

# Curs 11 PPOO

Prof. univ. dr. Cristian CIUREA

Departamentul de Informatică și Cibernetică Economică

[cristian.ciurea@ie.ase.ro](mailto:cristian.ciurea@ie.ase.ro)

# Agenda

- ▶ JavaFX vs. Swing
- ▶ Junit / TDD / BDD
- ▶ FindBugs / QAPlug-FindBugs

# Resurse utile JavaFX

- ▶ <https://www.slideshare.net/alexandercasall/javafx-10-things-i-love-about-you>
- ▶ <https://www.slideshare.net/alexandercasall/javafx-pitfalls>
- ▶ <https://o7planning.org/en/11009/javafx>
- ▶ <https://github.com/mhrimaz/AwesomeJavaFX>

# Swing sau JavaFX?

## Swing

- ▶ Oferă maturitate și stabilitate
- ▶ Include o mulțime de biblioteci de componente și framework-uri
- ▶ Oferă o multitudine de resurse

## JavaFX

- ▶ Modern, suport MVC, CSS, FXML
- ▶ Suport 3D și Animații
- ▶ Este posibil să nu fie încă destul de „solid” în producție
- ▶ Nu oferă atât de multe resurse

# JUnit

- ▶ **Unit Testing:** testarea celor mai mici componente ale codului sursă (clase sau metode).
- ▶ **Unit Testing** este o metodă de testare software prin care unitățile individuale de cod sursă sunt testate pentru a determina dacă acestea sunt bune pentru a fi utilizate.

# JUnit

De ce să efectuăm **Unit Testing**?

- ▶ testele sunt ușor de scris;
- ▶ teste pot fi scrise ad-hoc, atunci când avem nevoie de ele;
- ▶ pe baza acestora se pot defini colecții de teste - denumite TestSuites;
- ▶ acestea pot fi rulate automat ori de câte ori este necesar (se scriu o dată și se utilizează de mai multe ori);
- ▶ există mai multe framework-uri și instrumente care simplifică procesul de scriere și rulare a testelor;
- ▶ reduce timpul petrecut la depanare și pentru găsirea de bug-uri;
- ▶ reduce numărul de bug-uri în codul livrat sau integrat;
- ▶ mărește rata de identificare a erorilor în timpul fazei de scriere a codului.

# JUnit

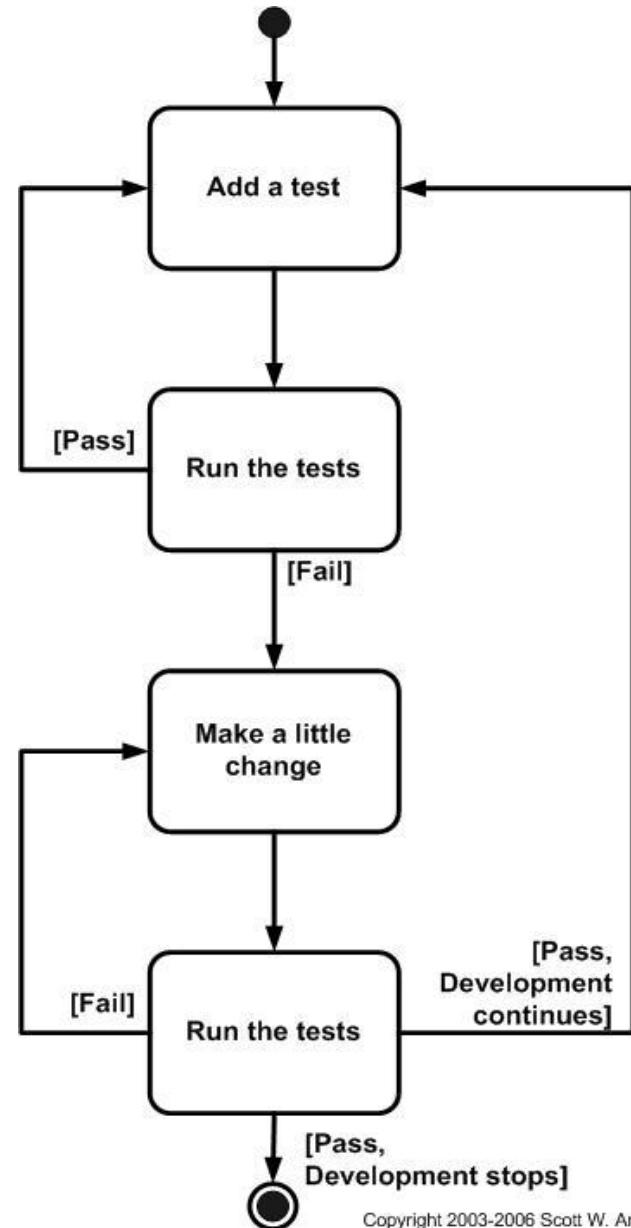
- ▶ JUnit este un instrument pentru dezvoltare Test-Driven (TDD).
- ▶ JUnit este un cadru de clase care permite scrierea și executarea de teste pentru diferite metode / clase din codul sursă.
- ▶ Arhitectura JUnit include componentele:
  - ▶ **TestRunner,**
  - ▶ **TestResults,**
  - ▶ **TestCase,**
  - ▶ **Assert**

# TDD

- ▶ **Test-Driven Development (TDD)** este un proces de dezvoltare software care se bazează pe repetarea unui ciclu de dezvoltare foarte scurt.
- ▶ **Test-Driven Development (TDD)** înseamnă:
  - ▶ scrierea unui test care eșuează;
  - ▶ modificarea codului sursă, astfel încât să meargă;
  - ▶ eliminarea redundanțelor.



# TDD



# BDD

- ▶ **Behavior-Driven Development (BDD)** este o metodă agilă de dezvoltare a software-ului introdusă de Dan North în celebra sa lucrare *Introducing BDD*.
- ▶ BDD este, de asemenea, cunoscut sub numele de **Acceptance Test Driven Development (ATDD)**, testarea poveștii sau specificarea prin exemplu.
- ▶ BDD încurajează colaborarea între dezvoltatori și utilizatori non-tehnici într-un proiect software pentru a defini specificațiile și a decide criteriile de acceptare prin scrierea cazurilor de testare într-un limbaj natural pe care non-programatorii îl pot citi.

# BDD

- ▶ **Behavior-Driven Development (BDD)** folosește instrumente precum:
  - ▶ *Cucumber* - Ruby
  - ▶ *Capybara* - Ruby
  - ▶ *Behave* - Python
  - ▶ *Lettuce* - Python
  - ▶ *JBehave* - Java
  - ▶ *SpecFlow.NET* - .NET

# FindBugs

- ▶ **FindBugs** caută bug-uri în programele Java. Acesta se bazează pe conceptul de șabloane de bug-uri. Un șablon de bug este un cuvânt sau secvență de cod sursă, care generează de multe ori o eroare.
- ▶ FindBugs utilizează analiza statică pentru a inspecta bytecode-ul Java pentru a determina aparițiile șabloanelor de bug-uri.
- ▶ FindBugs poate identifica bug-uri pur și simplu prin inspectarea codului sursă al unui program, fără ca execuția programului să fie necesară.

# FindBugs

Descrierea tipurilor de bug-uri identificate:

- ▶ **Bad practice** (încălări ale practicilor recomandate și esențiale de scriere a codului sursă);
- ▶ **Correctness** (o eroare aparentă de codificare care rezultă în cod și care nu a fost, probabil, ceea ce a dorit dezvoltatorul);
- ▶ **Malicious code vulnerability**;
- ▶ **Multithreaded correctness**;
- ▶ **Performance**;
- ▶ **Dodgy** (cod care este confuz sau scris într-un mod care conduce la erori).

# FindBugs

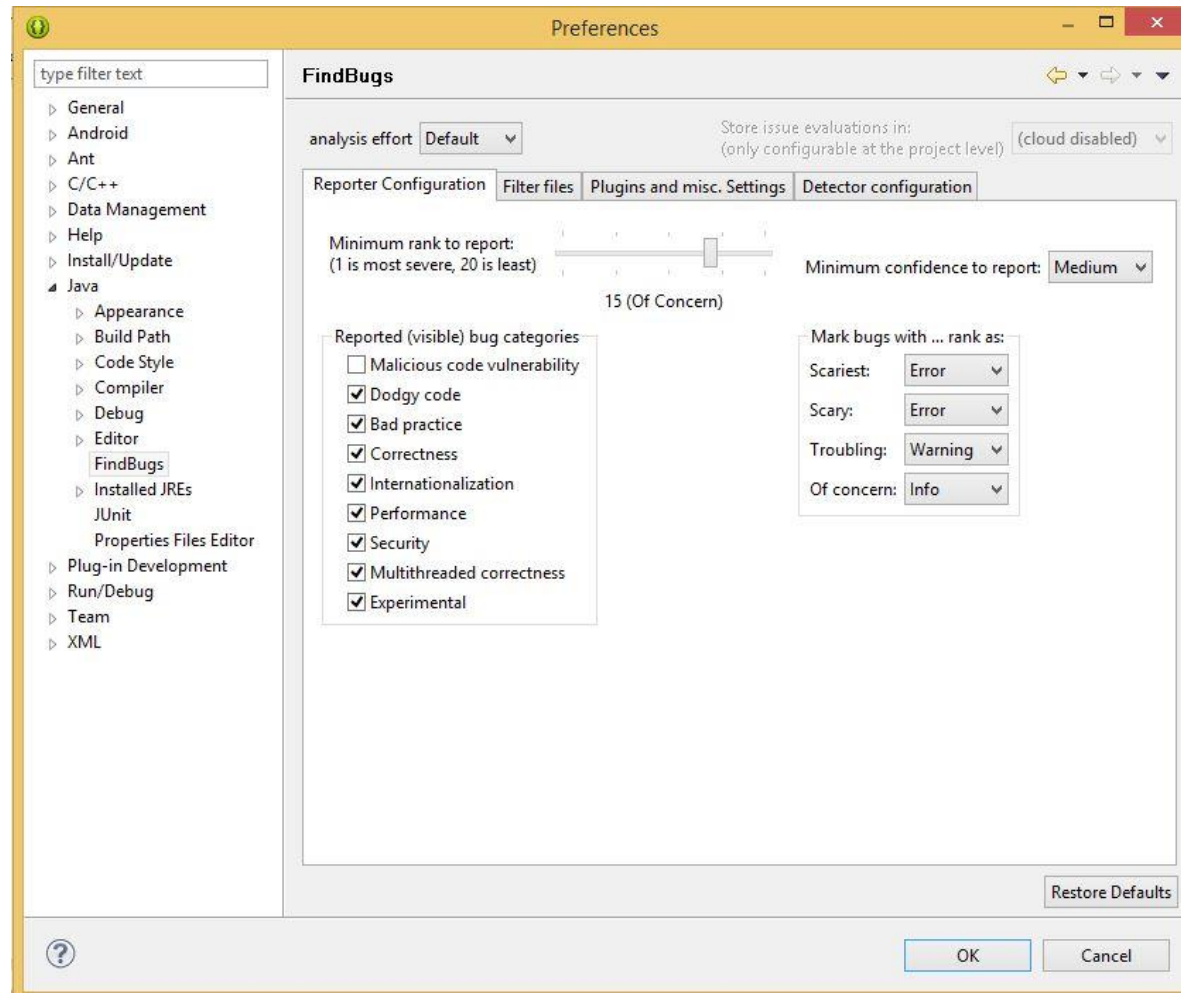
Instalare FindBugs Eclipse plugin:

- ▶ In **Eclipse**, select **Help->Install new software**
- ▶ Click **New Remote Site**
- ▶ Name: **FindBugs update site**
- ▶ URL: <http://findbugs.cs.umd.edu/eclipse>
- ▶ You should see **FindBugs Feature**
- ▶ Select the **I accept** option to accept the license and click **Next**

# FindBugs

Configurare FindBugs Eclipse plugin:

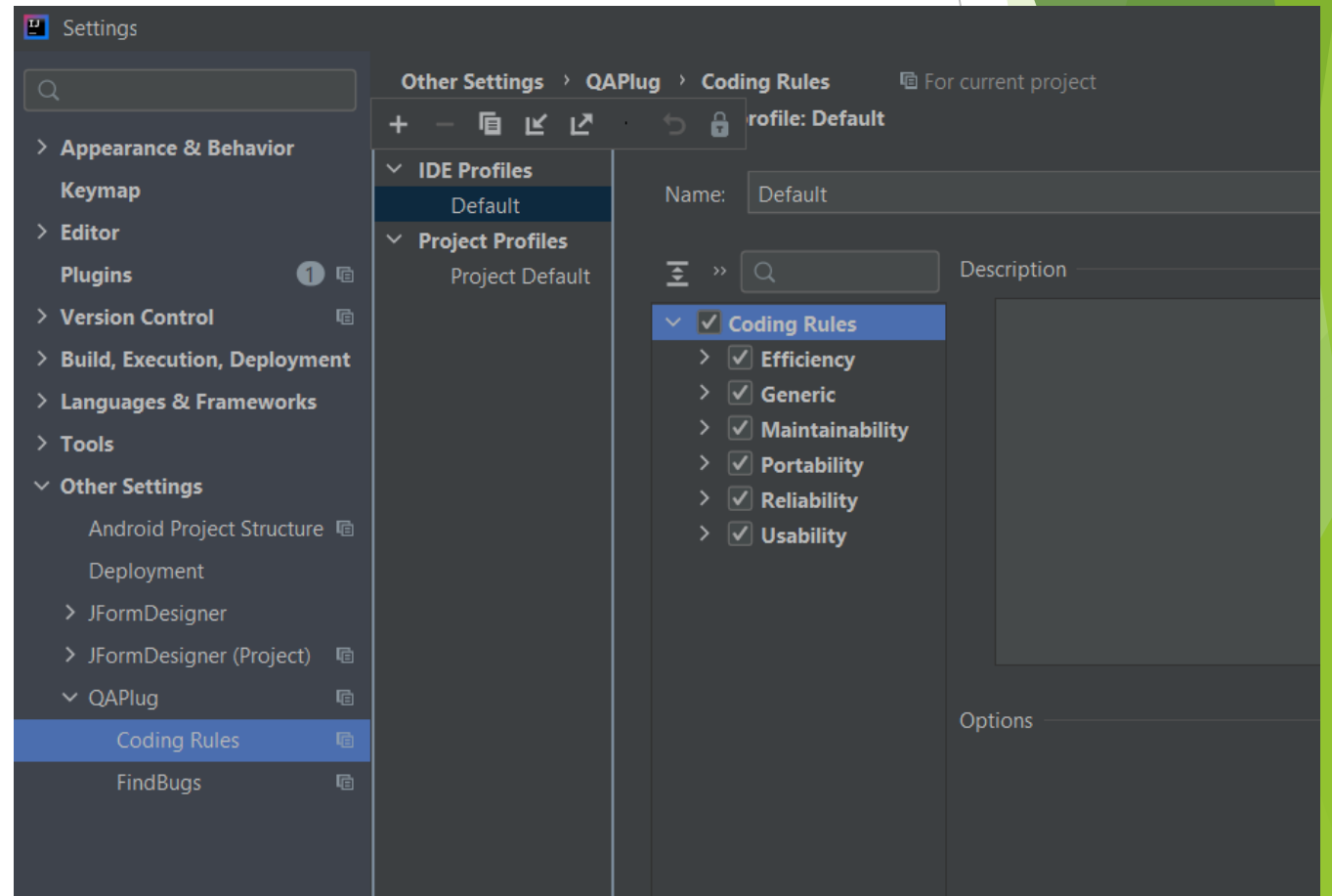
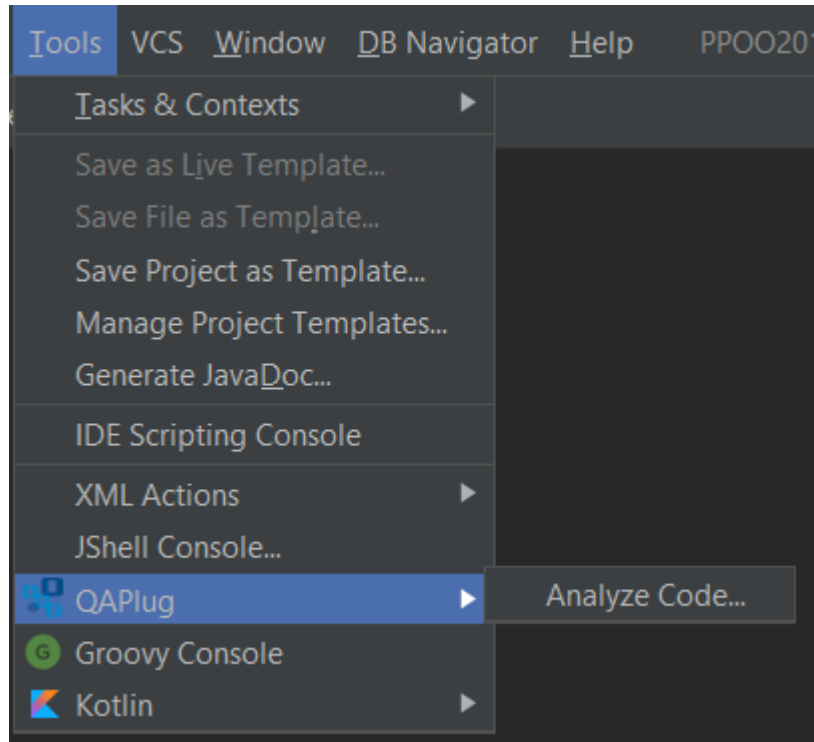
- In Eclipse, select **Windows->Preferences**, under **Java** select **FindBugs**



# QAPlug-FindBugs

Configurare plugin QAPlug-FindBugs in IntelliJ:

- In IntelliJ IDEA, select **File>Settings>Plugins**, select **QAPlug-FindBugs**





# Bibliografie

- ▶ [1] Jonathan Knudsen, Patrick Niemeyer - *Learning Java, 3<sup>rd</sup> Edition*, O'Reilly.
- ▶ [2] <http://www.itcsolutions.eu>
- ▶ [3] <http://www.acs.ase.ro>
- ▶ [4] [https://en.wikipedia.org/wiki/Swing\\_\(Java\)](https://en.wikipedia.org/wiki/Swing_(Java))
- ▶ [5] [http://wiki.dcae.pub.ro/index.php/Graphical\\_User\\_Interface\\_\(GUI\)\\_-\\_Java\\_Swing\\_%C8%99i\\_JavaFX](http://wiki.dcae.pub.ro/index.php/Graphical_User_Interface_(GUI)_-_Java_Swing_%C8%99i_JavaFX)
- ▶ [6] <https://www.javaguides.net/2019/07/javafx-hello-world-example-tutorial.html?m=1>
- ▶ [7] [https://ro.wikipedia.org/wiki/Spring\\_Framework](https://ro.wikipedia.org/wiki/Spring_Framework)