

Bankszámlák kezelésére alkossunk egy objektum orientált programot. Minden egyes bankszámla legyen a BankAccount osztály egy példánya. A BankAccount osztály az alábbi (privát láthatóságú) adattagokkal rendelkezzen:

- accountNumber,
- name,
- balance,

Valósítsa meg az alábbi konstruktorokat és példányszintű metódusokat! Minden metódust teszteljen a Main() metódusból!

Az összes kért metódus public láthatóságú legyen!

- BankAccount (string accountNumber, string name, decimal money) Három paraméteres konstruktor, amelyen keresztül kezdeti értékek adhatók az adattagoknak.
- BankAccount (string name, decimal money) Kétparaméteres konstruktor, mely kezdeti értéket ad a nev és egyenleg adattagoknak. A számlaszám véletlenszerűen álljon elő. Formátuma: NNNNNNNN-NNNNNNNN (8-8 karakter), ahol N egy tetszőleges számjegyet jelöl.
- BankAccount (string accountNumber, string name) Kétparaméteres konstruktor, mely kezdeti értéket ad a számlaszám és nev adattagoknak. Az egyenleg értéke legyen alapértelmezetten 0.0.
- BankAccount (string name) Egyparaméteres konstruktor, mely kezdeti értéket ad a nev adatagnak. A számlaszám véletlenszerűen álljon elő, a fentebb megadott mintának megfelelően, az egyenleg pedig 0.0 legyen.
- BankAccount () Paramétermentes konstruktor. A tulajdonos neve „Anonymus”, a számlaszámot és egyenleg értéket pedig az egyparaméteres konstruktornak megfelelően állítja be.
- string ShowMe() A metódus előállít egy stringet, aminek segítségével a számla aktuális adatai megjeleníthetők.
- bool Pay(decimal sum) Az egyenleg értéke csökken a paraméter értékével, ha elég pénz van a számlán. Ilyenkor a visszatérési érték true, ha viszont nincs elég pénz, akkor nem fizetünk és a visszatérési érték false.
- bool Increase(decimal sum) A számlára pénz érkezik, így az egyenleg értéke növekszik a paraméter értékével. Ha túlcserdulás történne, jelezzük a visszatérési értékkel

- i.     bool Interest (decimal percentage) A számla egyenlege a paraméternek megfelelő százalékkal növekszik (pl. 2,5%). Ha túlcsordulás történne, jelezzük a visszatérési értékkel
- j.     bool SendMoney(decimal amount, BankAccount WhereTo). A számláról pénzt küldök egy másik számlára, amennyiben van fedezetünk, és az ellenoldal képest fogadni. Használjuk fel a már megírt metódusokat.