OBJEKTUMVEZÉRELT RENDSZEREK TERVEZÉSE

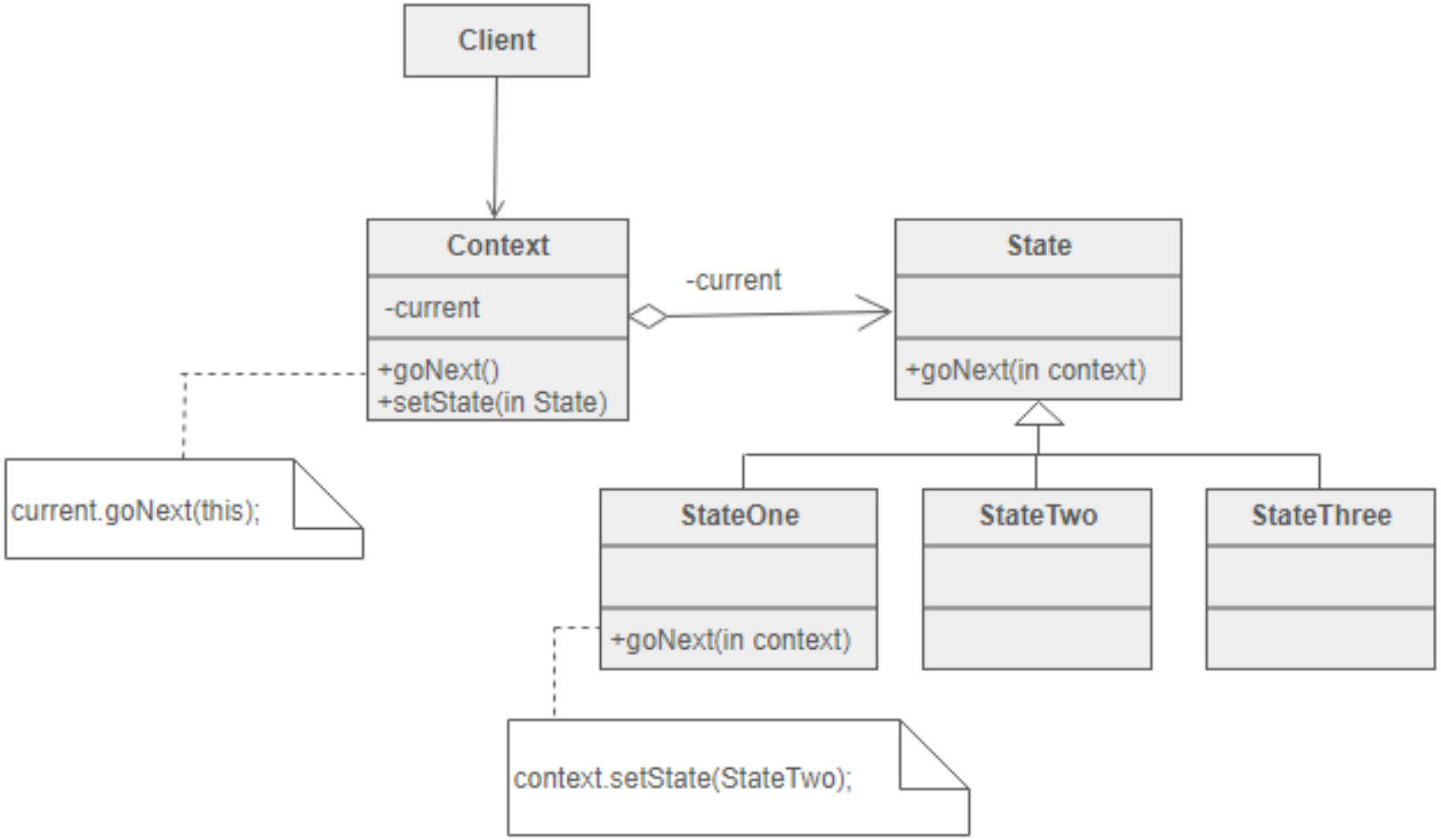
7. gyakorlat

STATE

Állapot

State

- ***Cél***: objektum viselkedésének megváltoztatása belső állapotváltozás hatására (kívülről úgy tűnik, mintha az osztályát változtatná meg)
- ***Alkalmazhatóság***:
  - objektum viselkedése állapotának függvénye, és az futás közben változhat
  - operációkban nagy elágazások vannak, többen ugyanazon szerkezettel, pl. valami konstanstól függővé téve


Client
Context
-current
+goNext()
+setState(in State)
-current
State
+goNext(in context)
current.goNext(this);
StateOne
+goNext(in context)
StateTwo
StateThree
context.setState(StateTwo);

State

- Egy RPG kalandjátékban a karakterünk attól függően, hogy milyen műveletet végzünk vele több állapotban is lehet.
- Az egyes műveletek végrehajtásának az eredménye függ az adott állapottól is.

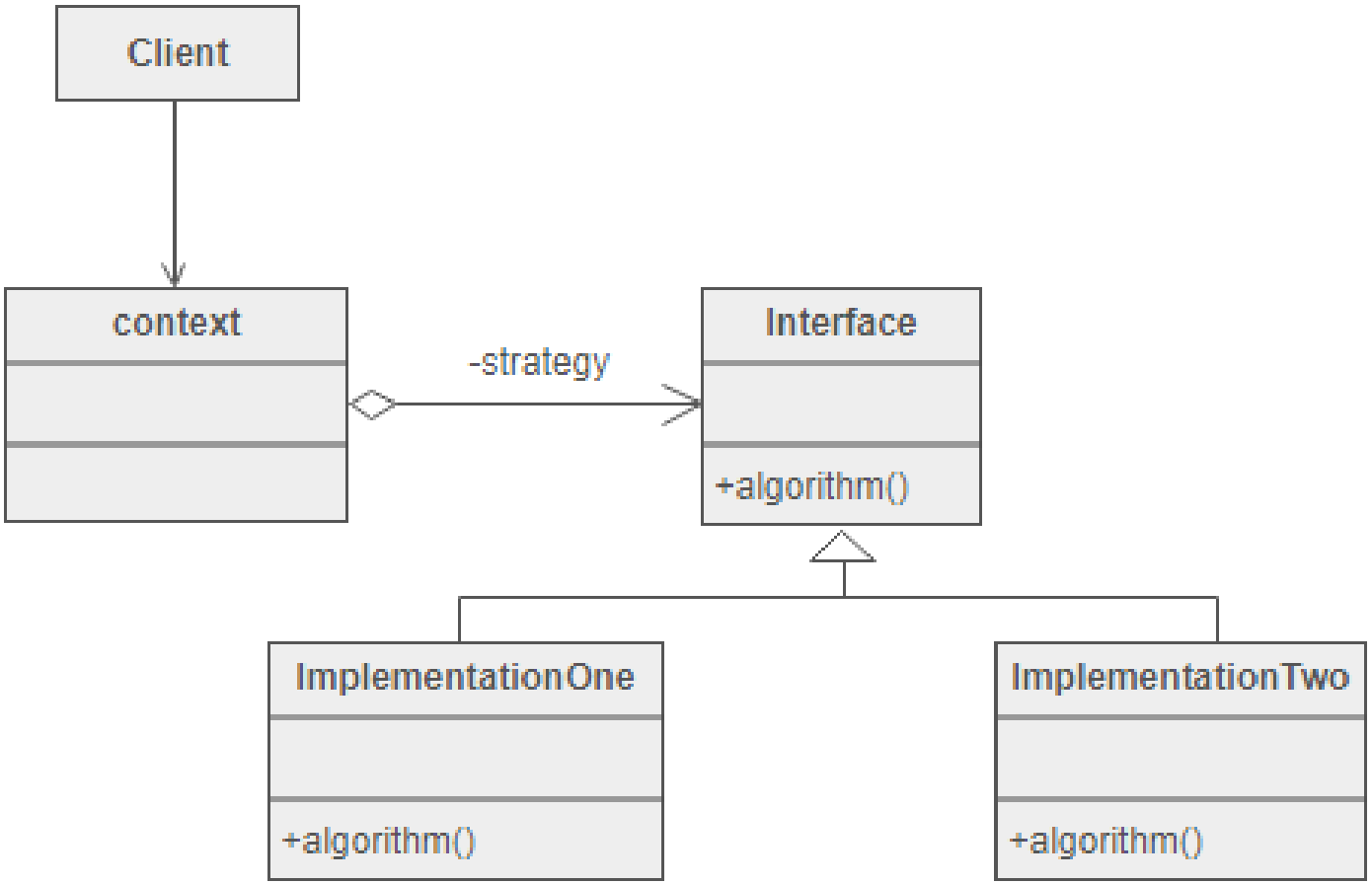
State példa

STRATEGY

Stratégia

Strategy

- ***Cél***: algoritmus-család meghatározása, melyben az egyes algoritmusokat egységbe zárjuk, és egymással felcserélhetővé tesszük.
- ***Alkalmazhatóság***:
  - hasonló osztályok csak viselkedésben különböznek, így a különbözően viselkedő objektumokhoz különböző viselkedést rendelhetünk
  - egy algoritmus több változata: idő/tár optimalizálás,


Client
context
-strategy
Interface
+algorithm()
ImplementationOne
+algorithm()
ImplementationTwo
+algorithm()

Strategy

- Egy alkalmazásban beállítástól függően gombnyomásra menthetünk json-be, vagy xml-be.
- A beállításokat módosíthatjuk bármikor, a gomb megnyomására pedig a beállított módon történik meg a mentés

Strategy példa

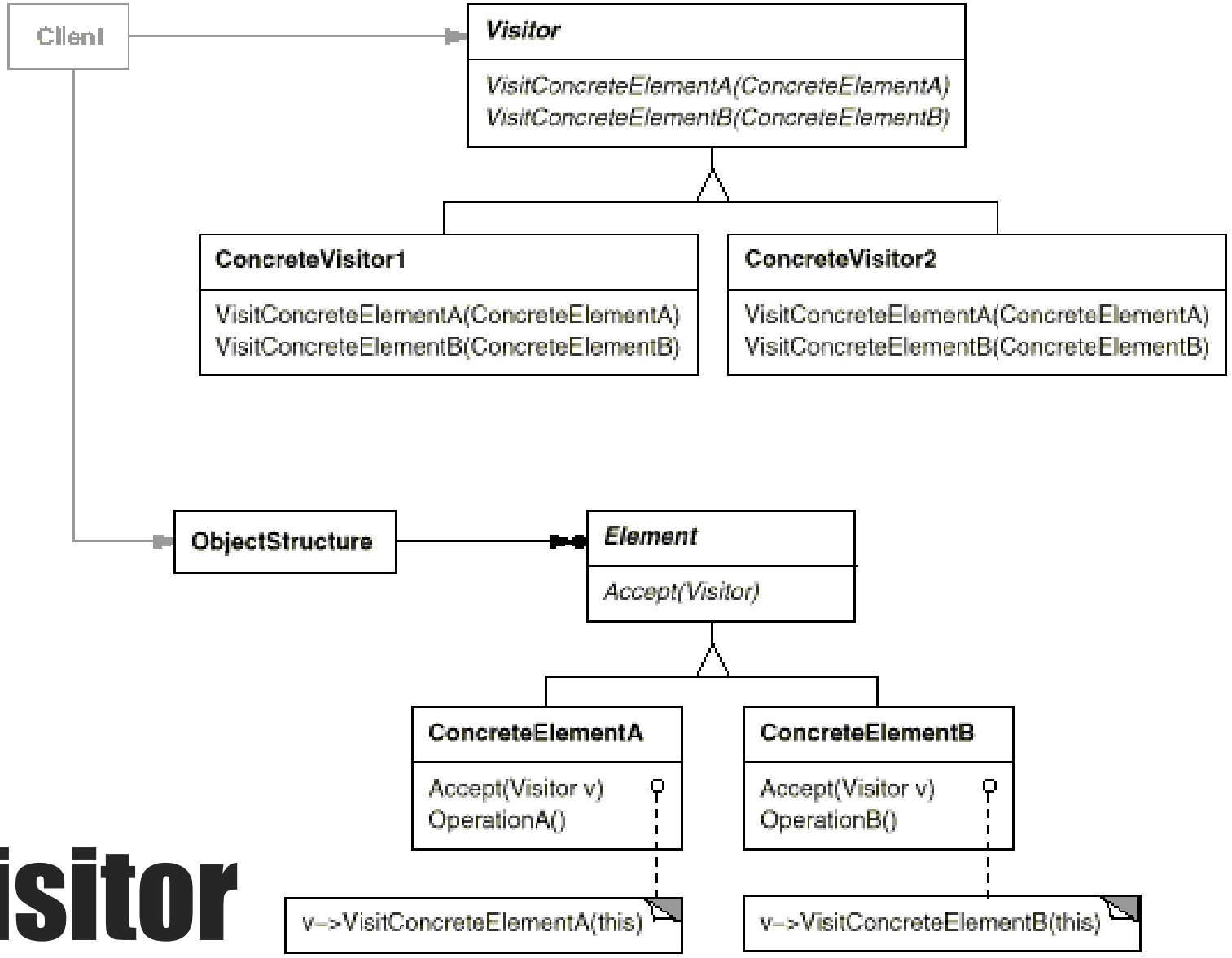
VISITOR

Látogató

Visitor

- ***Cél***: objektum-hierarchián végezhető művelet reprezentálása az osztályok megváltoztatása nélkül
- ***Alkalmazhatóság***:
  - több független műveletet kell elvégezni egy objektum-struktúrán (az osztályok „beszennyezése” nélkül)
  - a struktúrát meghatározó osztályok ritkán változnak, új műveleteket viszont sűrűn definiálhatunk

Visitor


Client
Visitor
VisitConcreteElementA(ConcreteElementA)
VisitConcreteElementB(ConcreteElementB)
ConcreteVisitor1
VisitConcreteElementA(ConcreteElementA)
VisitConcreteElementB(ConcreteElementB)
ConcreteVisitor2
VisitConcreteElementA(ConcreteElementA)
VisitConcreteElementB(ConcreteElementB)
ObjectStructure
Element
Accept(Visitor)
ConcreteElementA
Accept(Visitor v)
OperationA()
ConcreteElementB
Accept(Visitor v)
OperationB()
v->VisitConcreteElementA(this)
v->VisitConcreteElementB(this)

Visitor példa

- Composite mintánál vizsgált könyvtárszerkezet példája
- Van két műveletünk:
  - Fa-szerkezet listázás
  - Könyvtárak és fájlok számának megszámolása
- Két különböző műveletet számoljuk ki két különböző visitorral!

Visitor gondolatok

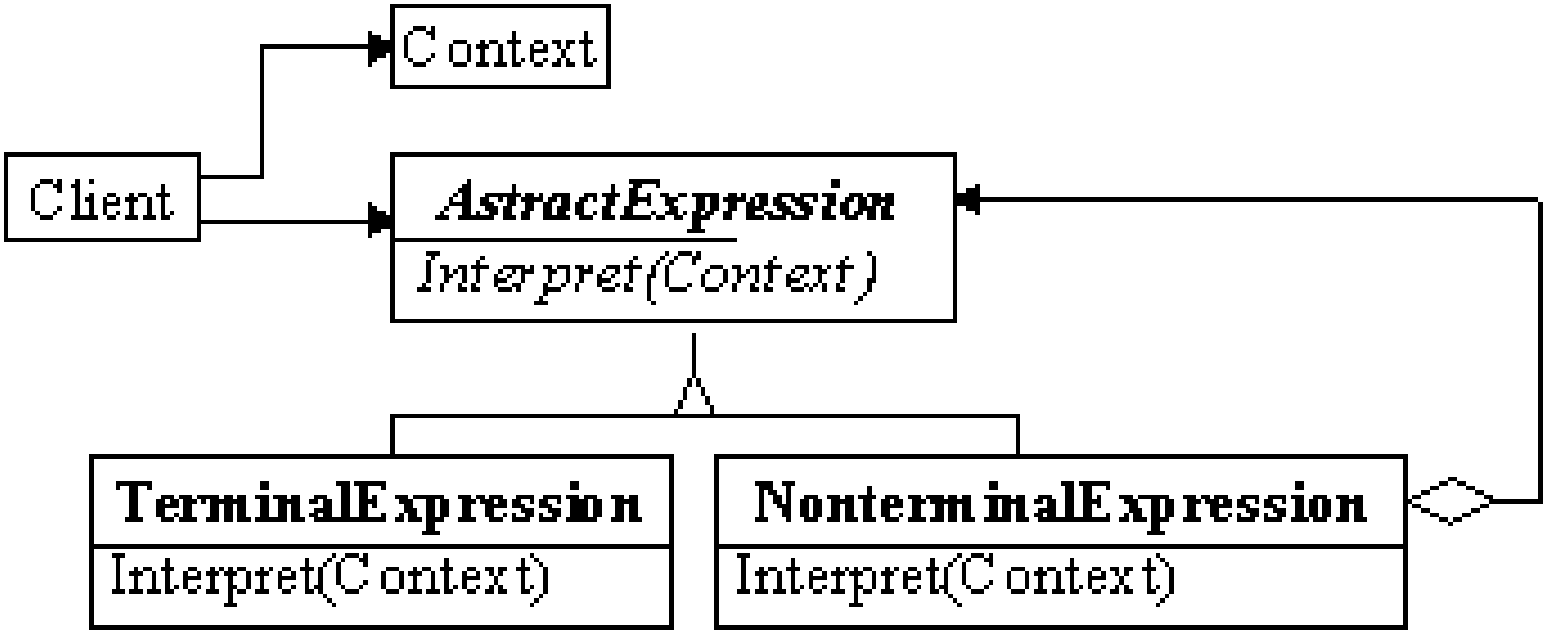
- visit(), visitEnd() → template method?
- generikus visit() metódus, ami általános objektumot vár az UML-en látottak helyett (egy absztrakt osztályban)
  - Java esetében reflection segítségével jó tud lenni

INTERPRETER

Értelmező

Interpreter

- ***Cél***: egy nyelv nyelvtanának reprezentálása a hozzá tartozó interpreterrel (szintaxis fa épül, majd a nyelv szavai ezen lesznek elemezve)
- ***Alkalmazhatóság***:
  - nyelv megadható, melynek szintaxisfáján történhet az elemzés
  - nagyobb feladatokhoz jobbak az elemző-generátorok (osztályhierarchia nagy lehet)
    - ANTLR



Interpreter

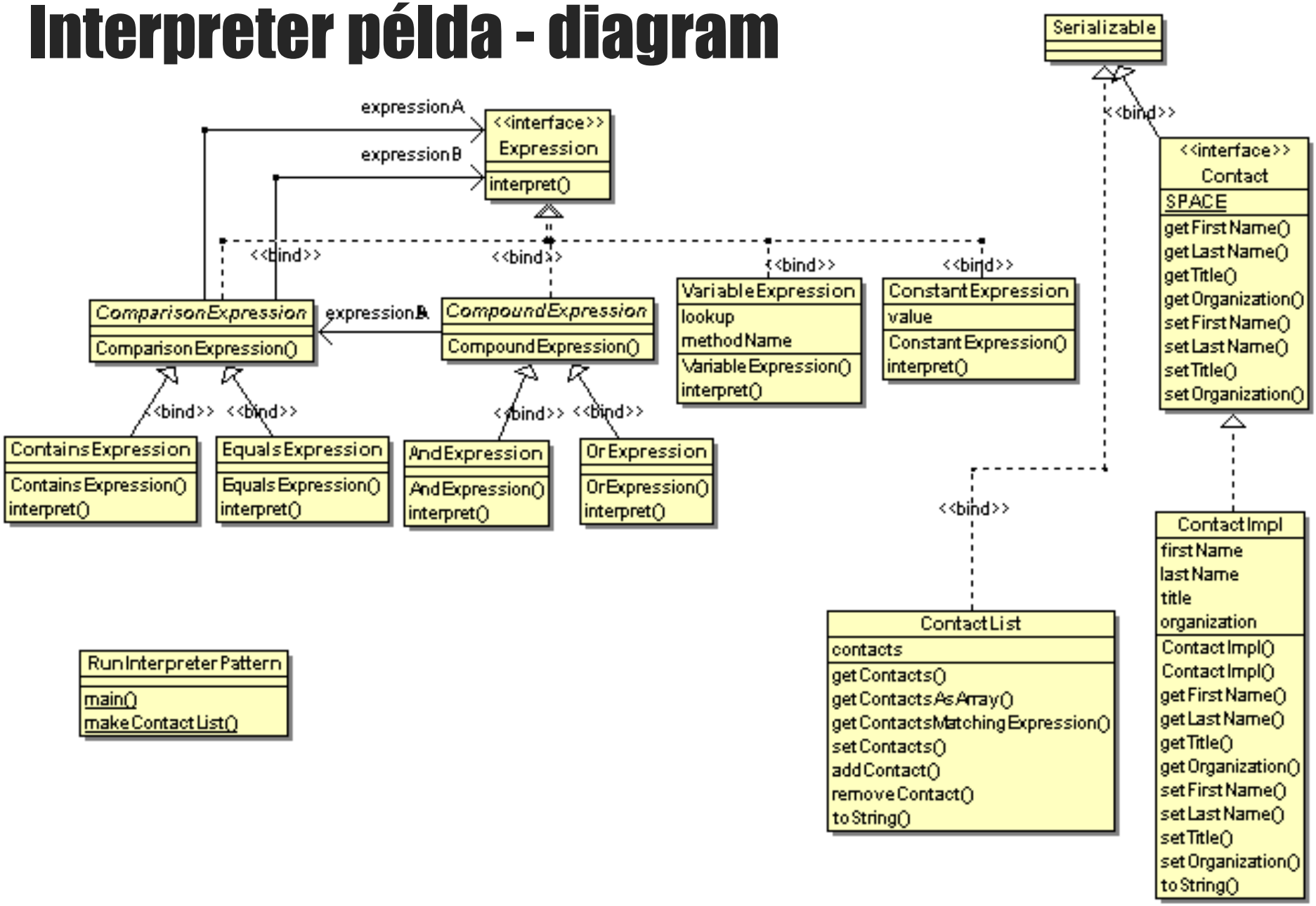
- Van egy személyek elérhetőségét (Contact) tároló kollekciónk (ContactList).
- Interpreter segítségével definiálunk egy kifejezést, amit arra használhatunk, hogy a kollekcióban az adott kritériumoknak megfelelő Contact-okra keresve visszakapjuk a megfelelő Contact-ok egy halmazát

Interpreter példa - leírás

Interpreter példa - résztvevők

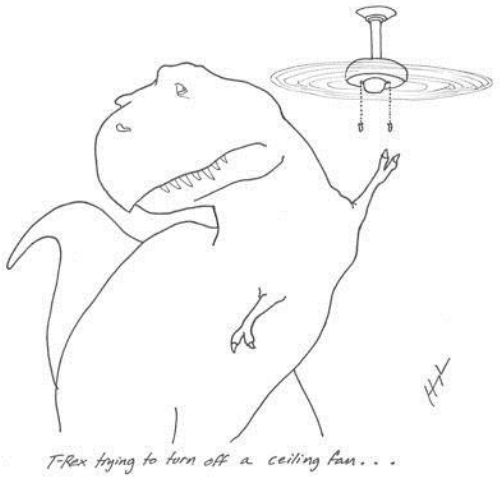
- AbstractExpression – Expression
- NonTerminalExpression
  - ContainsExpression
  - EqualsExpression
  - AndExpression
- TerminalExpression
  - VariableExpression
  - ConstantExpression

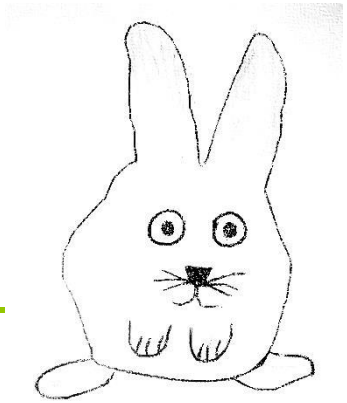
Interpreter példa - diagram



Feladat

- **hu.u_szeged.inf.ovrt.feladat** csomagban található a **CeilingFanExercise.java** fájl. Tanulmányozzuk, alakítsuk át úgy, hogy legyen benne tervezési minta
  - Melyik mintát alkalmazhatjuk?
  - Alakítsuk át, próbáljuk ki.
  - Adjunk hozzá egy új fokozatot is!





Feladat

- **hu.u_szeged.inf.ovrt.feladat** csomagban található a **MusicLibraryExercise.java**. Tanulmányozzuk, alakítsuk át úgy, hogy legyen benne tervezési minta. Jelenlegi gondok?!
  - Minden műfajt egyesével szeretném kikérni (rap, classic, stb). Mindenre kéne egy getter? Egybemosódik az adatszerkezet és az algoritmusok ☹
  - Mi van akkor, ha szeretnék rap és rock zenéket keresni? Egyesével, majd a halmazokat egyesítem? Írjak rá külön egy metódust? Sehogy sem túl jó jelenleg…
  - Melyik minta lehet?