**Dictionary:**

Előfordulhat olyan eset, amikor egy karakterláncot meg kell feleltetnünk egy másik karakterláncnak, ilyenkor tipikusan a szótár adatszerkezetet kell használnunk. Ezek a szótárak felépítése felfogható két egymással kapcsolatban álló listaként, az egyik listában vannak a kulcsok a másik listában, pedig a kulcshoz tartozó megfelelő érték, ezért szótárak esetén két db típus kell megadnunk a <> jelek között, vesszővel elválasztva. Az első a kulcs típusa lesz a második a kulcshoz tartozó érték típusa, természetesen nem muszáj két db string kulcs-értékpárt adni, lehet akár int és string kulcs-értékpár is. Deklarációkor is megadhatunk már adatokat így:

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

Deklaráció során a két adattípus közül az első adattípus a kulcs lesz, míg a második adattípus a hozzátartozó érték.

Dictionary<adattípus, adattípus> szótár = new Dictionary<adattípus, adattípus>();

Természetesen később is adhatunk hozzá kulcs-értékpárt, az Add metódussal: szótár.Add("zöld", "green"); Mint már említettem a szótárban kulcs-értékpáros (KeyValuePair) tárolódik el, így egy foreach ciklusnál ezt kell megadnunk, valamint a kulcs és az érték típusát, ezután a KeyValuePair típusú változóban lesznek eltárolva ezek az adatok. A kulcsot a .Key metódussal tudjuk elérni, míg az értéket a .Value metódussal. A szótár végigjárása:

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

**Főbb metódusok:**

* **ContainsKey(keresett kulcs)/ ContainsValue(keresett érték)**
  + Tartalmazás vizsgálat, ha az adott kulcs(Key) vagy érték(Value) szerepel a szótárban true-t, ha nem false-t ad vissza, a ContainsKey csak a kulcsok között, a ContainsValue csak az értékek között keres.



* **Count()** – mint a listánál
* **Remove()** – Törli a kulcsot és a hozzátartozó értéket a kulcs alapján. Ha sikerül akkor igaz, ha nem sikerül akkor hamis értéket ad vissza.
* új elem berakása: **Add(kulcs, érték),** ha a kulcs már létezik, akkor ArgumentException-t kapunk
* létezik-e a kulcs?: **szótáram.ContainsKey(kulcs),** igaz/hamis értéket ad vissza
* létezik-e az érték?: **szótáram.ContainsValue(érték),** igaz/hamis értéket ad vissza
* kulcsok halmaza: **szótáram.Keys,** egy a kulcs típusának megfelelő kollekcióban eltárolhatjuk a kulcsok értékeit, lekérdezéskor ToList metódussal listává konvertálható az adathalmaz **(szótáram.Keys.ToList())**
* értékek kollekciója (nem feltétlen halmaz) **szótáram.Values,** egy az érték típusnak megfelelő kollekcióban eltárolhatjuk a szótárban szereplő értékeket, lekérdezéskor ToList metódussal listává konvertálható az adathalmaz **(szótáram.Values.ToList())**
* elem eltávolítása kulcs alapján: **szótáram.Remove(kulcs),** a kulcsot és a hozzá tartozó értéket is eltávolítja a szótárból, ha sikerül, a metódus igaz eredményt ad, különben hamisat
* kivételkezelt értéklekérés: **szótáram.TryGetValue(kulcs, out érték),** igaz/hamis eredményt ad, ha a kulcs létezik, akkor a visszatérési érték igaz, az érték paraméterben eltárolja a hozzá tartozó értéket is
* elemszám: **szótáram.Count**
* összes elem törlése: **szótáram.Clear**

**Fontos:** Egy elem kiválasztása nem az index, hanem a kulcs alapján történik.

**Példa:**

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

Létrehozunk egy szótárt, amely egy string és egy int tárolására alkalmas. A string a kulcs, amely a név tárolására lesz alkalmas. Az int egy darabszámot fog tárolni, amely azt határozza meg, hogy melyik névből hány darabot vettünk fel.

Az elágazás segítségével meghatározzuk, hogy a megadott név szerepel-e a szótárban. Amennyiben szerepel, úgy növeljük a darabszámot, amennyiben nem szerepel hozzáadjuk a nevet a szótárhoz és megadjuk a kezdő darabszámot.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Foreach segítségével kiírjuk a konzolra a szótár tartalmát. Kiírjuk a kulcsot, amelyet a Key tárol, majd kiírjuk a hozzátartozó értéket, amelyet a Value tárol.

A képen képernyőkép, szöveg, Betűtípus látható

Automatikusan generált leírás

Meghatározzuk, hogy a három, mint értékel szerepel-e benne.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Összeszedjük a kulcsokat egy listába.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Összeszedjük az értékeket egy listába.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Meghatározzuk, hogy hány darab kulcs van a listában.

A képen képernyőkép látható

Automatikusan generált leírásLekérjük a kulcshoz tartozó értéket, amennyiben a kulcs létezik. Innentől a db változó tárolja az értéket, hogyha sikeresen megtaláltuk a kulcshoz tartozó értéket.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Jelen esetben töröljük a szótár teljes tartalmát.

**Feladatok:**

1. **Feladat:** Készíts egy programot, amely egy szótárban tárolja a különböző szavakhoz tartozó számokat. A feladat során hozd létre a szótárat, adj hozzá néhány kulcs-érték párt, majd írd ki a szótár tartalmát. (A szó és a szám is manuálisan legyen megadva.)
2. **Feladat:** Százszor dobunk egy hatoldalú dobókockával, határozzuk meg, hogy az egyes értékeket hányszor dobtuk! (A megoldást random számok segítségével oldjuk meg.)
3. **Feladat:** Olvassuk be a cimek.txt tartalmát egy szótárba! Az állomány sorai rendre egy nevet és egy e-mail címet tartalmaznak
   1. kinek mi az e-mail címe?
   2. melyik e-mail cím kinek a tulajdona?
   3. kérj be egy e-mail címet, ha tároljuk, add meg, hogy kié
   4. milyen e-mail címeket tárolunk?
   5. kiknek az e-mail címét ismerjük?
4. **Feladat:** Egy iskola tanulóit a menzán a nevük és az osztályuk megadása után szolgál ki a konyhás néni – így ellenőrzi, hogy a diák be van-e fizetve.
   1. Olvassuk be az ebed.txt állományt és írjuk ki a szótárban tárolt neveket a konzolra.
   2. A tanulók adatait egy struktúrában tároljuk el! Kérjen be konzolról 3 nevet és osztályt, majd írja ki, hogy a beírt személyek közül kik ebédelhetnek.
5. **Feladat:** A szotar.csv állományban néhány angol-magyar szó, illetve kifejezés párt találsz. Készíts programot, aminek a segítségével gyakorolni lehet! A programban lehessen választani, hogy angol-magyar, vagy magyar-angol gyakorlást szeretnénk-e. A választásnak megfelelően olvasd be egy szótárba a szópárokat, hogy az ellenőrzés minél hatékonyabb legyen. Számold, hogy egy-egy kifejezést hányszor fordított jól a felhasználó és azt is, hogy az összes kérdés közül mennyire tudta a helyes választ! A gyakorlásnak akkor legyen vége, ha a felhasználó válasz helyett ’0’-t ír be!
6. **Feladat:** Weblapok készítésénél nagyon fontos szempont, hogy tudjunk ugyanarra a színre több esetben is hivatkozni. Ezért hozták létre a színkódokat. A kodok.txt pontosvesszővel tagolt szövegfájl tartalmaz néhány színkódot és az angol nyelvi megfelelőjét. Készíts „kódfejtő” játékot! Menüből választhasson a játékos, hogy színkódhoz nevet, vagy névhez színkódot akar-e tippelni. A játék minden esetben 10 körből áll. Ha jó választ ad, akkor kap egy pontot és a további kérdések során ezt a szín már nem kérdezi tőle a program. A játék végén írd ki, hány pontot ért el!
7. **Feladat:** Készíts egy alkalmazást, amely egy telefonkönyvet szimulál. A telefonkönyv egy név-szám párokból álló listát tartalmaz, és lehetővé teszi új név-szám párok hozzáadását, név alapján történő keresést, név-szám párok törlését és a teljes telefonkönyv listázását.
   1. A programnak a következő funkciókat kell tartalmaznia:
      1. Egy üres telefonkönyv létrehozása, ami egy Dictionary-t használ a név-szám párok tárolására.
      2. Menü megjelenítése a következő opciókkal:
         1. Új név és szám hozzáadása a telefonkönyvhöz.
         2. Név alapján keresés a telefonkönyvben és az eredmények kiírása.
         3. Név-szám pár törlése a telefonkönyvből.
         4. Teljes telefonkönyv listázása.
         5. Kilépés a programból.
      3. Az opciók közül választva az alkalmazás végrehajtja a kiválasztott műveletet, majd visszatér a menübe, ahol újabb műveletet választhatsz.
      4. A telefonkönyv tartalma a program futása során maradjon meg.
      5. A képen szöveg, képernyőkép, Betűtípus, menü látható

         Automatikusan generált leírásPélda működés: