



UNIVERSITATEA BABEȘ-BOLYAI
Facultatea de Matematică și Informatică



ALGORITMI ȘI PROGRAMARE

Cursul 3

Programare modulară

Ionescu Vlad

vlad.ionescu@ubbcluj.ro

Feedback săptămâna 2

- Întrebarea de început
- Feedback

IDE-uri pentru Python **Exemplu**

- ❑ Un program real este format din mai multe fișiere
- ❑ **PyCharm** – IDE “official” pentru curs
- ❑ Eclipse + PyDev
- ❑ Visual Studio Code
- ❑ Un IDE ajută la:
 - Lucrul cu mai multe fișiere
 - Depanarea programului
 - Integrarea documentației
 - Respectarea unui stil de programare
 - Refactorizarea codului

Conceptul de programare modulară

- ❑ Programarea modulară: organizarea codului în module responsabile de anumite părți ale programului
- ❑ Un modul conține cod
- ❑ Modulele pot fi dezvoltate independent
- ❑ Cum am organiza programele de până acum în module?

În practică

- Are la bază descompunerea problemei în subprobleme având în vedere:
 - separarea conceptelor
 - arhitecturi stratificate
 - întreținerea și reutilizabilitatea codului
 - coeziunea elementelor dintr-un modul
 - cuplarea (legarea) modulelor între ele

Module în Python **Exemplu**

- ❑ Crearea unui modul
- ❑ Folosirea unui modul – modalități de importare

Packages în Python **Exemplu**

- ❑ O colecție de module înrudite ca rol
- ❑ Conferă și mai multă structură codului

Organizarea aplicațiilor în pachete și module

- ❑ Vom organiza aplicațiile în pachete și module corespunzătoare diverselor straturi
 - User Interface
 - ❑ Module care gestionează interacțiunea cu utilizatorul
 - Logic
 - ❑ Module care implementează partea logică a aplicației: calcule etc.
 - Domain
 - ❑ Module care modelează datele existente în problemă
 - Tests
 - ❑ Module care testează alte module

Exemplu de aplicație Exemplu

- ❑ Scrieți un program care implementează un calculator de numere raționale:
 - Identificați posibile cerințe
 - Scrieți scenarii de rulare pentru fiecare funcționalitate
 - Lucrați test-driven
 - Scrieți documentație

Programare modulară

Q & A