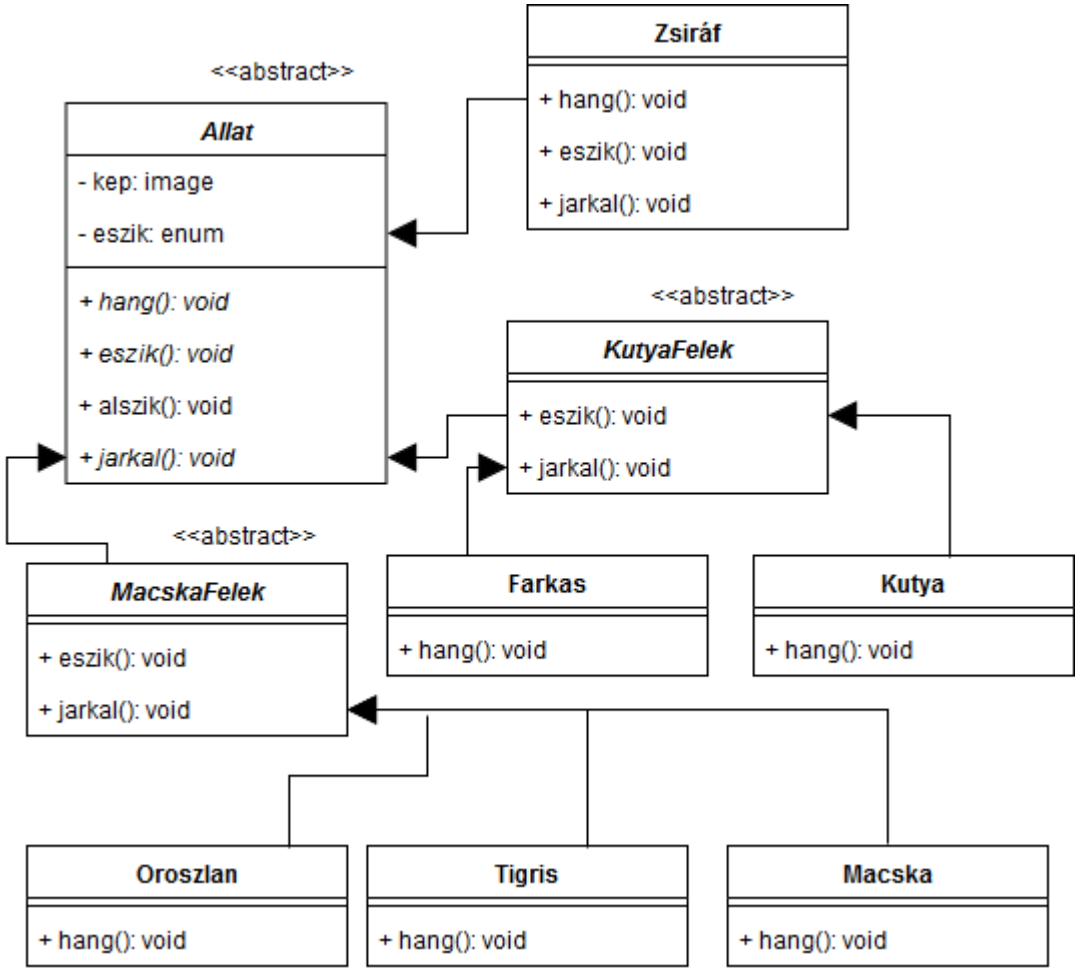
Állatfarm 3

- Egészítsük ki *Kedvenc* programmal:
- A Kutya tud Kedvenc lenni, de:
- Nem csak a Kutya viselkedhet Kedvenc módra
- A Kutya nem használható pl.: *Vad* programban

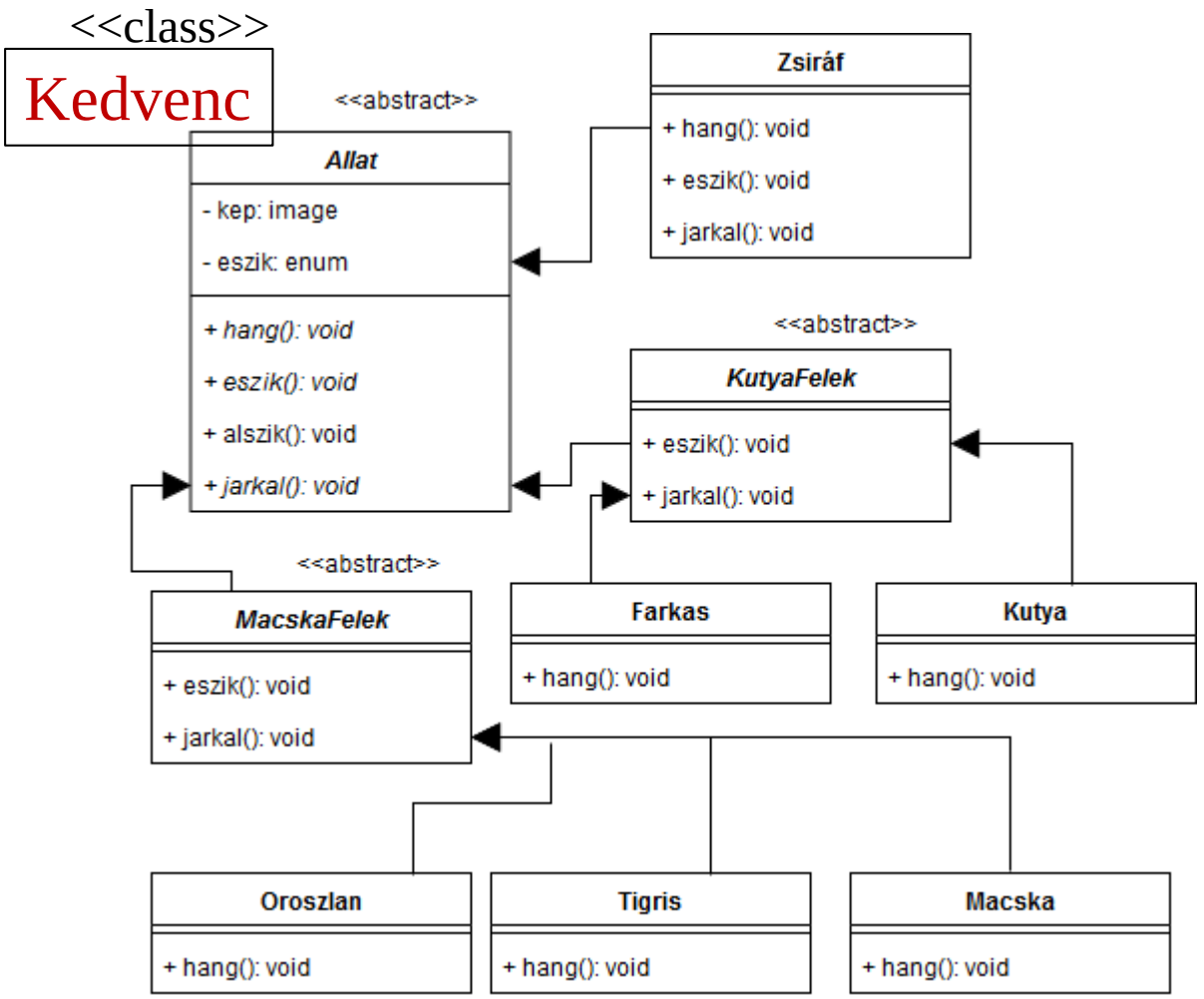


Állatfarm 3 – lehetőségek?

1. Konkrét megvalósítással Allat osztályban

Előny, hátrány?

Oroszlán nem Kedvenc, Kutya és Macska máshogy

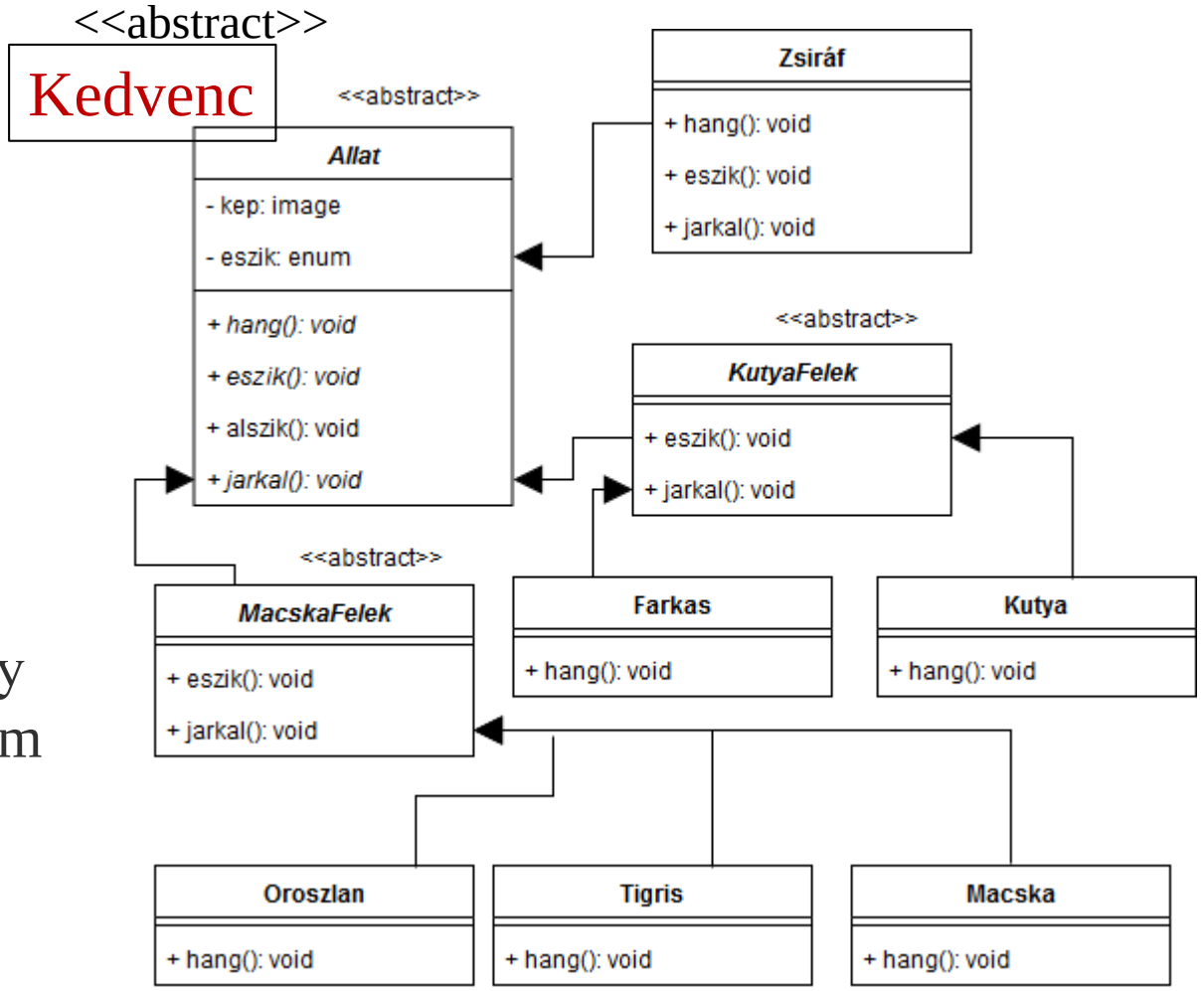


Állatfarm 3 – lehetőségek?

2. Absztrakt megvalósítással Allat osztályban

Előny, hátrány?

Oroszlán nem lesz Kedvenc, de tud így viselkedni, csak nem csinál rá semmit



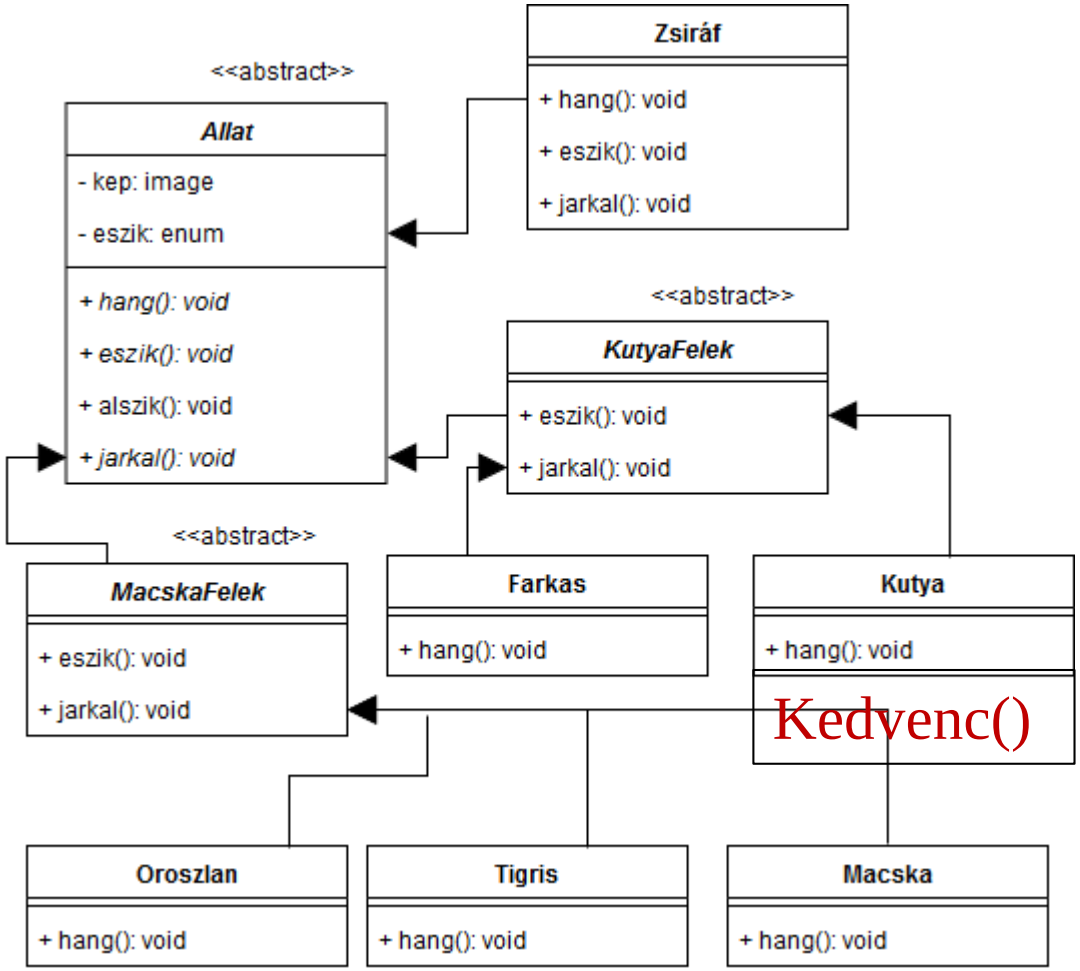
Állatfarm 3 – lehetőségek?

3. Tagfüggvényeket ahol kell, pl.: Kutya osztályban

Előny, hátrány?

Macska ugyan ezt kell tudja, 2 programozó hogy oldja meg?

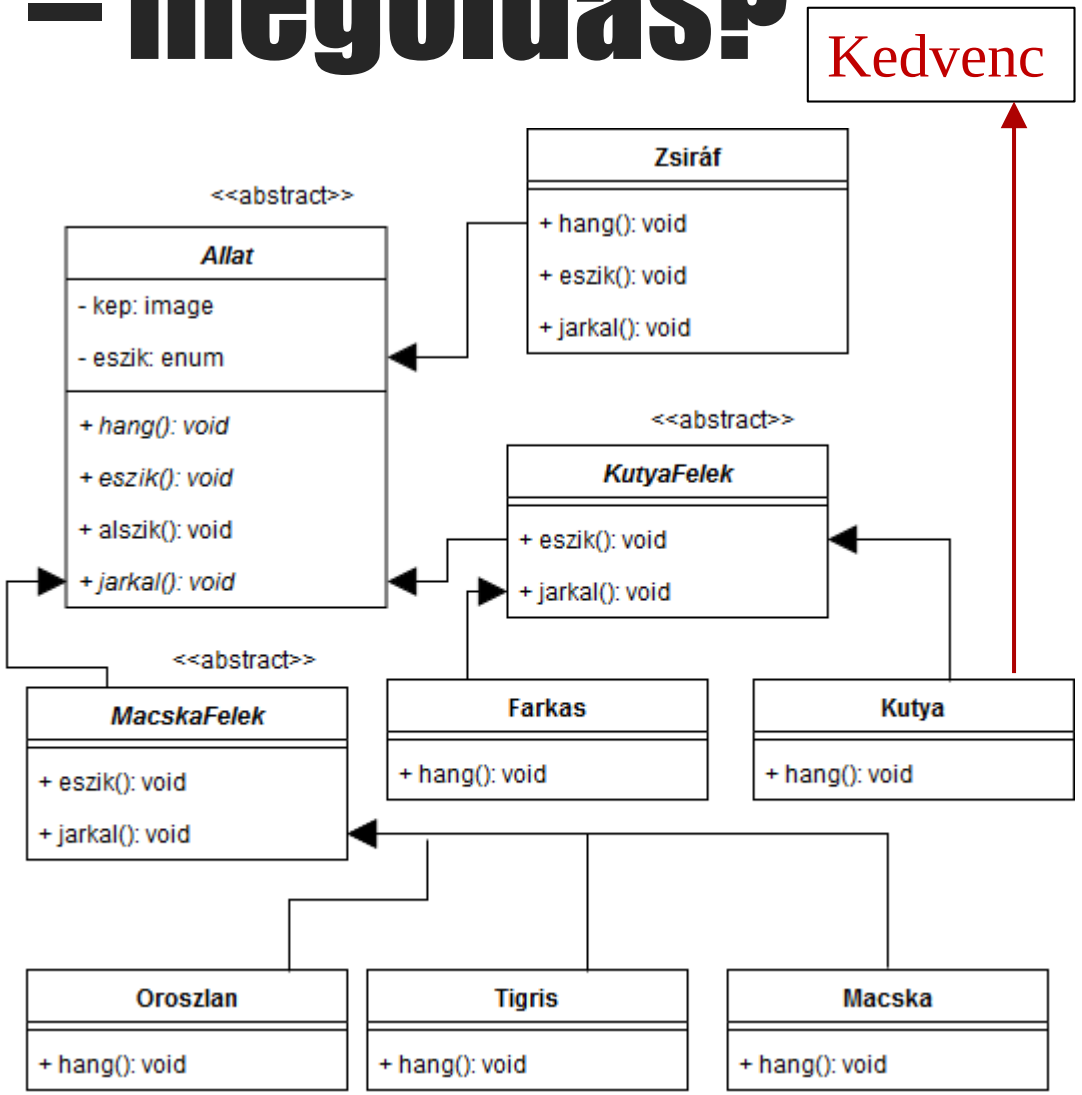
Az Allat nem lehet polimorf típus, nincs statikus típusában kedvenc(), hiába Kutya a dinamikus típus



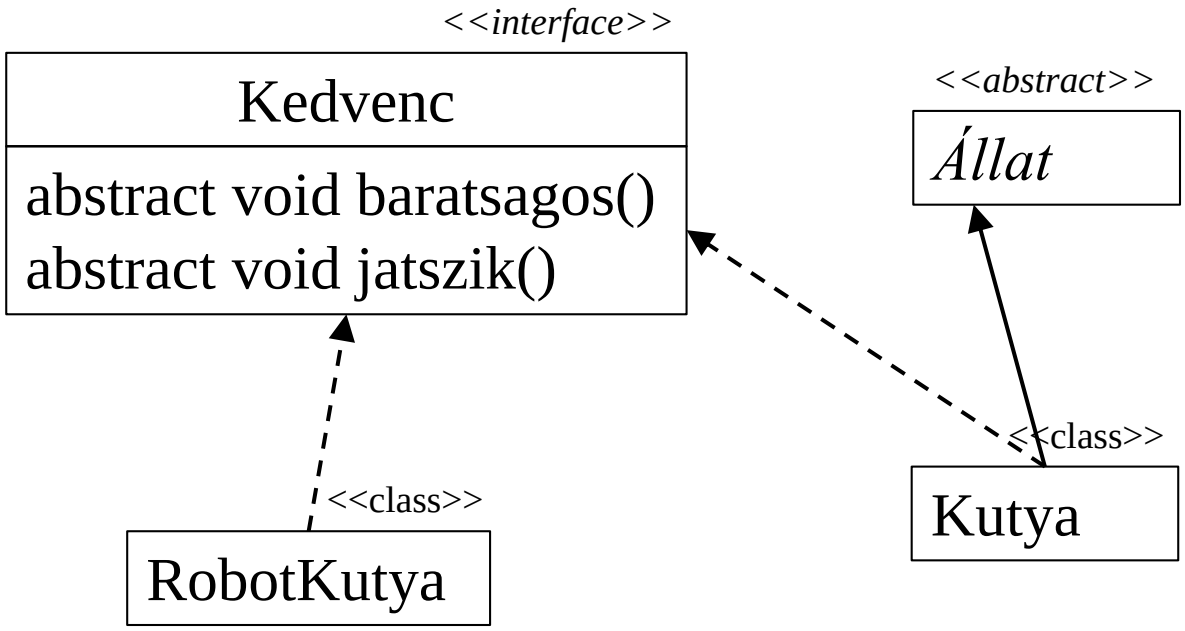
Állatfarm 3 – megoldás?

Csak házikedvenc osztályok mutassanak kedvenc *viselkedést*!

- **Minden kedvenc osztály ugyanazt mutassa**
- **Lehessen használni a többalakúságot**



Többszörös öröklés vs. interface



Alosztály: egy osztály konkrétabb változata kell
Elvont osztály: ha egy sablon kell
Felület: ha szerepet kell meghatározni

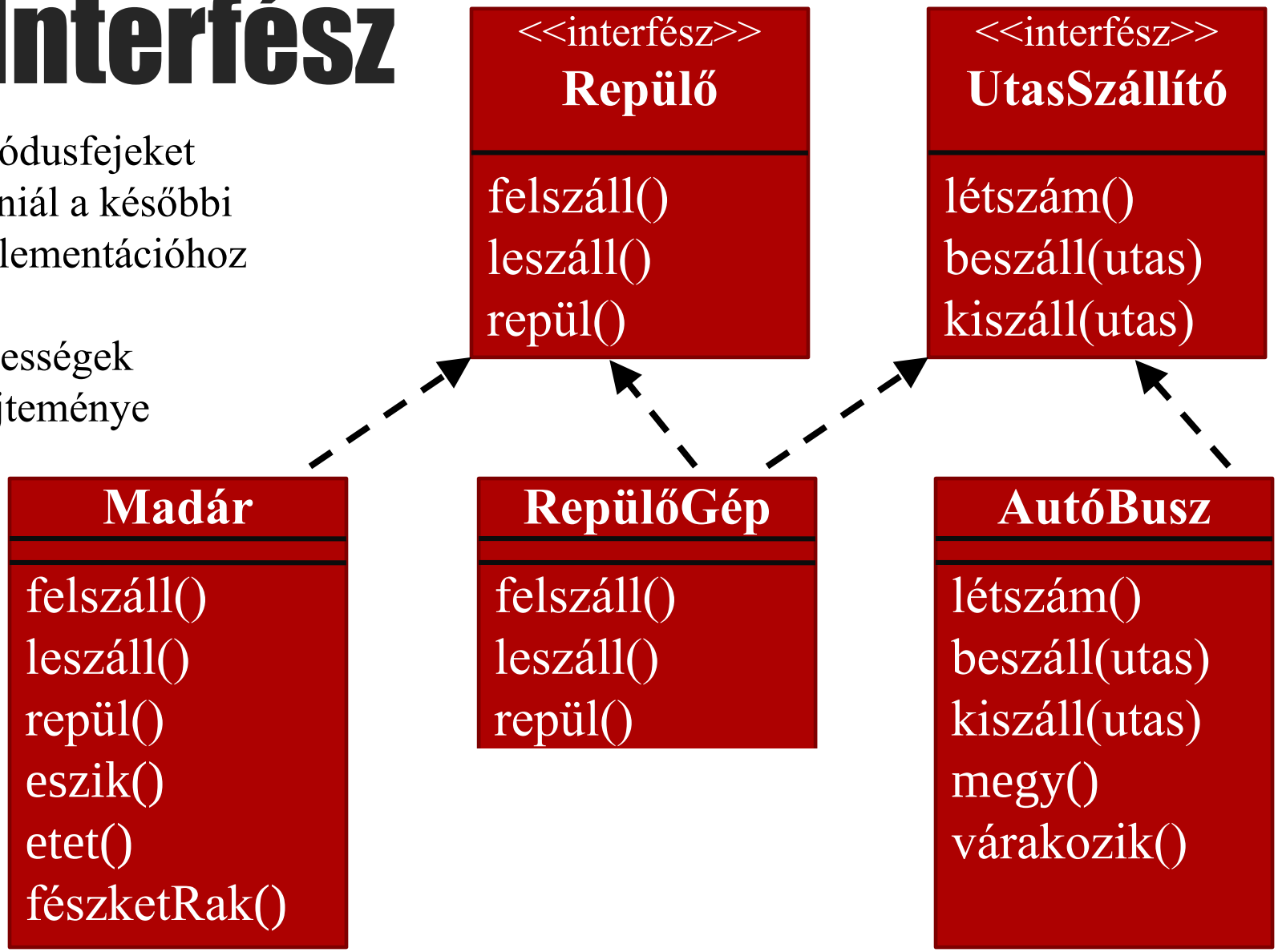
Interfész

- C#: konvencionálisan `INev` a típus neve
- Alapértelmezetten *abstract*, nem lehet példányosítani
- Csak **public** lehet (elhagyásakor csomagszintű)
- Konstansok: **public**, **static** és **final**, akár deklaráljuk, akár nem
  - Java, de C# nem engedi ->Properties
- Metódusfejek: **public**, és **abstract**, akár deklaráljuk, akár nem pontosvesszővel zárjuk, mint absztraktnál, törzs sincs
- Az összes megadott metódust implementálni kell
- *Abstract* osztálynál az utódban kell implementálni
- Több **interface** is implementálható 
- **Interface-**ek öröklődhetnek: 

Interfész

Metódusfejeket definiál a későbbi Implementációhoz

Képességek gyűjteménye



Implementáló osztályok