

**Objektum orientált programozás**  
**7. gyakorlat**  
**Öröklődés, felüldefiniálás, referenciák típusai**

1. Definiáljon saját névtérben (*myclasses*) **Áru** (*Product*) osztályt.

Adattagjai: név, nettó ár (egész), áfakulcs (egész, százalék)

Konstruktor: mindhárom adat megadásával inicializálja az adattagokat.

Metódusai:

- Számítsa ki a bruttóárát egy árunak (egész, kerekítve).
- Egy sztringbe összefűzve adja vissza az áru nevét és a bruttó árát. Definiálja felül az Object osztálybeli ToString metódust!
- Növelje meg az áru nettó árát a paraméterben megadott százalékos értékkel (egész).
- Hasonlítsa össze két áru bruttó árát. 1-et adjon vissza, ha az áru drágább mint a paraméterben megadott; -1-et ha olcsóbb és 0-t ha azonos árúak.

Készítsen a *myclasses* alnévtérben (*myclasses.products*) Kenyér osztályt, amely az Áru leszármazottja.

Adattagja: mennyiség (valós, pl. 0.75)

Konstruktor: négy paraméterben kapott adattal inicializálja az objektumot.

Metódusai:

- Definiálja felül a ToString metódust. Az ősoosztályban definiált metódushoz képest fűzze hozzá az egységárat (bruttóár/mennyiség).
- Adja vissza a mennyiséget.
- Hasonlítsa össze két kenyér egységárát. Az osztályszintű metódus akkor adjon vissza igazat, ha a paraméterként kapott két Kenyér közül az elsőnek nagyobb az egységára.

a) Készítsen új névtérben futtatható osztályt. Ebben hozzon létre egy Áru és egy Kenyér objektumot. Írja ki az adataikat, és hasonlítsa össze az árukat. Melyik osztály metódusával lehetséges az összehasonlítás?

b) Hozzon létre még egy Kenyér típusú objektumot, de ennek a referenciáját egy Áru típusú változóban tárolja (neve: product2). Próbálja ki, hogy a "product2" referenciára meghívva milyen eredményt ad a ToString metódus. Meg tudja-e hívni ezen a referencián keresztül a Kenyér mennyiségét visszaadó metódust?

c) Hozzon létre még egy Kenyér példányt (neve: bread2) és a "product2" referenciával hivatkozott másik Kenyérrel összehasonlítva a nagyobb egységárút kell kiírni (konvertálásra lesz szükség).

2. Készítsen saját névtérben (*myclasses*) **Személy** osztályt.

Adattagjai: név, kor

Konstruktor: két paraméterben kapott értékkel inicializálja az adattagokat

Metódusai:

- Getter tulajdonságok az adattagok lekérdezésére.
- A Személy adatait sztringben összefűző és ezzel visszatérő metódus.
- Adjon vissza igazat, ha a paraméterben megadott személy életkora kisebb.

Készítsen a *myclasses* alnévtérben (*myclasses.persons*) **Gyermek** és **Felnőtt** osztályokat a Személy osztály leszármaztatásával.

Adattagjaik: iskola (a Gyermek osztályban, string) ; munkahely (a Felnőtt osztályban, string)

Konstruktoraik: három paraméterben kapott értékkel inicializálják az adattagokat

Metódusaik:

- Mindkét osztályban definiálja felül az adatokat sztringben összefűző öröklött metódust.
- Adja vissza a Gyermek iskoláját, ill. a Felnőtt munkahelyét.

Készítsen futtatható osztályt új névtérben. Ebben deklaráljon két személy típusú referenciát, amelyek értéke a személy beolvasó függvény által visszaadott referencia lesz.

Írjon személy beolvasó függvényt, amelyben beolvassa egy személy adatait (név, kor). A referencia dinamikus típusát ellenőrzött módon határozza meg (ha 18 év alatti akkor Gyermeke, egyébként Felnőtt), majd értelemszerűen olvassa be a Gyermeke iskoláját ill. a Felnőtt munkahelyét. Hozza létre az objektumot, majd térjen vissza a referenciájával.

A main metódusban kérdezze le a létrejött objektumok típusát (használja az *is*, *as*, *typeof* operátorokat és a *GetType* metódust) és írassa ki az adataikat.

### Házi feladat:

#### 1. A 2. feladat folytatása

A Felnőtt osztály kiterjesztéseként hozzon létre ugyanabban a névtérben (*myclasses.persons*) egy **Alkalmazott** osztályt.

Adattagja: fizetés (egész), nyugdíjkorhatár (osztályszintű, egész: értéke 65)

Konstruktor: négy paraméterben kapott értékkel inicializálja az adattagokat (név, életkor, munkahely, fizetés).

Metódusai:

- Adja vissza a fizetést.
- Definiálja felül az adatokat sztringben összefűző öröklött metódust.

Készítsen új futtatható osztályt (a 2. feladatban létrehozott futtatható osztály alapján). Amikor meghatározza a Személy típusúra deklarált referenciák dinamikus típusát, a Felnőttek esetén ellenőrizze a munkahely értékét: csak akkor lehet Alkalmazott, ha a munkahely nem üres (nem minden felnőtt alkalmazott). Ha alkalmazotról van szó, akkor a fizetését is olvassa be. Ezek után kérdezze le a létrejött objektumok típusát és írassa ki az adataikat.

Annak vizsgálata, hogy a munkahely értéke (string) üres-e:

```
string.IsNullOrEmpty(workplace)
workplace.Length == 0
workplace == string.Empty
workplace.Equals("")
```

Módosítsa úgy a futtatható osztályban a Main metódus kódját, hogy hozzon létre egy *n* elemű Személy tömböt. *N* értékét olvassa be ellenőrzött módon. Az adatok beolvasásakor dőljön el, hogy az adott személy Gyermeke, Felnőtt, vagy Alkalmazott. Majd életkoruk szerint növekvően rendezve írja ki az adataikat. Használja a minimum kiválasztásos rendező eljárást (lásd 6. gyakorlat).

#### 2. Github classroomban

A Könyv osztály leszármazottjaként készítsen egy másik névtérben KönyvStílus osztályt.

Adattagjai: stílus (szöveg)

Konstruktor: minden adatát paraméterben kapott adatokkal inicializálja.

Metódusai:

- Adja vissza a stílust.
- Definiálja felül az ősosztálybeli sztringet visszaadó metódust úgy, hogy az ősbeli sztringhez fűzze hozzá a stílust is.

Készítsen futtatható osztályt. Ebben olvasson be *n* darab stílusos könyvet egy tömbbe. *N*-et ellenőrzött módon olvassa be (1 és 10 közötti érték).

a) Számolja meg hányféle különböző stílusú könyv szerepel a tömbben.

b) Írja ki a "SciFi" stílusú könyvek adatait.

c) Számítsa ki a "SciFi" stílusú könyvek átlagárát.

A sztringek egyezésének vizsgálatára használja a string osztály metódusát:  
`bool Equals(string value, StringComparison comparisonType)`