

# SEMINAR 1 INFO

## 1. Partea administrativa

Pagina cursului: <https://www.cs.ubbcluj.ro/~istvanc/fp/>

Foldere:

- Curs: materiale curs
- Exemple: exemple pregătite de domnul profesor
- Lab: cerințe și reguli laborator

Teams: team dedicat disciplinei (veți găsi cod de team în primul curs - if you're here, you probably found it :))

Materiale seminar: channel Seminar - grupa XYZ, secțiunea Files (cod scris live la seminar/alte exemple)

## 2. Reguli seminar

- Prezențe: 10/14 prezențe pentru a intra în examen (atât în sesiunea normală, cât și cea de restanțe)
- Prezența din primele 2 săptămâni este acordată automat pentru activități
- Link la tabelul unde se pot urmări prezențele pe parcursul semestrului va fi postat pe channel în cursul primelor 2 săptămâni
- Participarea la seminar se va face cu grupa din care face parte studentul, nu sunt permise schimburi

## 3. Seminar 1

### 1. Python

Download: <https://www.python.org/downloads/> (orice versiune 3.x - ultima, 3.12, e în regulă)

La instalare, selectați **ADD Python to PATH** în prima fereastră!

Pentru a verifica versiunea instalată (Windows):

Deschideți command prompt

Executați instrucțiunea `python --version`

Primele lab-uri/seminarii: **Python IDLE**

### 2. Exerciții

1. Se citește de la tastatură un număr natural. Pentru numărul citit, determinați și afișați:

- Câte cifre are
- Suma cifrelor sale impare

2. Evidența temperaturilor dintr-o zi: se citesc de la tastatură 24 de valori reprezentând temperaturile înregistrate în fiecare oră a unei zile (prima oră: 00:00, ultima: 23:00). Temperaturile introduse sunt reprezentate de numere reale. **Rezolvare: ex2.py<sup>1</sup>**

A. Determinați temperatura minimă și cea maximă din ziua respectivă, precum și orele la care acestea se înregistrează.

B. Determinați media temperaturilor date

---

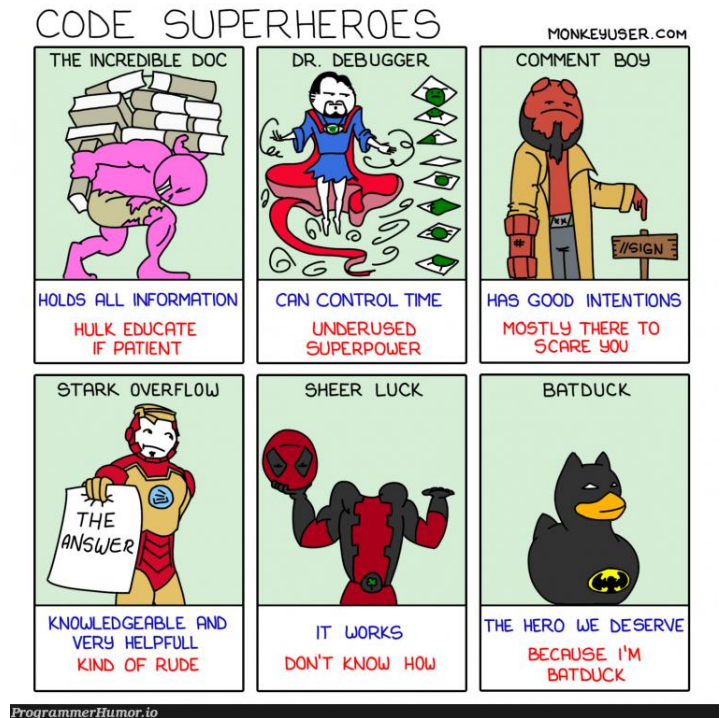
<sup>1</sup>În fișier există o variantă de implementare prin parcurgerea listei, care nu necesită utilizarea altor module/biblioteci, și una opțională (nu trebuie să o cunoașteți/folosiți), care folosește biblioteci adiționale - *numpy* (instalarea acestuia - și a oricărui modul dorit se poate realiza (momentan) prin deschidere command prompt - comanda `pip install nume-modul` (exemplu la finalul acestui document)

3. Se memorează datele despre o serie de răufăcători care dețin superputeri. Pentru fiecare răufăcător, sunt memorate *numele* și *superputerea* sa.

Aplicația care gestionează aceste date permite:

- A) Introducerea unui nume și afișarea profilului (superputerii) răufăcătorului cu acel nume de cod.
- B) Pentru un răufăcător cu nume dat, memorati o listă cu datele ultimelor incidente în care a fost implicat.
- C) [OPȚIONAL] Calculați câte zile au trecut de la ultimul incident pentru fiecare din răufăcătorii din baza de date.

Rezolvare: `ex3.py`<sup>2</sup>



### KEEP IN MIND (în general):

- ✓ Folosiți nume sugestive pentru variabile (think on larger scale - dacă avem multe linii de cod de parcurs, vrem să fie ușor inteligibil, să citim aproape ca pe o poveste, nu să ne oprim la fiecare variabilă să încercăm să ne dăm seama ce reprezintă)
- ✓ Constanță în modul de a scrie codul într-un fișier/aplicație
  - ...constanță care lipsește în implementările exemplu într-o anumită măsură :) - sunt incluse diverse variante de printare, parcurgeri, etc pentru a cuprinde mai multe moduri de a scrie cod care face același lucru - în general, ar fi bine să alegem 1 variantă și o folosim pe aceea unde este potrivit

Proces de dezvoltare incrementală bazată pe funcționalități

- 1) Enunț problemă (clarificare aspecte enunț, dacă e cazul)
- 2) Identificarea funcționalităților care trebuie implementate
- 3) Planificarea iterațiilor (în cazul în care există un număr mai mare de funcționalități)
- 4) Start iterație - pentru fiecare funcționalitate din iterație:
  - a) [MODELARE] Descriem scenariul de utilizare
  - b) Identificăm task-uri/activități prin intermediul cărora implementăm efectiv funcționalitatea (necesar în special când funcționalitățile sunt mai complexe)
  - c) Scriem teste (deocamdată tabele cu date de intrare - date de ieșire așteptate)
  - d) (finally...) Implementare

Exemplu - Problema 1:

Funcționalități:

<sup>2</sup> Primele 2 cerințe vizează folosirea tipului *dictionar*, parcurgerea lui, cum putem memora informații într-un dictionar, cum putem afișa elementele din dictionar, etc; a treia cerință este un exemplu de folosire a unui modul pentru ușurința calculului cu date, și parcurgerea rezolvării sau folosirea ulterioară a modulului este opțională

**F1:** Determinare număr cifre al unul număr dat

**F2:** Determinare suma cifrelor impare a numărului dat

PLAN DE ITERAȚII:

|    |        |
|----|--------|
| I1 | F1, F2 |
|    |        |
|    |        |

OBS. Dacă aveam un număr mai mare de funcționalități, puteam să ne planificăm iterația 1 pe seminarul 1, iterația 2 în seminarul 2 etc

ITERAȚIE (definiția din curs): perioadă de timp în cadrul căreia se realizează o versiună stabilă și executabilă a unui produs (împreună cu documentația suport)

**F1:** Determinare număr cifre al unul număr dat

SCENARIU RULARE

| Utilizator | Program                  | Descriere  |
|------------|--------------------------|--|
|            | Introduceți numărul:     | Programul tipărește un mesaj informativ pentru utilizator în vederea introducerii unui număr |
| 164        |                          | Utilizatorul introduce un număr  |
|            | Numărul 164 are 3 cifre. | Programul tipărește numărul de cifre al numărului introdus de utilizator.                    |

Deocamdată, nu avem un meniu prin care să oferim utilizatorului posibilitatea de a repeta operația de mai multe ori (citirea mai multor numere și afișarea numărului de cifre pentru fiecare), astfel că în acest caz, **scenariul de rulare** se încheie după afișarea rezultatului.

LISTĂ ACTIVITĂȚI

|               |   |
|---------------|---|
| <b>TASK 1</b> | Determinare număr cifre pentru valoarea introdusă |
| <b>TASK 2</b> | Implementare interfață utilizator                 |

CAZURI DE TEST

| Date de intrare | Date de ieșire |
|-----------------|----------------|
| 123             | 3              |
| 2               | 1              |
| 1000            | 4              |
| 0               | 1              |

IMPLEMENTARE

ex1.py

Se repetă procesul de documentare (scenariu de rulare, listă de activități, cazuri de test) pentru funcționalitatea 2, și apoi se implementează și aceasta.

---

Microsoft Windows [Version 10.0.22621.2283]  
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>pip install numpy

Collecting numpy

Obtaining dependency information for numpy from [https://files.pythonhosted.org/packages/2a/4e/0b/0e/27504c68ef6e808ebaa90ee95902/numpy-1.26.0-cp312-cp312-win\\_amd64.whl.metadata](https://files.pythonhosted.org/packages/2a/4e/0b/0e/27504c68ef6e808ebaa90ee95902/numpy-1.26.0-cp312-cp312-win_amd64.whl.metadata)

Downloading numpy-1.26.0-cp312-cp312-win\_amd64.whl.metadata (61 kB)

61.1/61.1 kB 1.6 MB/s eta 0:00:00

Downloading numpy-1.26.0-cp312-cp312-win\_amd64.whl (15.5 MB)

15.5/15.5 MB 3.6 MB/s eta 0:00:00

Installing collected packages: numpy

Successfully installed numpy-1.26.0

C:\Users\DELL>python --version

Python 3.12.0

C:\Users\DELL>