Rittgasszer Ákos

Z8WK8D

**Dokumentáció**

Italreceptek

Terv

A program feladata italreceptek eltárolása és kezelés. Több féle italt lehet kezelni (pl. alkoholt tartalmazó és alkoholt nem tartalmazó) Lehet létező alapanyagokból összeállítani új receptet, új anyagokat hozzáadni. Arra is lesz lehetőség, hogy a már létező recepteket eltávolítsuk az adathalmazból.

A tárolt italrecepteket ki lehet menteni fájlba és be is lehet olvasni a receptlistát fájlból. Ki lehet választani egy italt aminek aztán le lehet kérdezni az összetevőit. Egy recept az ital nevéből, összetevőiből és egy megjegyzésből fog állni. A tervezés során igyekszem betartani az objektum orientált programtervezés elveit (SOLID elvek). Ennek megfelelően minden recept egy osztály példánya lesz. Valamint az összetevők is egy osztály lesz. Ezekből lesznek leszármaztatva más osztályok közös tulajdonságaik alapján. Az adatok tárolására heterogén kollekciót fogok használni. Amikor elindul a program akkor beolvassa a hozzávalók és receptek listáját és amikor kilépünk belőle azokat kimenti fájlba.

A programot irányítani a standard inputról lehet parancsokkal és a lekérdezéseket is ide lehet kiíratni (kivéve a fájlba mentést és olvasást). A használathoz szöveges menü lesz amit karakterekkel lehet irányítani. Lesz listázás, hozzáadás, keresés és törlés menüpont. A listázáson belül lehet választani, hogy recepteket vagy összetevőket és, hogy standard outputra vagy fileba. A hozzáadásnál, lehet receptet vagy hozzávalót hozzáadni. A keresésnél lehet receptet keresni ami kilistázza a hozzávalókat. Törlésnél pedig megadott receptet eltávolítani.

Az Array class egy dinamikus tömbhöz szükséges. Ez egy template class. Az adatokat dinamikusan fogja kezelni, viszont ezt a külső használó nem fogja látni. Lesznek olyan függvényei amik a használatához szükséges, például operator[] vagy hozzáadás és eltávolítás.

Az Ingredient osztály példányai lesznek az összetevők. Ezeknek van egy neve és egy mennyisége ami egy mérőszámból és egy mértékegységből áll. Ennek lesz getter illetve setter függvénye amin keresztül lehet állítani.

A Drink osztály lesz alkalmas az egyes italok receptjeinek leírására. Minden Drink típusnak van egy neve, egy listája az összetevőkről (az Array osztály segítségével). Illetve egy leírása amiben tetszőleges dolgot bele lehet írni. Van az osztálynak két leszármazott típusa, egy az alkoholos egy pedig az alkoholmentes italoknak. Az alkoholos italoknak van egy olyan véltozója is amit megmondja hány százalékban tartalmaznak alkoholt.

Az italreceptek tárolására is az Array osztály lesz hasznalva úgy, hogy heterogén adattárolóként működik.

Mindegyik osztálynak meg lesz a kényelmes használathoz szükséges összes függvénye. Lesznek konstruktoraik és mindegyik zárt lesz a használó számára, csupán hasznáni lehet, a belső felépítésük nem fog látszani.

**Megvalósítás**

A megvalósításhoz létrehoztam egy Array nevű template osztályt, ez az array.h header fileban van specifikálva. Az osztály dinamikus tömbben tárol adatokat, úgy, hogy egy konstruktorban megadható értékként lefoglalódok a megadott hely és egy lokális változó mutatja, hogy mennyi hasznos adat van éppen a tömbben. Ha elfogy a hely akkor lefoglal egy kétszer akkora területet és bemásolja az eredeti adatokat. Ha hozzá akarok adni elemet akkor az amíg nincs tele a tároló gyorsan elvégezhető, ha tele van akkor már számítás igényesebb a művelet viszont erre az adatok számának növekedésével egyre ritkábban kerül sor. Léteznek gettter függvényei az osztálynak, illetve lehet elemet eltávolítani belőle. Valamint a használathoz szükséges operátorok is megtalálhatók (pl.: <<, =).

Stringek tárolásához a String osztályt használom (string5.h, string5.cpp). Ez az osztály az alapvető string műveletek elvégzéséhez szükséges. A stringet egy char\* ban tárolja a lezáró nulla nélküli hosszal együtt. A beolvasás nem spaceig olvassa be a stringet hanem a iscntrl() függvény ellenőrzi a végét. Ez azért hasznos, mert így tabulátorral választom el a különböző adatokat, vagyis egyszerűen tudok szóközökkel teli stringet beolvasni (például egy nevet).

Alapvetően két fontos típust használok. A hozzávalókat és az italokat. Az italokat Drink típusként tárolom el, míg a hozzávalókat Ingredientként. Ezeket egy-egy Array<> típusú tárolóban tárolom.

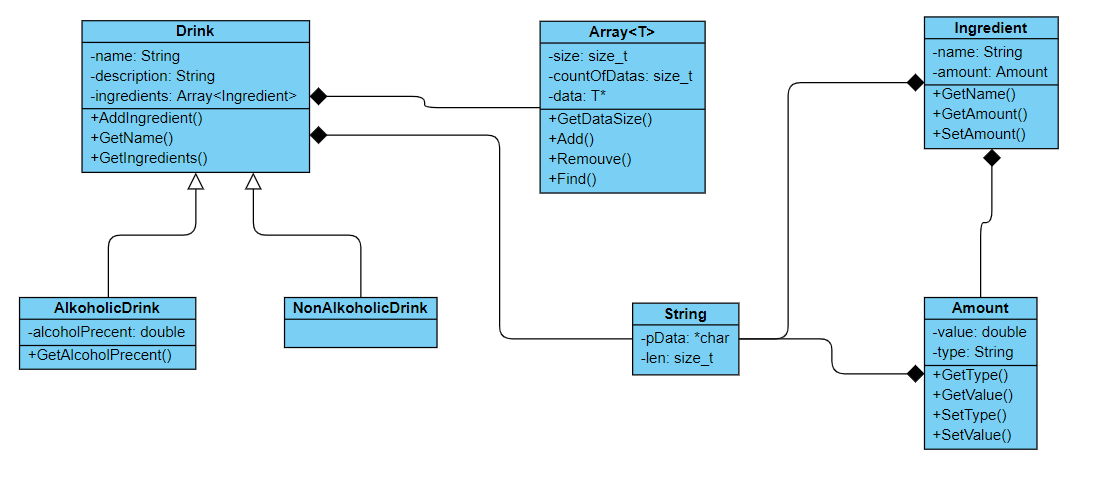
Az Ingredient osztály a ingredient.h állományban van definiálva. Ehhez használok egy másik típust Amount néven ami a mennyiséget tárolja egy double mennyiség és egy String mértékegység változó segítségével. A mennyiség csak azokban az esetekben fontos amikor egy italhoz rendelem hozzá a hozzávalót, amikor csak a hozzávalók listájában van akkor ennek értéke 0.0 és “”. Az Ingredient osztálynak van egy amount (Amount típusú) és egy name (String típusú) privát adattagja. Ezen kívül a szokásos operátorokat és getter valamint setter függvényeket tartalmaz.

Az egyes receptek tárolásához a Drink osztályt használom, ami a drink.h hederben van definiálva. Minden egyes ital példánynak van egy neve (String name), egy leírása (String description) és egy listája a hozzávalókkal (Array<Ingredient> ingredients) privát adattagja. Ebből leszármazik két osztály, az AlkoholicDrink és a NonAlkoholicDrink. Az alkoholos italoknak van egy plusz adattagja ami az alkoholszázalékot tárolja. Az osztályoknak továbbá megvannak a szokásos operátoraik, konstruktoraik és getter, setter függvényeik. Ezeken felül egy függvény amivel lehet hozzáadni hozzávalókat (AddIngredient).

A main függvényben egy karakteres menü vezérlése van, ez karakterekkel lehetséges, ahol egy switch hívja meg a megfelelő függvényt. Létrehozok a legelején egy-egy tárolót az italoknak illetve a hozzávalóknak. Ezután beolvasom az előzőleg elmentett adatokat. Ezt követően a felhasználó szerkesztheti az italokat és hozzávalókat. Amikor végzett elmentem egy fileba az egész adatszerkezetet.

A felhasználó a Hozzávaló Hozzáadás: h, Recept Hozzáadás: r, Eltávolítás: e, Listázás: l, Kilépés: k parancsokkal tud navigálni. Hozzávaló hozzáadásánal egy nevet kell megadni. Recept hozzáadásánál egy nevet, egy leírást és a hozzávalók indexét, mennyiségét és mértékegységét kell megadni. Eltávolítani italt egy indexszel lehet. Listázni pedig kilistázza először a hozzávalókat és a recepteket.

A memóriaszivárgás kiküszöböléséhez includáltam a memtrace.h header filet. Egy test.h csatolásával pedig az alapvető műveletek ellenőriződnek le.

A programban minden osztályban az adattagok privátok és getter illetve setter függvényekkel lehet kezelni őket. Ezen felül az osztályok kívülről nem mutatják a működésüket és tetszőlegesen bővíthetőek. Lehetőség szerint figyeltem a konstans adatok és függvények használatára, hogy a kód jól olvasható legyen. Az olvashatóságot a beszédes fájl, változó és függvénynevek segítik. A leszármazáskor létrejött osztályok használhatóak az ősosztály helyett, ezt mutatja az is, hogy heterogén tárolóban vannak elhelyezve. Ezen felül minden osztály egyetlen dologért felel.

A fenti osztálydiagramm ábrázolja az osztályokat, azok összefüggéseit és fontosabb függvényeit.