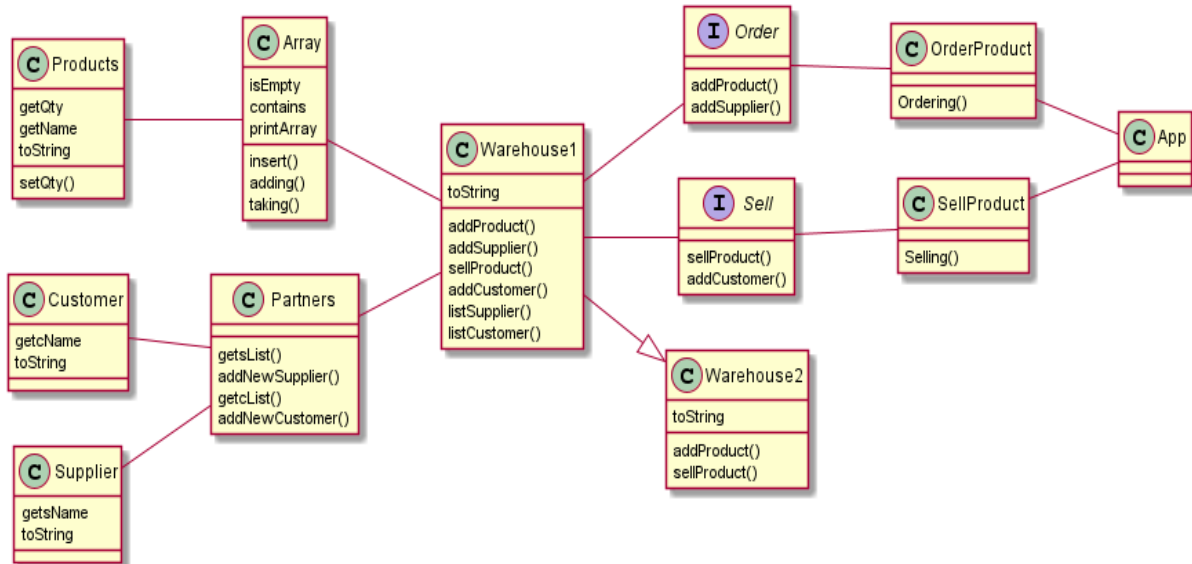


# Designterv raktármenedzsment programhoz

Programozási technológiák gy. (NBT\_PI154G2)

Seres Gergő – HMX501

Eszterházy Károly Egyetem, Gazdaságinformatikus szak, III. évfolyam



1.ábra: a program UML-diagramja

## A program elemei, osztályok szerint csoportosítva, metódusok felsorolásával:

### 1. Products osztály:

Ez az osztály a termékek tárolásának alapjaihoz szükséges, String típusú változóban („name”) tárolja a termék nevét, int típusú egész számként (qty) pedig a rendelkezésre álló darabszámot. Tulajdonképpen egy egyéni típus. A felépítése (private változó és setter) lehetővé teszi, hogy a termékeknek a nevét és a mennyiségét csak a meghatározott metódusok írják felül.

- a getQty: A termék darabszámának gettere, azaz visszaadja az adott termékből, a raktárban található darabok számát.
- getName: A termék nevének gettere, a megnevezés elérésére szolgál.
- toString: A metódus összeállít egy rövid szöveget, mely a termék egyértelmű azonosítására szolgál, az alábbi módon: [termék neve]: [darabszám] db.

### 2. Customer osztály:

A vevők listázásának típusa, ahol a nevek tárolása szöveges alapon történik.

- getName: Hozzáférést biztosít a vevő nevéhez, azaz a típusban tárolt értékhez.
- toString: A terminálra való kiíratást segíti elő.

### 3. Supplier osztály:

A beszállítók listázásának típusa, ahol a nevek tárolása szöveges alapon történik.

- getName: Hozzáférést biztosít a beszállító nevéhez, azaz a típusban tárolt értékhez.
- toString: A terminálra való kiíratást segíti elő.

#### 4. Array osztály:

Egy dinamikus tömb melyben Products típusú elemeket tárolunk, ez egy szándékosan erre a célra létrehozott adatszerkezet, kizárólag a legfontosabb metódusok szerepelnek benne, melyek elengedhetetlenek a termékek menedzseléséhez.

- isEmpty: Ez a függvény egy logikai típusú értékkel tér vissza attól függően, hogy a tömb elemszáma nulla-e, avagy sem.
- contains: Szintén egy boolean típusú értékkel visszatérő függvény, mely azt vizsgálja, hogy az eddig megadott termékek között van-e olyan, amely neve megegyezik a paraméterként megadott termék nevével. Ha van ilyen elem, akkor igaz értékkel tér vissza.
- printArray: Ez a függvény az adatszerkezet megfelelő szöveges megjelenítéséért felel egy szöveges típusú elem visszaadásával. Ebben az szövegben az összes, a tömbben szereplő elem fel van sorolva, úgy ahogy azt a Products osztály toString függvénye adja vissza, vesszőkkel elválasztva.
- insert(): A feltételként megadott elem beszúrása mellett a tömb dinamikus bővítését is végzi egy segédtömb segítségével, melyet akkor hoz létre amikor az adatszerkezet megtelik. Az új elemet mindig a tömb végére szúrja be.
- adding(): Ez az eljárás akkor kerül meghívásra, ha olyan termékből szerzünk be, amelyből már van készleten. Megkeresi az adott terméket név alapján és a mennyiségét megnöveli a rendelt mennyiséggel. Így biztosítva van, hogy ne legyen termék duplikáció.
- taking(): Az adding eljárás ellentettje, azaz a név alapján megkeresett termék mennyiségét annyival csökkenti amennyit eladtunk abból.

#### 5. Partners osztály:

Ez az osztály azért jött létre, hogy a beszállítókat és a vevőket egy helyen kezelhessük. Az osztály javát adó két ArrayList-et bármelyik másik osztályból el lehet érni (csak eljárások útján), így bármennyi raktáregység esetén is fenntartható a stabil partnerkezelés. Maguk a listák privát védelmi szintűek, hogy csak a tervezett módon lehessen hozzájuk férni.

- getList(): Visszaadja a beszállítók listáját vizsgálat és megjelenítés céljából.
- addNewSupplier(): A paraméterként megadott Supplier típusú elemet a listához fűzi az ArrayList metódusainak megfelelően.
- getcList(): Visszaadja a vevők listáját vizsgálat és megjelenítés céljából.
- addNewCustomer(): A paraméterként megadott Customer típusú elemet a listához fűzi az ArrayList metódusainak megfelelően.

#### 6. Warehouse1 osztály:

Mondhatni a program lelke, ebbe az osztályba van bekötve az összes eddig felsorolt objektum. Innen vannak meghívva a program működését adó metódusok. Ebben az osztályban van példányosítva a tároló adatszerkezet. Bizonyos vizsgálatok és visszajelzések is innen indulnak

- toString: A raktár teminálon való szöveges megjelenítését teszi lehetővé. Két lehetséges kimenete van. Üres készlet esetén hibaüzenet kerül a szövegbe, minden más esetben a raktárban szereplő termékeket sorolja fel.
- addProduct(): Megvizsgálja, hogy a beszerzett termékből van-e már raktáron, ha nincs akkor meghívja a beszúrásért felelős eljárást, ha van, akkor pedig egy másik utasítás segítségével kielégíti annak darabszámát.
- addSupplier(): A beérkező termékek feladóját hozzáadja a számukra fenntartott listához, ha még nem szerepelnek.
- sellProduct(): A paraméterként megadott, eladni kívánt termékeket megkeresi az adatszerkezetben, ha szerepel, és van belőle annyi, amennyi a keresett mennyiség, akkor csökkenti a készletet. Egyéb esetben hibaüzenetet ír a terminálra.

- addCustomer(): A vevők adatait szűri be az arra rendszeresített listába, ha nem még nem szerepelnek abban.
- listSupplier(): A program futása során, bármely raktár esetén felmerült beszállítók listájának kiírása a feladata.
- listCustomer(): A program futása során, bármely raktár esetén felmerült vevők listájának kiírása a feladata.

#### 7. Warehouse2 osztály:

A warehouse1 osztályból örököltetett alosztály, mely szülőjének majd összes metódusát átvette. Feladatát és funkcióit tekintve megegyezik az 1-es számú osztállyal. A különbség a reprezentált raktáregység kapacitását és az egyszerre beszerezhető termékek számát érinti. Előbbi maximum 100, míg utóbbi 5 db-ban van meghatározva.

- addProduct(): Lényegében megegyezik a másik osztály azonos nevű eljárásával, de kiegészítésre került a megfelelő darabszámok vizsgálatával, azaz, nem engedi, hogy összesen 100-nál több egység legyen tárolva, illetve, hogy 5-nél többet szerezzünk be egyszerre.
- sellProduct(): 5 darabnál kevesebb, tervezett eladás esetén bonyolítja az eladást, felette hibaüzenetet jelez.

#### 8. Order interfész:

Kapcsolatot tart a rendelési folyamat hívása és a tényleges folyamat között, lehetővé teszi, A paraméterként kapott adatfolyamat két részre bontva, hogy egy hívással elindítsuk a termékbeszerzési és a beszállító regisztrálási folyamatokat. Biztosítja a háttér biztonságos elérését.

- addProduct(): A termékvétel elindítása
- addSupplier(): A beszállítóregisztrálás indítása

#### 9. Sell interfész:

Kapcsolatot tart az eladási folyamat hívása és a tényleges folyamat között, lehetővé teszi, hogy egy hívással elindítsuk a termékbeszerzési és a beszállító regisztrálási folyamatokat. Biztosítja a háttér biztonságos elérését.

- sellProduct(): A termékadás elindítása
- addCustomer(): A vevőregisztrálás indítása

#### 10. OrderProduct osztály:

Ez az osztály azt teszi lehetővé, hogy az interfész, bárholnan meghívható legyen, egy egyszerű parancs segítségével, kevés példányosítás szükséglettel. A paraméterként kapott adatfolyamat két részre bontva biztosítja, hogy az adatokat, csak egyszer, egyszerre, egy helyen kelljen megadni.

- Ordering(): Az Order interfész és metódusainak meghívása

#### 11. SellProduct osztály:

Funkcióját tekintve megegyezik az orderProduct osztállyal, a különbséget az indukált folyamatok jelentik.

- Selling(): A Sell interfész és metódusainak meghívása

#### 12. App osztály:

A program futtatásához szükséges alap eljárás helye, közvetlen vagy közvetett módon ide van bekötve az összes részegység. Itt vannak példányosítva a raktárak.

## **A program működése röviden:**

A vezérlés az App osztályban deklarált metódusok útján történik. A raktárak alaphelyzetben üresek. A termékbeszerzés és az eladás során is három – három adatot kell megadni az adott eljárás hívás paramétereiként: a termék neve, mennyisége, és a partner neve. A program a terminálon keresztül, szövegesen kommunikál, az egyes típusú eredményekhez más-más színt rendelve (megfelelő futtatókörnyezet megléte esetén).

- Zöld színnel jelzi a sikeres termékbeszerzést,
- Piros színnel jelzi a hibaüzeneteket (pl. Üres raktár, nem található termék),
- Fehér színnel az elérhető raktárkészletet,
- Sárgával a sikeres eladást,
- A Kékkel a folyamat során felmerült vevőket és beszállítókat.