


# PROGRAMOZÁS 2

5.hét

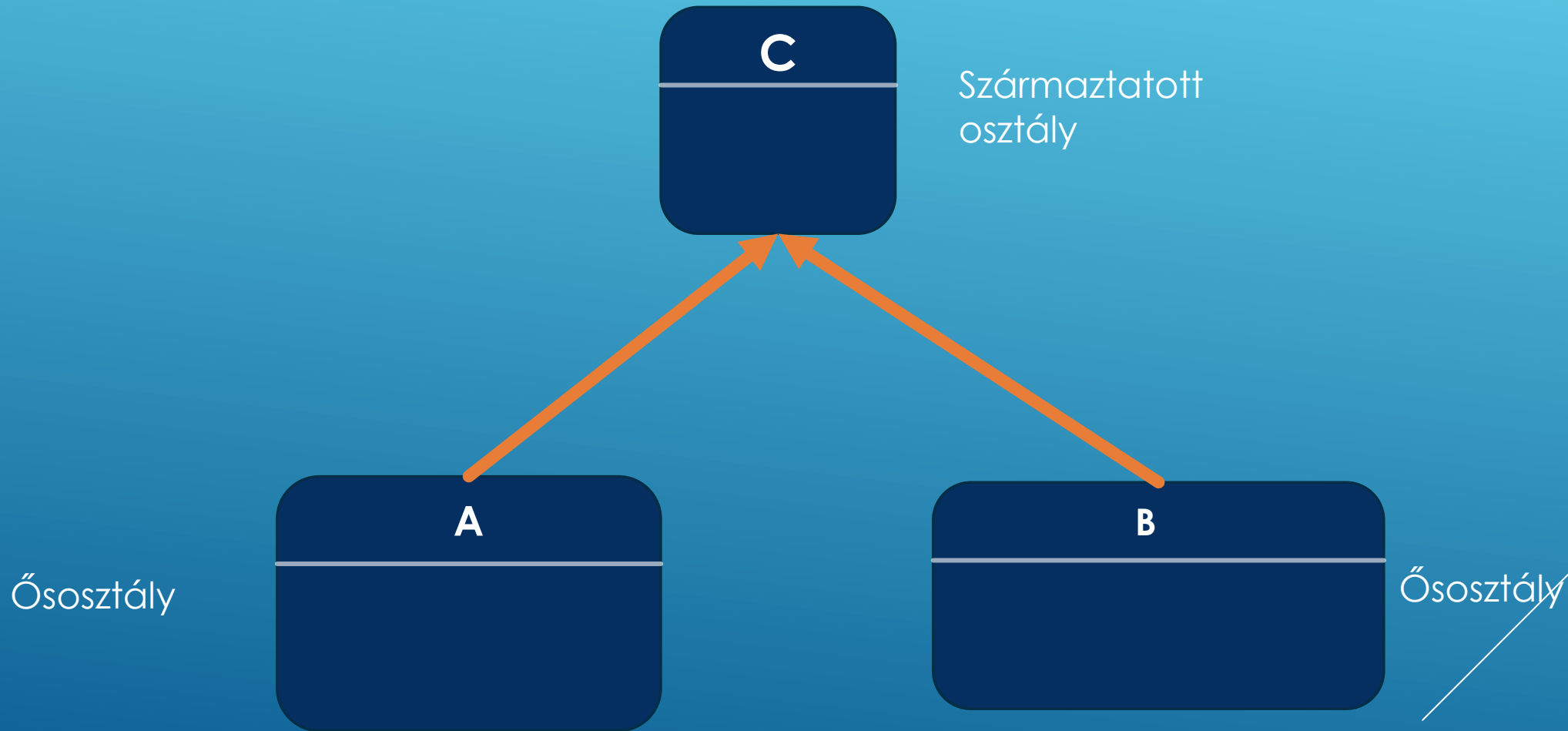
# ÁTTEKINTÉS

- ▶ Többszörös öröklődés
  - ▶ Beágyazott osztályok, láthatóság
  - ▶ Feladatok
- 
- A decorative graphic consisting of several parallel white lines of varying lengths, slanted upwards from left to right, located in the bottom right corner of the slide.

# TÖBBSZÖRÖS ÖRÖKLŐDÉS

- ▶ Pythonban lehetőség van arra, hogy a származtatott osztálynak több őszülője is legyen -> generalizáció
- ▶ A láthatóság és elérés ugyanúgy igaz lesz, mint a sima öröklődés esetében.
- ▶ MRO (multiple-resolution order): itt azt jelenti, hogy a meghívott metódust milyen sorrendben fogja keresni az interpreter az osztályok között.

# TÖBBSZÖRÖS ÖRÖKLŐDÉS



# BEÁGYAZOTT OSZTÁLY

- ▶ Lehetőség van arra is, hogy egy osztály egy másik osztályban változóként megjelenjen.
- ▶ Akkor alkalmazzuk, amikor egyik része a másiknak pl.: egy autónak van kereke, motorja stb.
- ▶ A beágyazott osztályt az azt tartalmazó osztályban példányosítjuk.
- ▶ A láthatóság megegyezik azzal, amikor kívülről (mainből) érjük el.

# ÖRÖKLŐDÉS VS BEÁGYAZOTT OSZTÁLY

- ▶ Mikor melyiket használjuk?
- ▶ Ilyenkor azt érdemes eldönteni, hogy az adott probléma esetében a másik osztály önmagában is nevezhető az eredeti osztálynak (típusa/fajtája-e), vagy alkotórésze.
  - ▶ Példa: Az AUDI A6, Mercedes C osztály önmagában is autók, tehát ilyenkor öröklődést használunk.
  - ▶ Példa: A motor és egyéb alkatrészek önmagukban nem adnak ki egy egész autót, viszont minden autó tartalmazza ezeket (jobb esetben 😊 ). Ilyenkor ezek beágyazott osztályok lesznek, mert részei az autónak.