

# OBJEKTUMVEZÉRELT RENDSZEREK TERVEZÉSE

8. gyakorlat

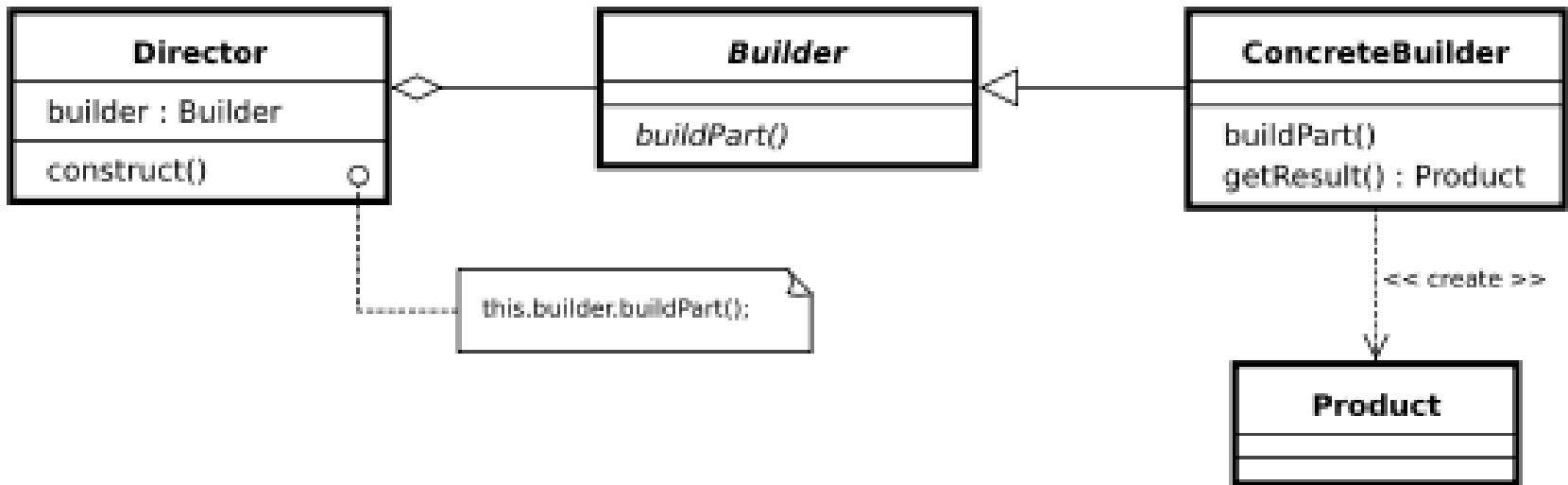
---

# BUILDER

Építő

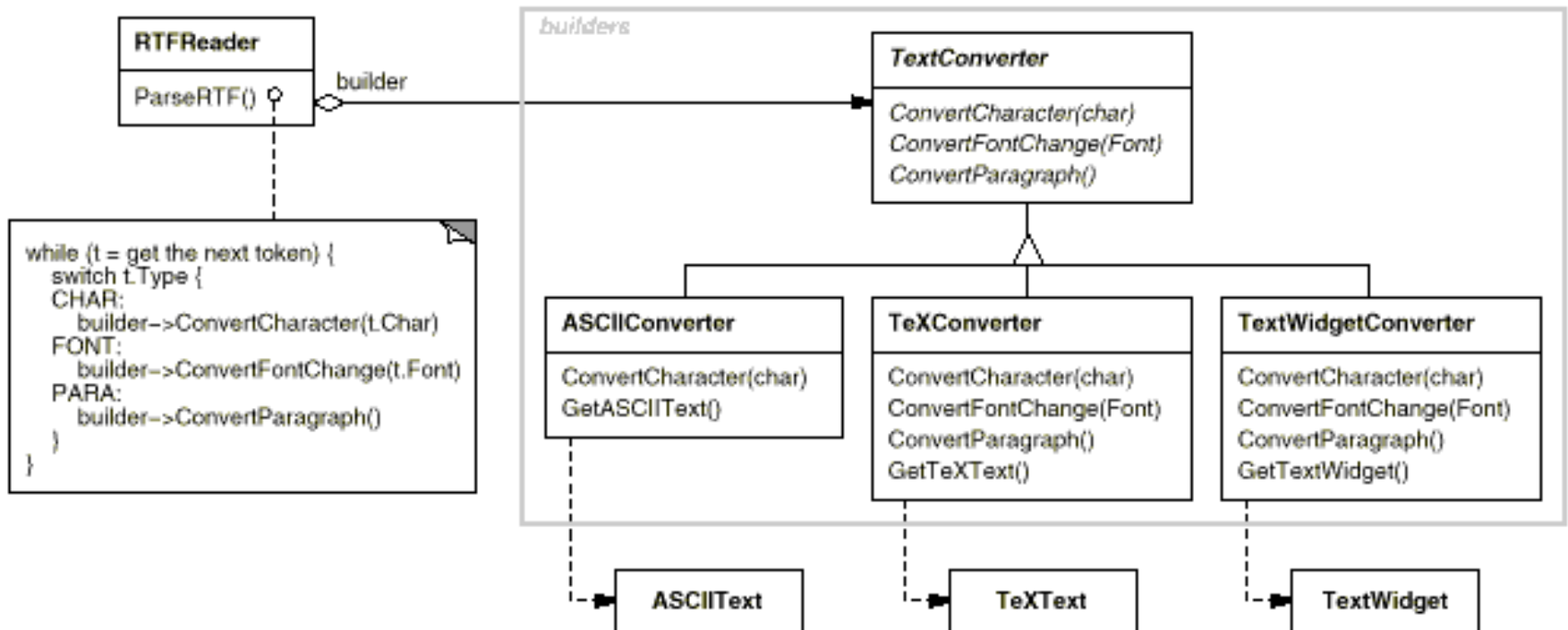


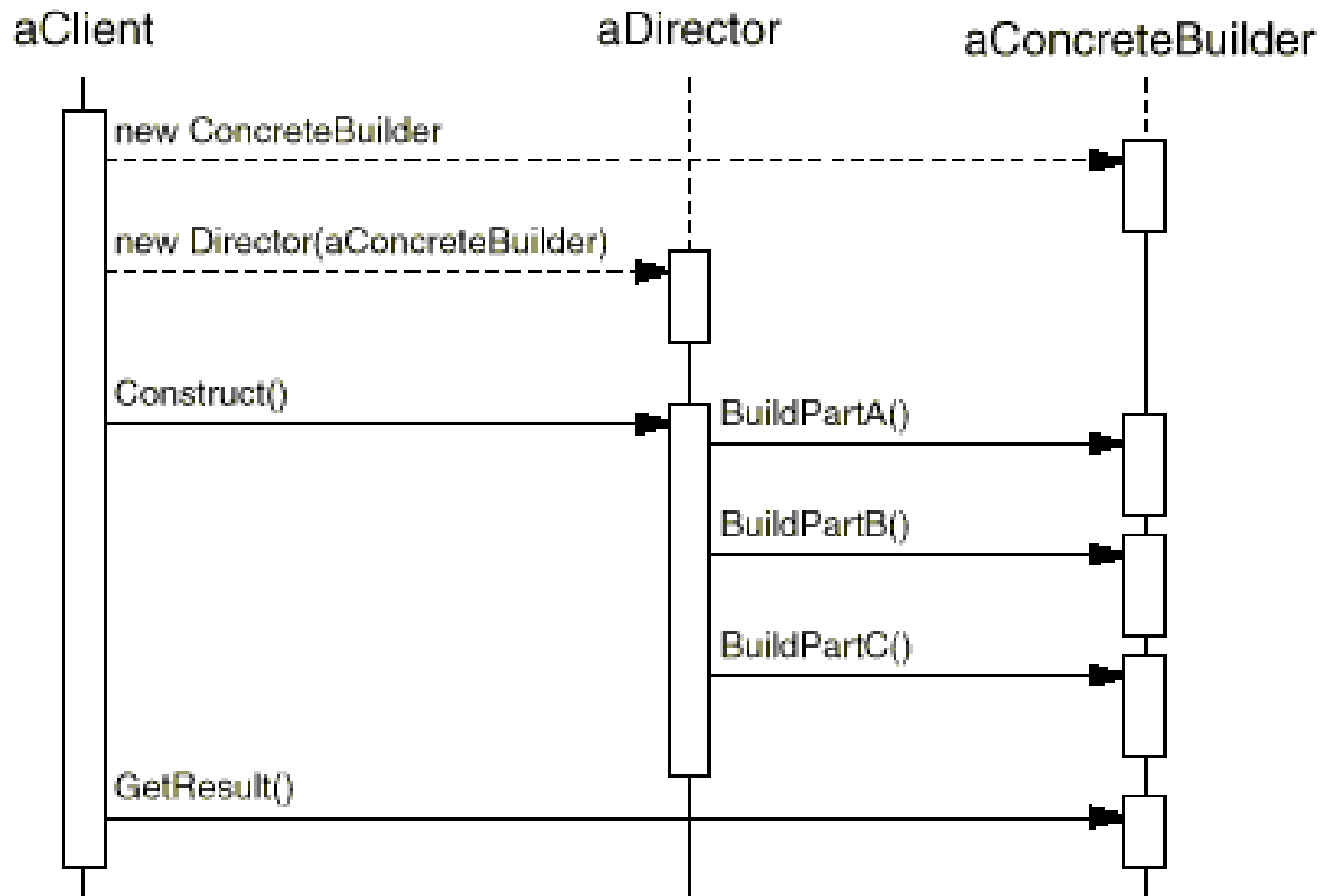
- **Cél:** összetett objektum létrehozási folyamatának elkülönítése a tényleges reprezentációtól
- **Alkalmazhatóság:**
  - összetett objektum gyártási algoritmusra független kell hogy legyen az összetevők részleteitől
  - a gyártási folyamat különböző reprezentációkat kell hogy (egységesen) kezeljen



# Builder

- RTF konverter, ami tetszőleges formátumokba tud exportálni
- Könnyen bővíthetőnek kell lennie új formátumokkal



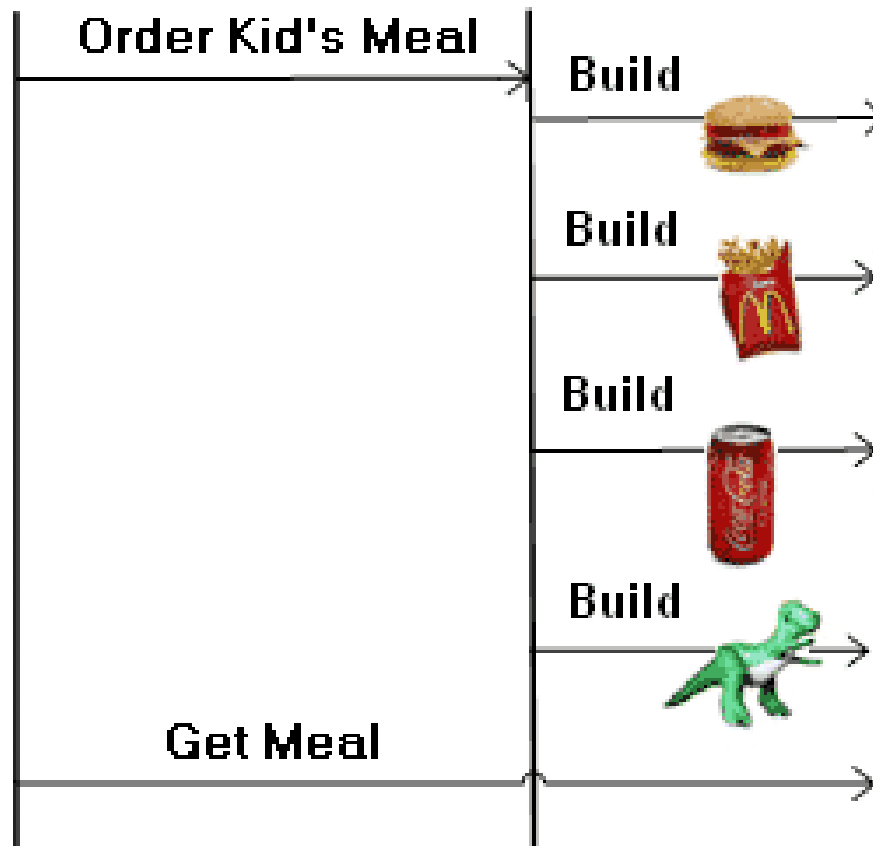


# Builder

**Customer**  
(client)

**Cashier**  
(director)

**Restaurant Crew**  
(builder)





# **CHAIN OF RESPONSIBILITY**

Felelősséglánc

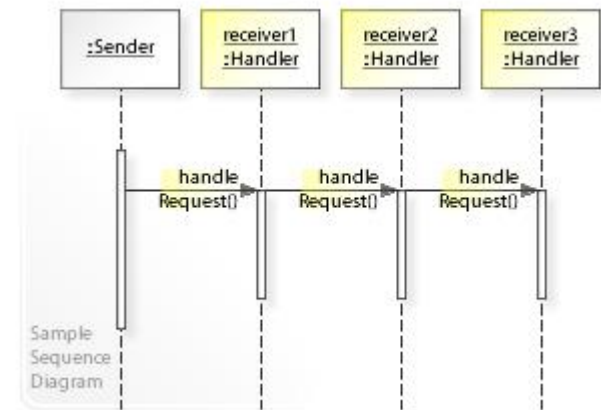
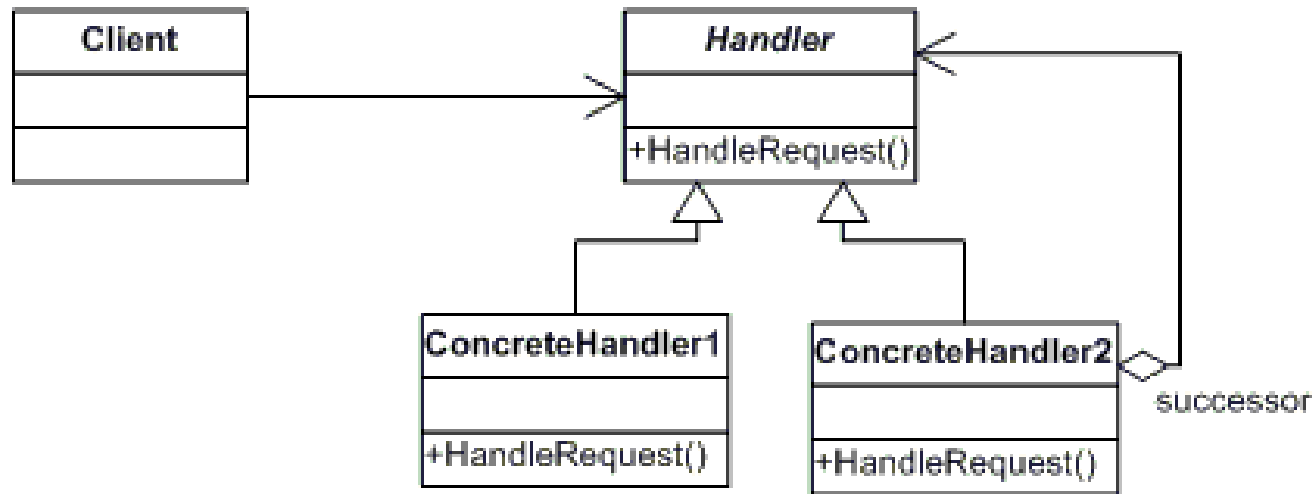
---

- **Cél:** A minta arra szolgál, hogy elkerüljük a kérelem küldőjének és fogadóhoz való kötését. Ezt úgy érjük el, hogy több objektumnak is jogot adunk a kérelem kezelésére. A fogadó objektumokat láncba állítjuk, amelyen a kérelem addig halad, amíg el nem ér egy objektumot, ami képes a kezelésére.
- **Alkalmazhatóság:**
  - Egy kérelmet egynél több objektum kezelhet
  - A kezelő (fogadó) előre nem ismert
  - A kezelők (fogadók) halmazát dinamikusan jelöljük ki

# Chain of responsibility

---





# Chain of responsibility

- Súlyó megjelenítő
- ATM (Példakód)
- Különböző feladatkörhöz/jogosultsághoz tartozó kezelők (üzleti logika)

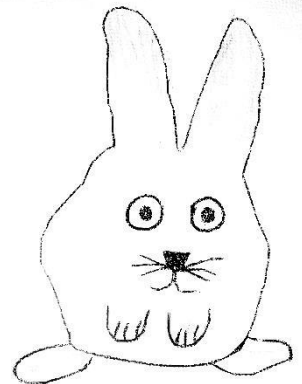
# Chain of responsibility

---

- Tegyük fel, hogy egy osztály Torta objektumokat épít fel pl. tojás, tej, liszt, stb kötelező elemekből és gyümölcs, cseresznye, stb opcionális elemekből. Készítsünk hozzá **Buildert**.

# Feladat

---



- Tegyük fel, hogy egy osztály Torta objektumokat épít fel pl. tojás, tej, liszt, stb kötelező elemekből és gyümölcs, cseresznye, stb opcionális elemekből. Készítsünk hozzá **Buildert**.

# Feladat

---

