

|                                              |
|----------------------------------------------|
|                                              |
| <h2>Objektum-orientált tervezési minták</h2> |

---

---

---

---

---

---

---

|                                                                                                                                                                                                                                                                                                                                                                                        |  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <b>OOP tervezési minták</b>                                                                                                                                                                                                                                                                                                                                                            |  |
| <ul style="list-style-type: none"><li>■ Tipikus de különböző kontösben visszatérő problémák objektum-orientált modelljei</li><li>■ Példák tervezési mintákra (design patterns):<ul style="list-style-type: none"><li>• Sablonmetódus</li><li>• Gyártó metódus</li><li>• Iterátor</li><li>• Toldalék</li><li>• Homogén összetétel</li><li>• Stratégia</li><li>• ...</li></ul></li></ul> |  |
| 2 / 36                                                                                                                                                                                                                                                                                                                                                                                 |  |

---

---

---

---

---

---

---

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <b>Tervezési minták leírása</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |  |
| <ul style="list-style-type: none"><li>■ <b>Minta neve:</b> tömören és találóan leírja a minta koncepcionális tartalmát</li><li>■ <b>Alapötlet:</b> probléma típusa, mi a megoldás célja</li><li>■ <b>Esettanulmány:</b> alkalmazási mintapélda tervezésének leírása</li><li>■ <b>Alkalmazhatóság:</b> a minta milyen helyzetekben alkalmazható, mozgástér, kompromisszumok, kötöttségek</li><li>■ <b>Felelősségek és együttműködés:</b> tervezési minta osztályai, felelősségek hozzárendelése, együttműködés sémája</li></ul> |  |
| 3 / 36                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |  |

---

---

---

---

---

---

---

## Sablonmetódus

### ■ Alapötlet:

- egy művelet megvalósításának vázát az őosztályban absztrakt metódusok meghívásával definiáljuk
- az absztrakt metódusokat a származtatott osztályban implementáljuk

### ■ Esettanulmány

- pl. az `AbstractList` osztály `indexOf()` művelete sablonmetódus
- a `get()` és a `size()` absztrakt műveleteket a származtatott `ArrayList` osztályban implementáljuk

4 / 36

---

---

---

---

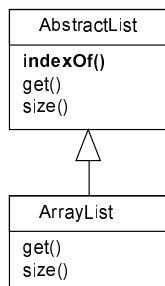
---

---

---

---

## Sablonmetódus



### ■ `indexOf()`

- definiálja egy adott elem megkeresésének algoritmusát
- ehhez szükség van a `get()` és a `size()` metódusokra, melyek implementálását a származtatott `ArrayList` osztályra bizza

5 / 36

---

---

---

---

---

---

---

---

## Alkalmazhatóság

- A sablonmetódust kódmegosztásra és elvonatkoztatásra használjuk
- A származtatott osztályok hasonló algoritmusainak vázát emeljük ki az őosztályba
- A primitív műveletek implementálását a konkrét származtatott osztályokra hárítjuk

6 / 36

---

---

---

---

---

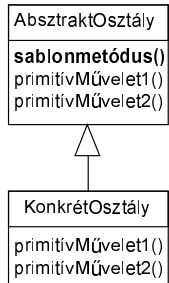
---

---

---

## Felelősségek és együttműködés

- **AbsztraktOsztály**
  - implementálja az algoritmus vázát mint sablonmetódust
  - deklarálja az absztrakt elemi műveleteket
- **KonkrétOsztály**
  - az algoritmus vázát az AbsztraktOsztálytól örökli
  - megadja az elemi műveletek implementációját



7 / 36

---

---

---

---

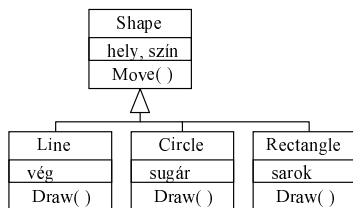
---

---

---

## Alkalmazási példa

- **Move():** sablonmetódus, absztrakt draw() metódusokat hív meg
- **Draw():** primitív művelet, melyet a származtatott osztályok implementálnak



8 / 36

---

---

---

---

---

---

---

## Gyártó metódus

- **Alapötlet**
  - a példányosítást egy absztrakt metóduson keresztül végezzük
  - a példányosítandó osztály kiválasztását nem kell a példányosítást igénylő kódra bízni

9 / 36

---

---

---

---

---

---

---

## Gyártó metódus

### ■ Esettanulmány

- minden Collection interfészt implementáló adatszerkezet bejárásához szükség van egy Iterator interfészt implementáló objektumra
- az iterator objektumot a Collection.iterator() **gyártó metódus** hozza létre
- az iterator objektum konkrét típusát nem kell ismernünk a példányosításhoz
- a Collection.iterator() implementációja ugyanakkor az adatszerkezetet reprezentáló osztály felelősége

10 / 36

---

---

---

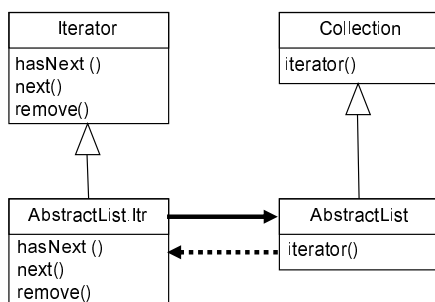
---

---

---

---

## Esettanulmány



11 / 36

---

---

---

---

---

---

---

## Alkalmazhatóság

- A példányosítást kezdeményező osztály nem akarja, vagy nem képes a példányosítandó osztályt rögzíteni
- Függetleníteni lehet egy interfész klienseit az interfész implementációjának kiválasztásától
- Gyakran párhuzamos típushierarchiákat köt össze

12 / 36

---

---

---

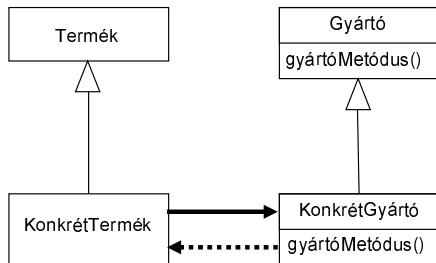
---

---

---

---

## Felelősségek és együttműködés



13 / 36

---

---

---

---

---

---

---

## Felelősségek és együttműködés

- Termék
  - definiálja a gyártó metódus által példányosított osztályok őstípusát
- KonkrétTermék
  - implementálja a Termék interfészt
- Gyártó
  - deklarálja a gyártó metódust, melynek visszatérési értéke a Termék interfész implementációja
- KonkrétGyártó
  - implementálja a gyártó metódust

14 / 36

---

---

---

---

---

---

---

## Iterátor

- Alapötlet
  - adatszerkezettől és absztrakt adattípustól független interfész aggregátumok bejárásához
- Esettanulmány
  - az adatszerkezet bejárásának mechanizmusát egy olyan független iterátor osztályba emeljük ki, amely implementál egy közös Iterator interfészt

15 / 36

---

---

---

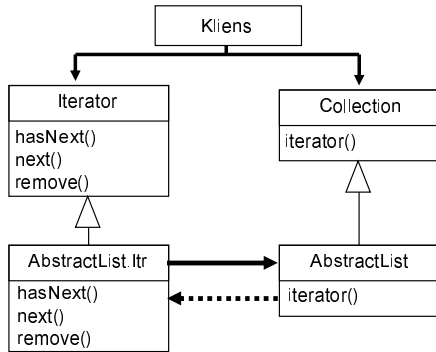
---

---

---

---

## Esettanulmány



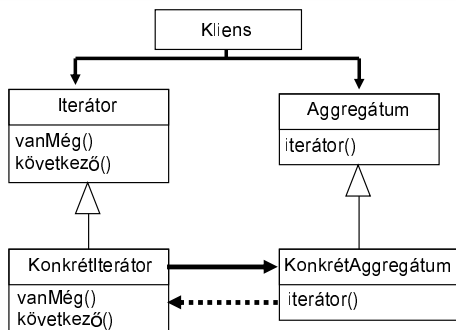
16 / 36

## Alkalmazhatóság

- A bejárásához szükséges interfészt és implementációját függetleníthetjük az adatszerkezettől
- Közös őssel nem rendelkező aggregátumokat is egyazon interfésszel tudunk bejárni
- Az iterátor belső állapota független az aggregátumtól, így egyszerre több bejárás is folyamatban lehet
- Egy aggregátum többféle iterációt is támogat (pl. lista esetén előlről hátra vagy hátulról előre)

17 / 36

## Felelősségek és együttműködés



18 / 36

## Felelősségek és együttműködés

- Iterátor
  - definiálja az adatszerkezet bejárásához szükséges interfészt
- Konkrétiterátor
  - implementálja az Iterátor interfészt
  - tárolja az iteráció aktuális állapotát
- Aggregátum
  - deklarálja az iterátort gyártó absztrakt metódust
- KonkrétAggregátum
  - implementálja a gyártó metódust, amely létrehozza a Konkrétiterátor osztály egy példányát

19 / 36

---

---

---

---

---

---

---

## Toldalék

- Alapötlet
  - a funkcionális kiterjesztést nem örökléssel, hanem objektumkompozícióval és delegációval valósítjuk meg
  - a kiterjesztések egymással dinamikusan kombinálhatók

20 / 36

---

---

---

---

---

---

---

## Toldalék

- Esettanulmány
  - input/output kezelése Jávában csatornákon keresztül történik
  - csatorna felelősségei:
    - fizikai tárolási szerkezet (pl. fájl vagy memória) hozzárendelése
    - pufferek biztosítása
    - elemi típusok kódolása/dekódolása
  - a csatorna különböző felelősségei külön osztályokhoz vannak hozzárendelve
  - a csatorna bizonyos műveleteket nem maga hajt végre, hanem azokat egy alárendelt csatornához delegálja

21 / 36

---

---

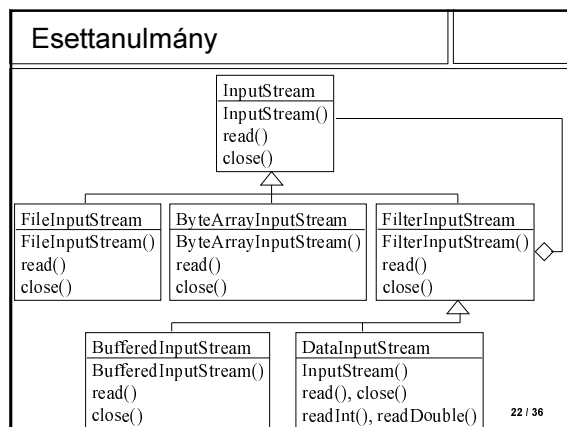
---

---

---

---

---




---

---

---

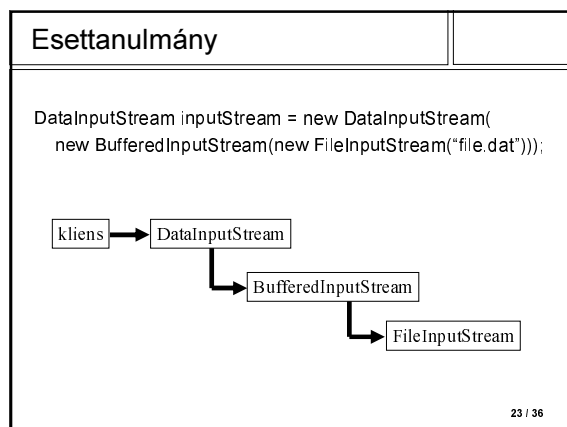
---

---

---

---

---




---

---

---

---

---

---

---

---

| Alkalmazhatóság                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <ul style="list-style-type: none"> <li>■ Egy osztály többféle, egymástól független kiterjesztésére van szükség, amelyeket még kombinálni is szeretnénk</li> <li>■ A toldalékos megoldás előnyei                         <ul style="list-style-type: none"> <li>• toldalék több osztály példányaihoz is kapcsolható<br/>⇒ kvázi többszörös öröklődés</li> <li>• toldalékok kombinálhatók</li> <li>• a toldalékot dinamikusan kapcsoljuk az objektumhoz</li> <li>• a toldalékok interfészen keresztül kapcsolódnak a kiterjesztett osztályhoz ⇒ függetlenül módosíthatók</li> </ul> </li> </ul> |  |
| 24 / 36                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |  |

---

---

---

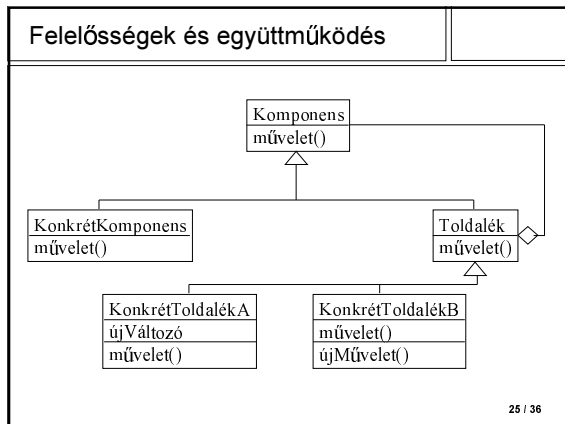
---

---

---

---

---




---

---

---

---

---

---

---

---

- |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <b>Felelősségek és együttműködés</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |  |
| <ul style="list-style-type: none"> <li>■ <b>Komponens</b> <ul style="list-style-type: none"> <li>• definiálja a todalékkal bővíthető objektumok közös interfészét</li> </ul> </li> <li>■ <b>KonkrétKomponens</b> <ul style="list-style-type: none"> <li>• definiál egy konkrét osztályt, melynek viselkedése todalékokkal kiegészíthető</li> </ul> </li> <li>■ <b>Toldalék</b> <ul style="list-style-type: none"> <li>• egy komponens objektumhoz kapcsolódik, amelyhez delegálja a műveleteket</li> <li>• maga is kompatibilis a Komponens interfésszel</li> </ul> </li> <li>■ <b>KonkrétToldalék</b> <ul style="list-style-type: none"> <li>• új viselkedéssel terjeszti ki a Komponens interfész implementációit</li> </ul> </li> </ul> |  |
| 26 / 36                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |  |

---

---

---

---

---

---

---

---

- |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <b>Homogén összetétel</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |  |
| <ul style="list-style-type: none"> <li>■ <b>Alapötlet</b> <ul style="list-style-type: none"> <li>• homogén rész-egész hierarchia elemeit egységes interfészen keresztül kezeljük</li> <li>• elemi és összetett objektumokat összefonó polimorfizmus</li> </ul> </li> <li>■ <b>Esettanulmány</b> <ul style="list-style-type: none"> <li>• AWT (Abstract Windowing Toolkit): a felhasználói felület komponenseit objektumok reprezentálják</li> <li>• elemi objektumok: címke (Label), gomb (Button)</li> <li>• összetett objektumok: ablak (Window), panel (Panel), görgetőablak (ScrollPane)</li> <li>• a hierarchikus építkezés végett az elemi és az összetett objektumokat egy egységes Component interfészen keresztül látjuk</li> </ul> </li> </ul> |  |
| 27 / 36                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |  |

---

---

---

---

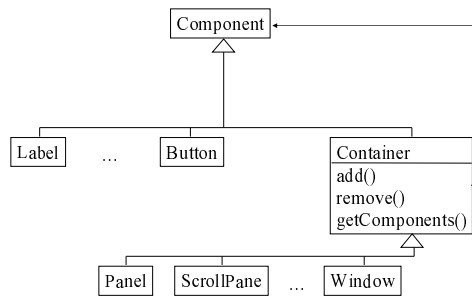
---

---

---

---

## Esettanulmány



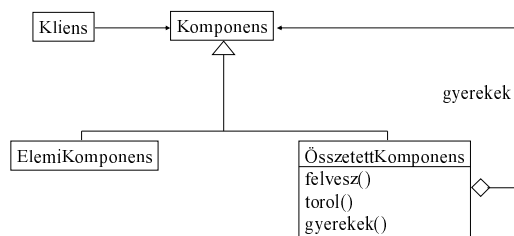
28 / 36

## Alkalmazhatóság

- azonos őstípusú komponensekből álló rész-egész hierarchiák felépítése
- elemi komponensek polimorfizmusát kiterjesztjük az összetett komponensekre
- összetett komponensek rekurzív kompozíciója
- az összetett komponens kliensének nem kell foglalkozni a gyerekek kezelésével

29 / 36

## Felelősségek és együttműködés



30 / 36

## Felelősségek és együttműködés

- **Komponens**
  - definiálja a rész-egész hierarchia elemeinek közös interfészét
- **ElemiKomponens**
  - definiál egy elemi komponenszt reprezentáló osztályt
- **ÖsszetettKomponens**
  - definiálja a gyerekek kezeléséhez szükséges interfészt és annak általános implementációját
  - a műveleteket gyakran rekurzívan továbbterjeszti
- **Kliens**
  - a rész-egész hierarchia elemeit a Komponens interfészen keresztül kezeli

31 / 36

---

---

---

---

---

---

---

## Stratégia

- **Alapötlet**
  - egy algoritmust külön objektumként implementálunk
  - objektumkompozícióval választhatunk a variánsok közül
- **Esettanulmány**
  - AWT komponensek elrendezése sémák szerint történik
  - az elrendezés algoritmus nem a Container leszármazottaihoz van hozzárendelve (redundancia és kombinatorikus robbanás kiküszöbölése)
  - az elrendezési sémák egy LayoutManager ösinterfészen keresztül külön hierarchiával vannak definiálva
  - az elrendezési séma a konténer stratégiája

32 / 36

---

---

---

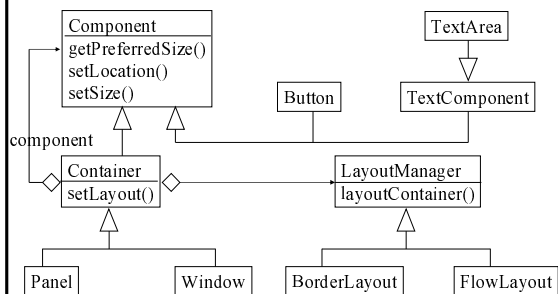
---

---

---

---

## Esettanulmány



33 / 36

---

---

---

---

---

---

---

## Alkalmazhatóság

- Az objektumok viselkedésének egy bizonyos aspektusát függetlenül akarjuk kezelni:
  - nem csak egy osztály, hanem egy osztályhierarchia viselkedését akarjuk variálni (kombinatorikus robbanás elkerülése)
  - egy adott viselkedést leíró kód egységbe zárása
  - az eredeti osztály egyszerűsítése azáltal, hogy a viselkedés egy aspektusa jól elkülönül egy interfészen keresztül

34 / 36

---

---

---

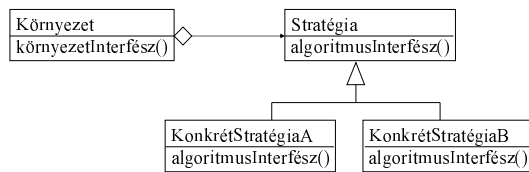
---

---

---

---

## Felelősségek és együttműködés



35 / 36

---

---

---

---

---

---

---

## Felelősségek és együttműködés

- Stratégia
  - definiálja a viselkedési variációk közös interfészét
- KonkrétStratégia
  - megvalósít egy konkrét viselkedési variációt
- Környezet
  - viselkedésének egy részét a KonkrétStratégia objektummal együttműködve határozza meg
  - a kommunikáció a Stratégia interfészen keresztül történik
  - átadja a variáció megvalósításához szükséges adatokat, vagy a saját referenciáját a KonkrétStratégia objektumnak

36 / 36

---

---

---

---

---

---

---