**Abstract Factory патерн**

**Дефиниција**

Abstract Factory патерн обезбеђује стварање фамилије повезаних или зависних објеката, не откривајући њихове конкретне класе.

**Појашњење дефинције**

Кроз апстрактну фабрику (AbstractFactory) корисник може позвати методе као што су CreateProductA() I CreateProductB(), а да не мора експлицитно наводити конкретне производе (Product A1, ProductB2…).

**Структура Abstract Factory патерна:**

***A diagram of a product

Description automatically generated***

**Учесници*:***

* **Client** : Користи интерфејсе AbstractFactory и AbstractProdcut, као и њихове изведене класе, за креирање сложеног производа.
* **AbstractFactory**: Декларише интерфејс за операције (CreateProductA(), CreateProductB()) које ће креирати производе.
* **ConcreteFactory**: Имплементира операције (CreateProductA (), CreateProductB ()) интерфејса AbstractFactory, чиме се стварају конкретне инстанце производа (ProductA1, ProductA2, ProductB1, ProductB2).
* **AbstractProdcut** : Декларише интерфејс (AbstractProdcutA, AbstractProdcutB) за производе.
* **ConcreteProduct** : Дефинише конкретне производе (ProductA1, ProductA2, ProductB1, ProductB2) који ће бити креирани путем класа ConcreteFactory. Имплементира операције интерфејса AbstractProduct

**Кориснички захтев *AF1* :**

Менаџер хотелског ланца, одговоран за развој система за резервацију смештаја, послао је захтев тиму развојних инжењера ради креирања базних елемената система:

а. **Екранска форма за унос и приказ резервација:** Израда интерфејса који омогућава корисницима да уносе и прегледају информације о резервацијама. Форма треба да садржи одговарајућа поља за име и презиме госта, шифру резервације, избор хотела, број гостију и датуме почетка и завршетка боравка.

б. **Контролер за размену података са базом:** Контролер треба да обезбеди функционалности за читање, унос, ажурирање и брисање резервација и комуникацију између корисничке форме и брокера базе података.

ц. **Брокер базе података:** Компонента која служи као посредник између система за управљање базом података и апликације за резервацију. Брокер треба да обезбеди функционалности за трајно чување података, измену, брисање и учитавање информација из базе.

Након успешног развоја, менаџер ће бити одговоран за надгледање процеса израде софтверског система и обезбеђивање његове функционалности у складу са захтевима хотелског ланца.

***Веза између елемената структуре решења Abstract Factory патерна и структуре решења примера AF1***

|  |  |
| --- | --- |
| **Елементи Abstract Factory патерна** | **Елементи структуре примера AF1** |
| Client | Menadžer |
| AbstractFactory | Projektant |
| ConcreteFactory | Projektant1, Projektant2, Projektant3, Projektant4 |
| AbstractProductA | EkranskaForma |
| AbstractProductB | BrokerBazePodataka |
| AbstractProductC | Kontroler |
| ProductA | EkranskaForma1, EkranskaForma2 |
| ProductB | BrokerBazePodataka1, BrokerBazePodataka2 |
| ProductC | Kontroler1, Kontroler2 |

**Дијаграм класа примера *AF1*:**

Menadžer

<<interface>>

Projektant

B

Panel1

B

A

Panel1

Panel2

Panel2

SoftverskiSistem

EkranskaForma1

<<abstract>>

EkranskaForma

EkranskaForma2

Panel2

<<abstract>>

Panel

Panel1

BrokerBazePodataka2

<<abstract>>

BrokerBazePodataka

BrokerBazePodataka1

Kontroler2

<<abstract>>

Kontroler

Kontroler1

A

Rezervacija

Projektant2

Projektant3

Projektant4

Projektant1