

# Programare Procedurală

Marinescu–Ghemeci Ruxandra

[verman@fmi.unibuc.ro](mailto:verman@fmi.unibuc.ro)

[ruxandra.marinescu@fmi.unibuc.ro](mailto:ruxandra.marinescu@fmi.unibuc.ro)

[ruxandra.marinescu@unibuc.ro](mailto:ruxandra.marinescu@unibuc.ro)

# Programa



# Programa

## ► Programare procedurală în limbajul Python

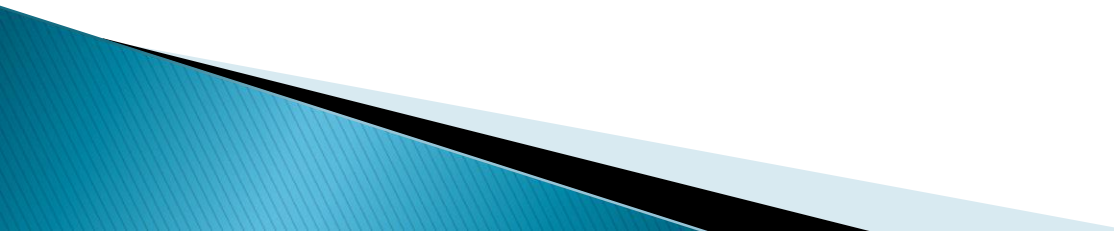
- Elemente de bază (tipuri de date, variabile, operatori, expresii, instrucțiuni, citire/scriere..)
- Colecții (liste, tupluri, mulțimi, dicționare)
- Șiruri de caractere
- Funcții (predefinite, definite de utilizator, recursivitate). **Module**
- Fișiere
- Excepții

# Programa

## ► Tehnici de programare:

- Algoritmi. Complexitate
- Greedy
- Divide et Impera
- Backtracking
- Programare dinamica

# Objective

- ▶ formarea deprinderilor de **programare structurată** în limbaje de programare clasice și modern (descompunerea unei probleme complexe în subprobleme relative simple și independente)
  - ▶ abilități de implementare și testare a algoritmilor în limbajul **Python**
  - ▶ dezvoltarea unei **gândiri algoritmice**
  - ▶ abilități de justificare a **corectitudinii** algoritmilor propuși și de determinare a **complexității** acestora
- 

# Obiective. Motivații

## ▶ Python

- elemente de bază
- lucrul cu structuri de date

# Obiective. Motivații

## ► Python – avantaje

- sintaxa simplă, sugestivă
- dinamic
- de actualitate
- <https://spectrum.ieee.org/the-top-programming-languages-2023#toggle-gdpr>
- numeroase facilități (incluse automat): dezvoltare software, web, GUI, module pentru IA, ML (Google – motoare de căutare)
- portabil
- open-source: [www.python.org](http://www.python.org)
- garbage collection
- comunitate de programatori mare și activă

# Obiective. Motivații

## ► Tehnici de programare

- algoritmi **eficienți**

*"Perhaps the most important principle for the good algorithm designer is to refuse to be content" –*

Aho, Hopcroft, and Ullman, The Design and Analysis of Computer Algorithms

## Exemple de probleme

- Aflarea minimului și maximului dintr-un vector
- Cele mai apropiate două puncte dintr-o mulțime de puncte din plan dată
- Numărul de inversiuni dintr-un vector
- Înmulțirea a două numere / matrice



# Obiective. Motivații

## ► Tehnici de programare

- algoritmi **corecți**

### Exemple de probleme

- Dată o mulțime de intervale, să se determine o submulțime de cardinal maxim de intervale care nu se suprapun
- Dată o mulțime de intervale, fiecare interval având asociată o pondere, să se determine o submulțime de intervale care nu se suprapun având ponderea totală maximă

# Objective. Motivații

- Numeroase aplicații
  - Statistică, Geometrie computațională ..
  - Proiectare, jocuri, strategii
  - Bioinformatică, procesare texte, imagini
  - Căutare web, similitudini, aliniere
  - Probleme de planificare
  - Baze de date – arbori de căutare optimi
- Probleme interviuri

# Structura

## ▶ Curs

- 2 ore pe săptămână
- finalizat cu examen scris

## ▶ Laborator

- 2 ore pe săptămână
- finalizat cu test de laborator

# Evalutare



# Evaluare

## ► Test de laborator

- după vacanță, dar până în sesiune, la laborator
- doar Python, nu și tehnici de programare
- **Eliminatoriu**: nepromovarea testului de laborator nu permite participarea la examen (ul din sesiune)

## ► Examen scris în sesiune

**Nota finală** = media celor două note

Condiții necesare:

- **Nota test laborator (nerotunjită)  $\geq 5$**
- **Nota finală (nerotunjită)  $\geq 5$**

# Bibliografie



# BIBLIOGRAFIE – Python

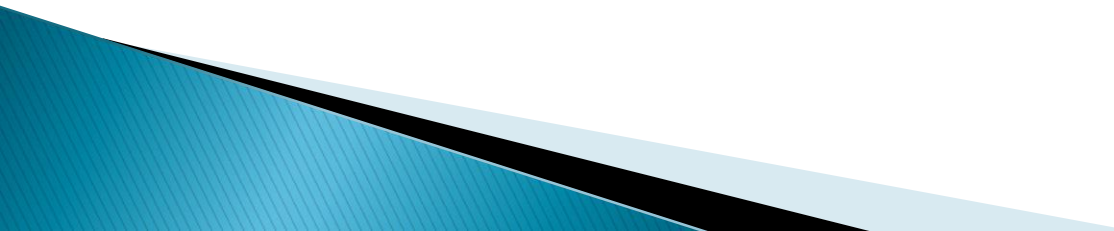
- ❖ <https://docs.python.org/3/> și <https://docs.python.org/3/tutorial/index.html>
- ❖ <https://www.w3schools.com/python/>
- ❖ M. Summerfield, **Programming in Python 3**  
<https://cs.smu.ca/~porter/csc/227/ProgrammingInPython3.pdf>
- ❖ Magnus Lie Hetland– **Beginning Python From Novice to Professional** – Apress (2017)
- ❖ Naomi Ceder – **The Quick Python Book** –Manning Publications, 3rd ed (2018)

# BIBLIOGRAFIE

- ❖ Jon Kleinberg, Éva Tardos, **Algorithm Design**, Addison–Wesley 2005  
<https://www.cs.princeton.edu/~wayne/kleinberg-tardos/>
- ❖ T.H. Cormen, C.E. Leiserson, R.R. Rivest – **Introducere in algoritmi**, Mit Press, trad. Computer Libris Agora
- ❖ S. Dasgupta, C.H. Papadimitriou, U.V. Vazirani, **Algorithms**, McGraw–Hill, 2008



# BIBLIOGRAFIE

- ❖ Horia Georgescu. **Tehnici de programare**. Editura Universității din București 2005
  - ❖ Leon Livovschi, Horia Georgescu. **Sinteza și analiza algoritmilor**. 1986
  - ❖ Dana Lica, Mircea Pașoi, **Fundamentele programării**, L&S Infomat
- 

# BIBLIOGRAFIE

- ❖ **coursera.org**

Algorithms, Part II – Princeton University

Algorithms: Design and Analysis – Stanford University

- ❖ **MIT** <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/>

- ❖ **infoarena.ro , pbinfo.ro**

# Materiale curs, laborator

- ❖ MoodleUB

- ❖ MS Teams Echipa

<https://tinyurl.com/PPSerial0>

Cod echipă: **0a3rv5z**

- ❖ Consultații

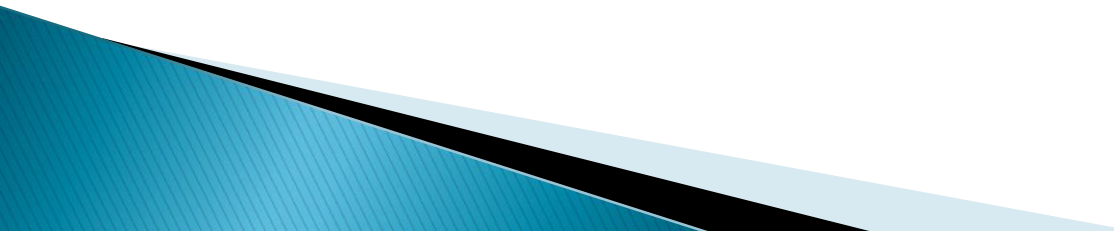
# Suita tehnologică



# Suita tehnologică utilizată în cadrul cursului și laboratorului

- <https://www.python.org/downloads/>
- mai multe variante pot coexista
- variabila de mediu PATH – bifați la instalare opțiunea **Add python to PATH**
- linie de comanda: comanda `python`
  - ❖ Medii de dezvoltare IDE : **PyCharm, Spyder** etc
  - ❖ Detalii instalare – laborator
  - ❖ Jupiter JupiterLab/ Notebook – online (de exemplu <https://jupyter.org/try-jupyter/lab/index.html> )

# Instalare Anaconda (cu Spyder, Jupiter Notebook)

- Pas1:** Accesați <https://www.anaconda.com/products/individual>, scrollați, alegeți sistemul de operare potrivit (Win / MacOS / Linux) și downloadați ultima versiune stabilă. Pentru Windows, cel mai probabil aveți nevoie de versiunea pe 64 de biți.
- P2:** Deschideți fișierul downloadat și urmați instrucțiunile afișate fără a bifa sau debifa nimic pe parcurs.
- P3:** În Windows, zona de search, scrieți “Anaconda Navigator” și deschideți programul găsit în urma căutării.
- P4:** Bifați opțiunea dorită în legătură cu transmiterea datelor și dați click pe butonul “Ok, and don’t ask me again”.
- P5:** Felicitări! Aveți access la [Jupyter Lab](#), [Jupyter Notebook](#), [Spyder](#) și alte aplicații conexe. Apăsăți pe butonul de Launch al oricăreia dintre cele trei enumerate anterior pentru a începe să lucrați cu acestea.
- 

# Instalare PyCharm

- P1:** Accesați <https://www.jetbrains.com/pycharm/download>, alegeți sistemul de operare potrivit (Win / MacOS / Linux) și downloadați versiunea **Community** (free, open source).
- P2:** Deschideți fișierul downloadat și urmați instrucțiunile afișate; bifați opțiunile “Create Associations” și “**Update PATH variable**”.
- P3:** În Windows, zona de search, scrieți “PyCharm” și deschideți programul găsit în urma căutării.
- P4:** Bifați opțiunea “Do not import settings”, alegeți tema preferată (Light sau Dark) și alegeți să instalați Markdown.
- P5:** Dați click pe “Create new project”, schimbați locația din “untitled” în “primul\_meu\_proiect” (sau orice altceva credeți potrivit), dați click pe drop-down-ul “Project Interpreter” și dați click pe butonul “Create”.
- 