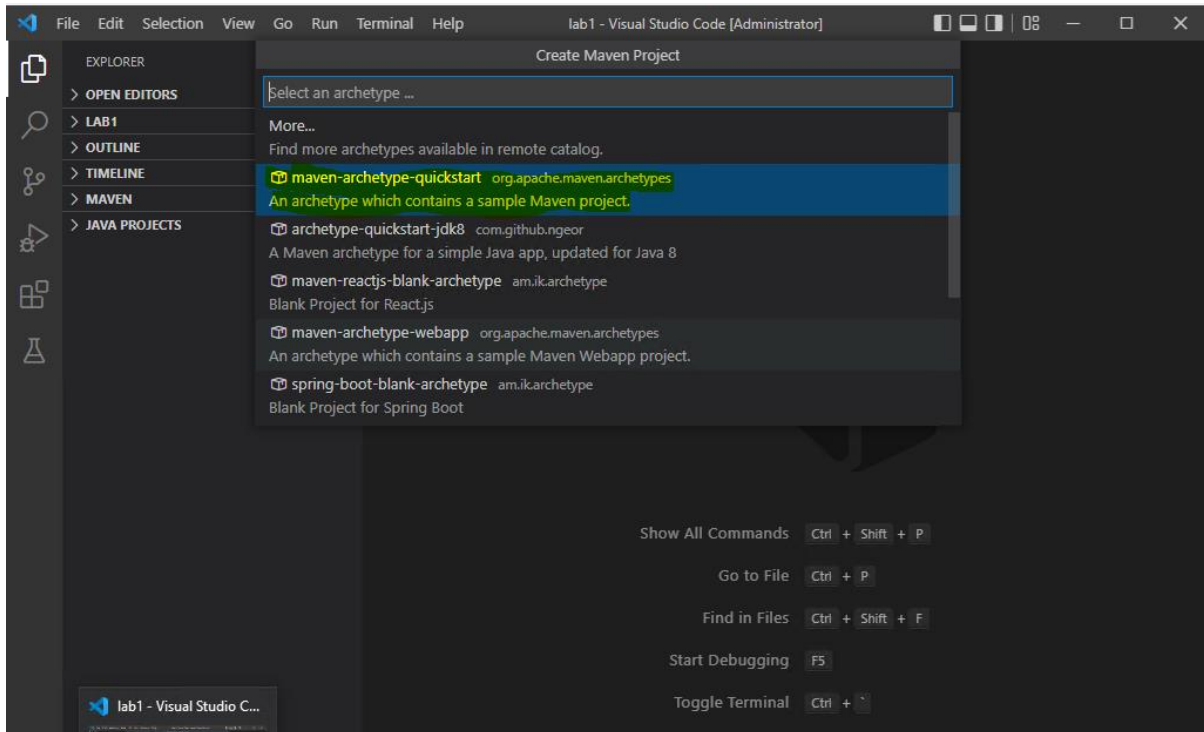
1. Se apasă **CTRL + SHIFT + P** sau **F1** sau din meniu **View -> Command Palette**, iar în câmpul de căutare apărut se tastează **java**. Se alege **Java: Create Java Project ...**



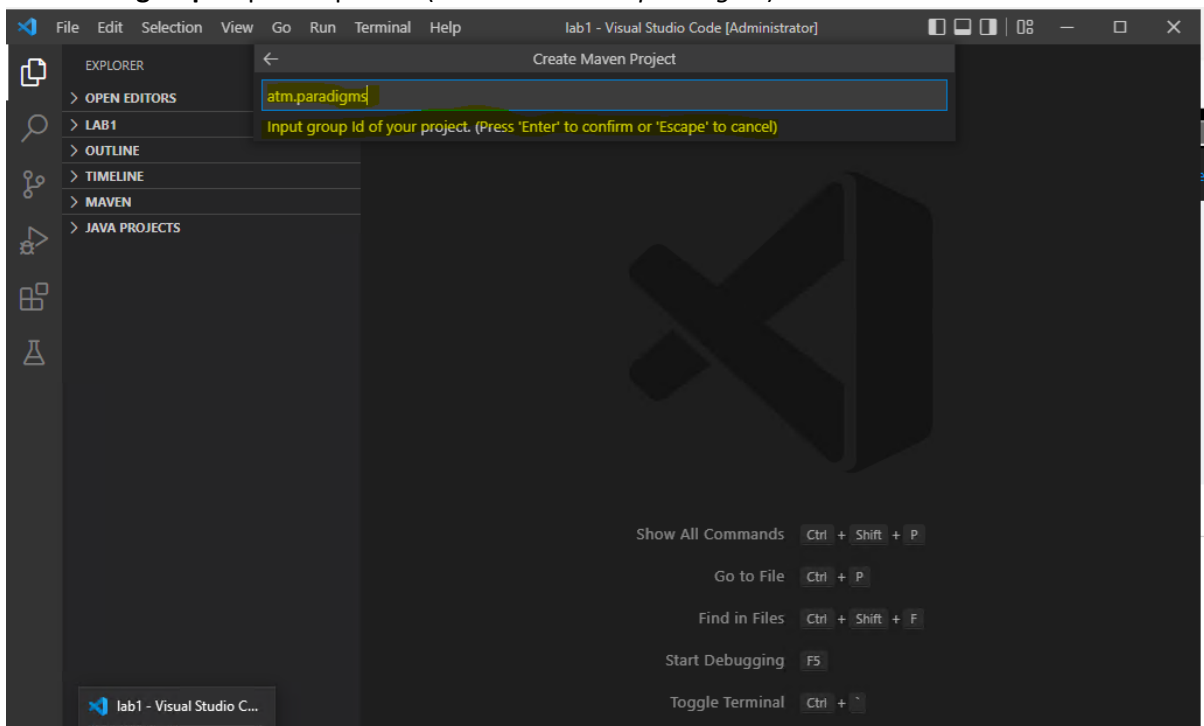
2. Se alege **Maven create from archetype**



3. Se alege **maven-archetype-quickstart** (pentru aplicații Java)



4. Se alege versiunea 1.4 (implicit)
5. Se setează **group ID** pentru proiect (se utilizează *atm.paradigms*)



6. Se setează artifact ID (nume proiect – **lab1**)



7. Se alege locația pentru crearea proiectului (folder-ul cu numele studentului)
8. Începe crearea proiectului. Se confirmă în consolă cu **ENTER** sau schimbă versiunea și numele pachetului (același cu group ID)
9. La finalizare se deschide proiectul

```
File Edit Selection View Go Run Terminal Help App.java - lab2 - Visual Studio Code [Administrator]
EXPLORER
  pom.xml
  App.java src/main/java/atm/para...
  LAB2
    src
      main\java\atm\paradigms
      App.java
    test
    target
    pom.xml
  OUTLINE
  TIMELINE
  JAVA PROJECTS
  MAVEN
  0 2
  Ln 14, Col 1 Spaces: 4 UTF-8 CRLF {} Java
```

```
src > main > java > atm > paradigms > App.java > ...
1 package atm.paradigms;
2
3 /**
4  * Hello world!
5  *
6  */
7 public class App
8 {
9     Run | Debug
10    public static void main( String[] args )
11    {
12        System.out.println( x: "Hello World!" );
13    }
14 }
```

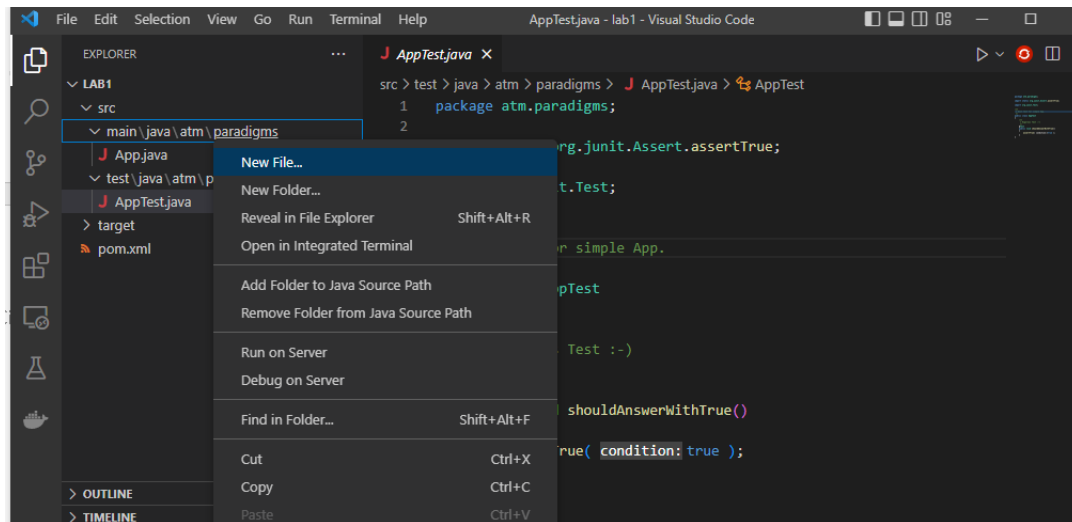
10. Se rulează proiectul având clasa **App** deschisă în editor prin apăsarea butonului **Run Java**. Se observă rezultatul în consolă.

```
J App.java X
src > main > java > atm > paradigms > J App.java > App > main(String[])
1 package atm.paradigms;
2
3 /**
4  * Hello world!
5  *
6  */
7 public class App
8 {
9     Run | Debug
10    public static void main( String[] args )
11    {
12        System.out.println( x: "Hello World!" );
13    }
14 }
```

11. Se modifică mesajul și se execută din nou



- Se creează o nouă clasă cu numele **MyClass** prin clic dreapta în Explorer pe folder-ul *paradigms*, Se apasă New Folder apoi în câmpul apărut se introduce **MyClass.java**



- Se deschide fișierul automat în editor în editor și se alege **class** din meniul contextual
- În corpul clasei se tastează și se alege **main** din meniul contextual. La clasă se adaugă metoda statică main. Folosind metoda **System.out.println()** se adaugă un mesaj în *main* ce va fi afișat la rulare.
- Se salvează și rulează

Utilizare VSC

- Închide proiectul creat anterior cu **Ctrl + K** apoi **F** sau **File -> Close Folder**
- Deschide proiectul **lab1** cu **Ctrl + K** și **Ctrl + O** sau **File -> Open Folder ...**
- Deschide fișierul **MyClass.java** și testează următoarele scurtături de taste

Scurtătură	Descriere
Ctrl+Space	Afișează sugestii (se selectează System)
Ctrl+Shift+Space	Afișează sugestii parametrii (se selectează println)
Shift+Alt+F	Formatează documentul deschis
F2	Redenumeste simbol (se selectează clasa MyClass și se redenumeste MyClass1)
Ctrl+.	Sugestii de remediere probleme (importuri, conversie tip etc.)
Ctrl+/	Comentează/elimină comentariul pentru linia/liniile selectate
F8	Duce cursorul la următoarea eroare

- Creează din Explorer, sub **paradigms**, folder-ul **model** și copiază fișierul **MyClass1.java** cu *drag and drop*.