

# Tehnici Avansate de Programare

Recapitulare și noțiuni de bază

---

Petru Rebeja, Marius Apetrii

25 Februarie 2021

Facultatea de Matematică

Universitatea Alexandru Ioan Cuza, Iași

# Introdurre

---

## Ce am discutat data trecută

# Ce am discutat data trecută

- Ciclul de dezvoltare al aplicațiilor
- Dezvoltarea în iterații
- Unelte de lucru
- Fluxul de lucru Git
- Bune practici

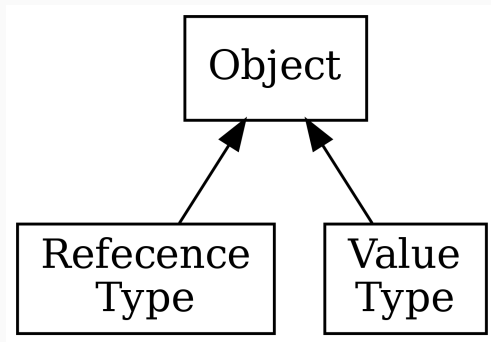
# Ce vom discuta azi

- Tipuri de date în `.net`
- Principiile programării orientată-obiect

# Tipuri de date în .net

---

## Tipuri referință și tipuri valoare



# Tipuri referință și tipuri valoare

## Exemple

Tip Referință	Tip Valoare
<code>string</code>	<code>int</code>
<code>System.Object</code>	<code>float</code>
<code>System.Action</code>	<code>bool</code>
<code>System.IO.MemoryStream</code>	Tipuri struct
<code>System.Collections.Generic.List&lt;T&gt;</code>	Tipuri enum



- O variabilă a unui tip valoare conține o instanță a tipului respectiv.
- La crearea unei instanțe se alocă o singură zonă în memorie.
- Dimensiunea zonei alocate este dată de tip.

## Tipuri referință

- O variabilă a unui tip referință conține adresa de memorie unde este alocată instanța propriu-zisă.
- La crearea unei instanțe se alocă două zone de memorie:
  - O zonă pentru adresă (stivă),
  - O zonă pentru instanță (memoria heap).
- Dimensiunea zonei alocate pentru instanță poate să varieze (ex. `List<T>`).

## Cazul special: `string`

- Deși tipul `string` este un tip referință, acesta se comportă ca un tip valoare.
- Această proprietate se numește *imuabilitate*

# Principiile Programării Orientate-Obiect

---

- **Încapsularea** — îmbinarea datelor și metodelor care le procesează.
- **Moștenire** — proprietățile tipului părinte se păstrează și la copii.
- **Polimorfism** — același comportament manifestat de mai multe tipuri.

## Context

Pentru exemplificare vom modela o serie de operațiuni bancare.

## Cerință I

În calitate de posesor al unui cont bancar vreau să pot depune și extrage diverse sume de bani din cont.

În situația de față, **încapsularea** ne permite să îmbinăm retragerea de fonduri cu verificarea dacă sunt fonduri suficiente.



## Cerință II

În calitate de posesor de conturi, pot avea conturi de diferite tipuri: economii și debit. Pentru retragerea de fonduri din fiecare cont se percep comisioane diferite în funcție de tipul acestuia:

- Pentru contul de debit: 0 %
- Pentru contul de economii: 0.5 %

## Cerință III

În calitate de posesor de conturi pot avea un cont de credit cu un comision de retragere de 0.7%

În exemplul dat **moștenirea** ne permite să declarăm o clasă de bază cu proprietățile comune și să implementăm particularitățile în clasele derivate.

Polimorfismul ne permite să aplicăm reguli de calcul diferite pentru aceeași metodă.

# Înceiere

---

- Tipuri de date în `.net`.
- Principiile programării orientată-obiect.

Vă mulțumesc!

Mulțumesc pentru atenție!