A stratégia minta bemutatása (Strategy)

Készítette: Kardinál Zsófia – bihl3m
Gazdaságinformatikus levelező

A programtervezési mintákkal lehetőségünk van általános, újra felhasználható kódot írni, amit elsősorban objektum orientált környezetben használunk. Összesen 23 féle tervezési minta létezik, amit háromféleképpen lehet csoportosítani. Vannak a létrehozási, a szerkezeti, valamint viselkedési minták. A viselkedési minták (angolul Behavioral-patterns) elsősorban algoritmusokkal, illetve az osztályok és objektumok közötti kommunikációval, a felelősségi körök kijelölésével foglalkoznak.

A továbbiakban a stratégiai mintát fogom bemutatni.

A stratégiai minta vagy más néven policy, egy olyan viselkedési minta, ami lehetővé teszi, hogy több, különböző algoritmust objektumba zárjunk egy közös interfésszel. A stratégiai minta célja, hogy egységbe foglalja az algoritmusokat és ezek az algoritmusok egymással felcserélhetőek legyenek úgy, hogy közben az őt használó klienstől függetlenül változhasson. Az egységbe foglalt algoritmusok hasonló, de nem azonos feladatokat látnak el. Például fájlbeolvasás vagy dokumentum titkosítása is történhet többféleképpen. A minta lehetővé teszi, hogy a későbbiekben a kódunk átfogóbb, bővíthetőbb, elegánsabb, valamint hatékonyabbak legyenek. A stratégiai minta kapcsolatban áll az nyílt-zárt alapelvvel.

A stratégiai mintát használhatjuk, ha:

- több osztály csak a viselkedésükben térnek el egymástól
- egy algoritmus különböző változataira van szükségünk
- olyan adatot használ, amiről a kliensnek nem szabad tudomást szereznie
- sok viselkedést definiál és ezek többféle utasítással vannak jelen

A stratégiai mintának három résztvevője van:

- Strategy
- ConcreteStrategy
- Context.

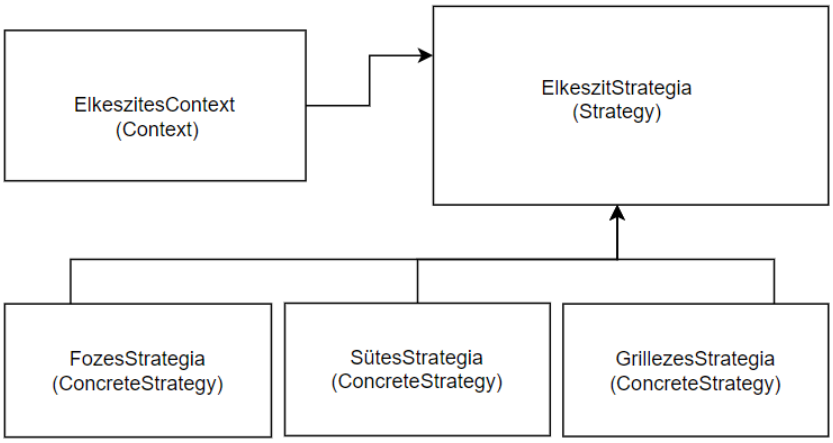
A stratégia a közös interfészt adja meg, a concretestrategy az algoritmust a stratégia interfésszel valósítja meg és a context a kliens egyetlen kapcsolattartási pontja, ami lehetővé teszi a stratégia számára, hogy hozzáférjen az adatokhoz és fenntartsa az objektumra való hivatkozást.

A stratégiai minta a hétköznapokban is jelen van, például játékszoftverek fejlesztésében, adószoftverek készítésében, szupermarketben az egyes leárazásoknál, rendeléseknél, amikor eldöntjük, hogy mivel szeretnénk fizetni, milyen közlekedési eszközzel szeretnénk elmenni valahová vagy akár az alkalmazásokba való bejelentkezésnél, amikor névvel és jelszóval vagy mobiltelefonszámmal és egy egyszer használatos jelszóval lépünk be.

A stratégiai minta bemutatására egy egyszerű programot választottam, amelyben ételek elkészítésének módjai közül lehet választani. Ilyen elkészítési mód lehet például, hogy az adott alapanyagot (mondjuk marhahús) megfőzzük, megsüssük vagy grillezzük, nyársaljuk stb. Az ételeket bármikor elkészíthetjük, de mindig más módon, más eljárással. Az objektum orientált programozásban ezeket az elkészítési lehetőségeket különböző osztályokba vehetjük fel. Majd

a felhasználó, miután kiválasztotta, hogy melyik lehetőséget választja az étel elkészítéséhez, ezt a lehetőséget a Stratégia tervezési mintával implementáljuk.

A mintaprogramomat úgy kezdtem, hogy létrehoztam az *ElkeszitStrategia* absztakt osztályt, amelyben egy *Elkeszit* metódust helyeztem el, amit minden másik osztályban is implementálni kellett. Ezt követően létrehoztam háromféle elkészítési stratégiát, amelyek a *FözesStrategia, SütesStrategia* és *a GrillezesStrategia*. Ezek az osztályok a korábban említett ConcreteStrategy osztályhoz tartoznak a stratégia mintában. Ezután az *ElkeszitesContext* osztályt hoztam létre, ami a nevéből is látszik, hogy a Context osztályhoz kapcsolódik, ami hivatkozást tart fenn az ételre, amit el szeretnénk készíteni és kapcsolatban áll a létrehozott *ElkeszitStrategia* absztrakt osztállyal is. A főprogramban a válaszadónak lehetősége van megadni, hogy mit szeretne főzni és azt milyen módszerrel szeretné elkészíteni.



A minta előnyei:

- az öröklődés segíti az algoritmusok közös funkcionalitását
- egy algoritmus a másik algoritmustól függetlenül változtatható, így könnyebben karbantartható és bővíthető
- különböző megoldási lehetőséget kínál, ugyanarra a problémára
- adatokat változtatni a program futása közben is lehetséges
- bővíthetőségre törekszik

A minta hátrányai:

- szoros belső függőség van az ügyfelek és a stratégiák között
- kommunikációs átfedés miatt van, hogy egyes adatokat, paramétereket nem használnak fel az osztályok
- objektumpéldányok száma megnövekedhet, felhalmozódhat (erre megoldás a flyweight minta, ami ezt kiküszöböli)

A stratégiai minta hatékonysága főként abban mutatkozik meg, hogy az alosztályokra bontás egyszerűsíti és optimalizálja a kódot és annak karbantartását.