Bogdan Roxana, sg 1.1

TEMA 2 PSSC

Domain driven design – factory

Ce este o fabrică?

O fabrică este un obiect care are o singură responsabilitate, aceea de a crea alte obiecte. Fabrica este un obiect care produce agregate complicate sau, uneori, entități și obiecte valoare. Acestaa pot fi folosite în contextul a mai multor scenarii. Sunt adesea utilizate pentru a crea agregate. Agregatele oferă încapsulare și setează o limită consistentă în jurul unui grup de obiecte.

Problemă: Un constructor complex face ceva mai mult decât stabilirea valorilor parametrilor în câmpurile de obiecte.

Soluție: Se crează o metodă de tip fabrică pentru a înlocui apelurile constructorului.

```
public class Employee
{
  public Employee(int type)
  {
    this.type = type;
  }
  //...
}
```

```
public class Employee
{
  public static Employee Create(int type)
  {
    employee = new Employee(type);
    // do some heavy lifting.
    return employee;
  }
  //...
}
```

De ce se folosesc fabricile?

✓ Standardizează și oferă o modalitate de instanțiere a unui obiect astfel încât crearea de obiecte noi nu este pierdută în ambiguitate odată cu cresterea aplicatiei.

✓ Oferă încapsularea cunoștințelor necesare pentru a crea un obiect complex sau un agregat. Fabrica trebuie să știe foarte multe despre structura internă și dependențele obiectului, însă fabrica va proteja această complexitate de lumea exterioară prin furnizarea unei interfețe care să reflecte obiectivele clientului și o vedere abstractă asupra obiectului creat.

✓ Se asigură că obiectele nu trebuie să fie responsabile de creația lor. Obiectele pot fi foarte ușor supraîncărcate cu complexitate atunci când acestea sunt responsabile pentru creația proprie. Obiectele trebuie să aibă o singură responsabilitate. În vederea îndeplinirii acestei cerințe, se va diminua complexitatea unui obiect se până când nu rămâne nimic care nu se referă strict la semnificația sa.

✓ Furnizează un strat important de abstractizare care împiedică clientul de a fi dependent de o anumită clasă conretă. Este de datoria fabricii să instanțieze și să returneze o anumită clasă bazată pe un anumit tip. Acest lucru protejează dependențele din interiorul fabricii.

Există în principal 3 tipuri de fabrici:

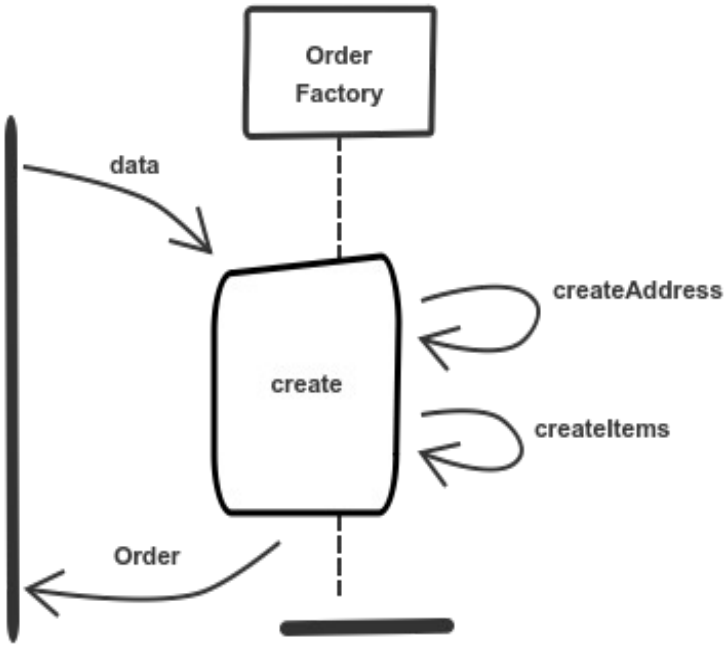
Factory method: o utilizare obișnuită este de a avea o metodă de tip fabrică în interiorul unei rădăcini agregate, astfel încât să își poată construi propriile obiecte.

Abstract factory class: folosită atunci când trebuie să determinăm de ce fel de obiect specific are nevoie clientul. Clasa fabrică încapsulează atât logica deciziei, cât și logica creării.

Builder class: utilizată atunci când există o logică complexă necesară pentru a construi un obiect sau un întreg agregat. Aceasta tip este diferit de factory method prin aceea că obiectul fabricat nu aparține producătorului, iar diferența față de abstract factory este reprezentată de lipsa logicii pentru a decide ce clasă concretă să construiască.

În Domain Driven Design este vorba despre modelarea domeniului unei aplicații. Fabricile reprezintă o parte importantă a domeniului unei aplicații. Domeniul aplicației este responsabil pentru crearea și lucrul cu obiectele de domeniu.

Fabricile nu sunt o neapărat reprezentative când vine vorba de Domain Driven Design, dar sunt necesare la un moment dat pentru a menține obiectele de domeniu clare, curate și pentru a le separa de logică irelevantă.



Beneficii:

- ✓ metoda de tip fabrică nu returnează neapărat un obiect al clasei în care a fost aplelată. Adesea, acestea ar putea fi subclasele sale, selectate pe baza argumentelor date metodei.
- ✓ metoda de tip fabrică poate avea un nume mai bun care descrie ce și cum se returnează și implicit, ceea ce face, comparativ cu un constructor al unei clase.
- ✓ metoda de tip fabrică poate returna un obiect deja creat, spre deosebire de un constructor, care creează întotdeauna o nouă instanță.