Programozási technológiák beadandó

A beadandó feladat egy raktárkezelő program tervezése volt, amelynek a megadott funkcionalitásokat és alrendszerek megtervezett interfészeit és „csontvázát” kellett tartalmaznia.

-Több inhomogén raktárrendszerrel való kommunikáció, és közös menedzsment felület

-Áru menedzsment

-Rendelés menedzsment

-Beszállító és vásárló menedzsment

A program nem tartalmaz felhasználói felületet, csak a raktárkezelés csontvázát. Az én raktárkezelő programomban játékkonzolokat tárolunk két híres gyártótól, a Playstation-től meg az Xbox-tól. Létezik egy fő raktár, aminek van két alraktára a gyártótól függően. Ezek az alraktárak kommunikálnak a fő raktárral. A termék a gyártótól függően kerül elhelyezésre egy raktárban. A raktárakból való rendelést is kétféleképpen különböztetjük meg. Adhat le rendelést egy személy és egy bolt is. Természetesen a program ellenőrzi, hogy a személynek, vagy akár a boltnak van-e kerete arra, hogy kifizesse a rendelést. Hogyha nincs elég pénze, akkor a rendelést elutasítja. Külön osztály felel a számla létrehozásáért, amin a vevő adatai, a vásárolt mennyiség és a vásárolt termék szerepel.

A projektben használt tervezési minták a következők:

* Stratégia
* Singleton
* Observer

A számítógép-programozásban a **Stratégia** minta (vezérelv mintaként is ismert) egy szoftvertervezési minta, amely lehetővé teszi, hogy egy algoritmus viselkedését a futás során válasszuk meg. A stratégia minta meghatározza az algoritmusok egy családját, egységbe foglal minden algoritmust, és a családon belül cserélhetővé teszi ezeket az algoritmusokat. A projektben ez a tervezési minta a vevő típusához van használva, mivel vásárolhat személy és bolt is.

A **Singleton**, másnéven Egyke programtervezési minta, egy objektumra korlátozza egy osztály létrehozható példányainak számát. Gyakori, hogy egy osztályt úgy kell megírni, hogy csak egy példány lehet belőle. Ebben a programban a főraktár (MainStock) az, amiből csak egyre van szükség.

Az **Observer**, vagy Megfigyelő minta egy olyan szoftvertervezési minta, melyben egy objektum, melyet alanynak hívunk, listát vezet alárendeltjeiről, akiket megfigyelőknek hívunk és automatikusan értesíti őket bármilyen állapotváltozásról, többnyire valamely metódusuk meghívásán keresztül. A programban ez a játékkonzol beérkezésénél valósul meg, mivel a fő raktárnak vezeti a listát az alraktárakról és a típusnak megfelelően rakja be az adott alraktárba.

A program a következő osztályokat tárolja:

* MainStock (Fő raktár, ami létrehozza az ObserverInfo interfészt.)
* PlaystationStock (Az egyik kisebb raktár, ami a Playstationöket tartalmazza és 2 interfészt (Observer és ProductOperations))
* XboxStock (A másik kisebb raktár, ami az Xboxokat tartalmazza és ugyanúgy az Observer és az ProductOperations interfészt)
* Console (A konzolok adatai)
* Customer (A vevők adatai)
* PersonCustomerStrategy (A személy rendelési stratégiája. Implementálja az CustomerStrategy interfészt.)
* ShopCustomerStrategy (A boltok rendelési stratégiája. Implementálja az CustomerStrategy interfészt.)
* Order (A rendelések)
* Invocie (A számlázás adatai)
* ConsoleSupplier (A beszállítók adatai)
* PayFailed (Ha nem sikerül a rendelés pénz hiányában)

A projekt interfészei:

* CustomerStrategy (Szükséges interfész a stratégia tervezési mintához.)
* ObserverInfo (Szükséges interfész az observer tervezési mintához)
* Observer (Szükséges interfész az observer tervezési mintához)
* ProductOperations (A játékkonzolokkal kapcsolatos műveletekhez szükséges interfész)

A programban egy ENUM különbözteti meg a játékkonzolok fajtáit.

Mivel a projekt nem tartalmaz felhasználói felületet ezért a tesztek bizonyítják a működését, amik a következők:

* InvoiceTest
* OrderTests
* PayFailedTests
* StorageTests