

Opjektum Orientált Programozás

POLIMORFIZMUS

Készítette: Vastag Atila

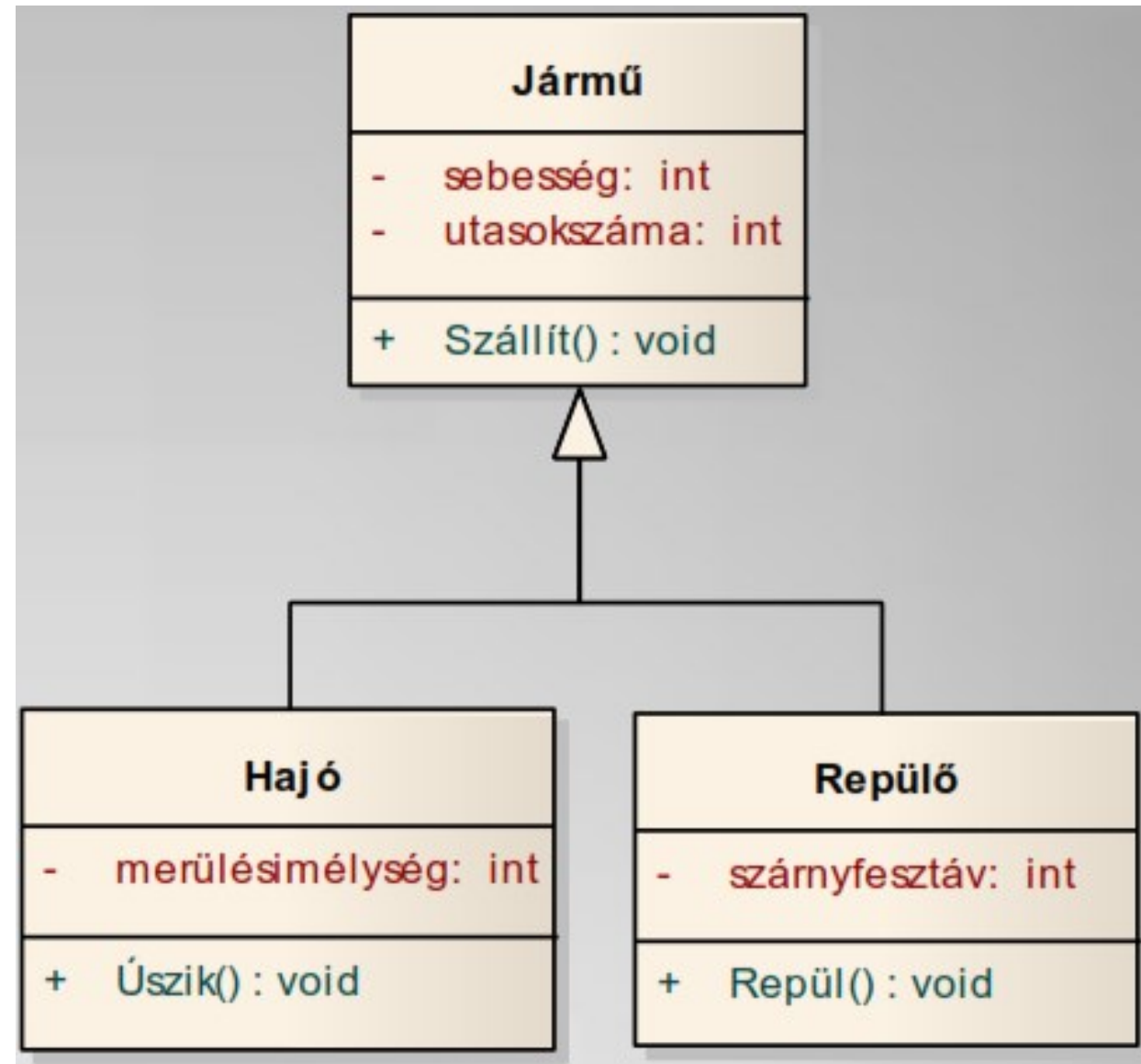
2020

Öröklés fogalma

- Egy létező osztályból egy (vagy több) újat készítünk
- A létező osztályt ősnek, az újakat utódnak nevezzük
- Az utódok öröklik az ős minden tagját
- Az utódok az öröklött tagokat újakkal bővíthetik
- Az öröklés használata egyszerűbbé és átláthatóbbá teszi a programot

Öröklési szemléletmódok

- Specializáció:
Az osztályt (általános) készítjük el először, ebből pedig az utódokat (speciális).
- Generalizáció:
Az utódokat készítjük el először, majd ezekből a közös tagok kiemelésével az őst.



- Minden osztály egyszerre lehet ős és utód is
- Ebből egy osztályhierarchia alakul ki, amelyben az osztályok fát alkotnak

```

class Jarmu:
    def __init__(self):
        super().__init__()

        self.sebesseg : int = 0
        self.utasokSzama : int = 0

    def szállit(self) -> None:
        raise NotImplementedError("A leszármazott osztály nem implementálja ezt az absztract metódust!")

```

```

from jarmu import Jarmu

```

```

class Hajo (Jarmu):
    def __init__(self):
        super().__init__()
        Jarmu.__init__(self)

        self.merulesiMelyseg : float = 0

    def uszik(self) -> None:
        print("A hajó uszik.")

    def szállit(self) -> None:
        print(f"A hajó {self.utasokSzama} utast szállít.")

```

```

from jarmu import Jarmu

```

```

class Repulo (Jarmu):
    def __init__(self):
        super().__init__()
        Jarmu.__init__(self)

        self.szarnyfesytav : int = 0

    def repul(self) -> None:
        print("A repülő repül.")

    def szállit(self) -> None:
        print(f"A repülő {self.utasokSzama} utast szállít.")

```

Polimorfizmus

- Őstípusú referenciaváltozó hivatkozhat ős-, és utód típusú objektumra is

```
hajo: Hajo = Hajo();  
repulo: Repulo = new Repulo();
```

```
hajo: Jarmu = Hajo();  
repulo: Jarmu = Repulo();
```

- Fordítva nem igaz

```
hajo: Hajo = Jarmu();  
repulo: Repulo = Jarmu();
```

Ilyet nem lehet, mivel az öröklött osztály kiterjesztett (többet tud) változata az ős osztálynak.

Ha ezt lehetne, akkor a egy „*buta*” hajót kapnánk, vagy repülőt amely csak egy „egyszerű” jármű lenne

EXPLORER

OPEN EDITORS

× main.py
jarmu.py
hajo.py
repulo.py

PYTHON

> __pycache__
hajo.py
jarmu.py
main.py
repulo.py

main.py × jarmu.py hajo.py repulo.py

main.py > ...

```
1  #main.py
2
3  from jarmu import Jarmu
4  from hajo import Hajo
5  from repulo import Repulo
6
7  hajo: Jarmu = Hajo()
8  hajo.utasokSzama = 100
9
10 repulo: Jarmu = Repulo()
11 repulo.utasokSzama = 50
12
13 hajo.szallit()
14 repulo.szallit()
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

A hajó 100 utast szállít.
A repülő 50 utast szállít.