

a)

- Készítsen egy *Befektetes* osztályt, amelynek a következő adattagjait kell eltárolni: *toke*, *futamiod*, *kamat*.
- Készítse el az adattagokhoz tartozó *get* és *set* metódusokat és az osztályt példányosító három-paraméteres konstruktort. Illetve a futamidő végén felvehető teljes összeget adja vissza a *getLejaratiToke()* metódus
- Az osztály implementálja a *Comparable* interfészt és az implementálandó *compareTo* metódus a befektetéseket a futamidő végén felvehető összeg alapján rangsorolja.
- Az osztály *equals* metódusa akkor térjen vissza igazsal, ha két befektetésnek megegyezik a lejáratot követő értéke.

b)

- Készítse el a *Main* osztályt, amely a megadott *befektetesek.txt*-ből olvas be értékeket és hozza létre a *Befektetes* osztály különböző példányait.
- A példányokat gyűjtse kollekcióba, amely csak a különböző összegű befektetéseket tárolja, vagyis szűrje a duplikációkat.

c)

- Készítsen egy *HiányosAdatok* kivétel osztályt, amelyet akkor dobjon el, amikor a *befektetesek.txt* fájlból olyan sorokat olvas, amelyek nem tartalmazzák a példányosításhoz szükséges összes adatot.
- A kivétel kezelésénél tájékoztassa a felhasználót, hogy a fájl hányadik soraiból nem tudott példányosítani. Vagyis a kivétel eldobásakor a kivételt a megfelelő módon példányosítsa.

d)

- Írja meg a *Main* osztályba a fájlba írást megvalósító kódot. A korábban beolvasott adatokból létrehozott kollekciót rendezze az osztályban megírt *compareTo* metódus segítségével, majd írassa ki annak elemeit a *veglegesített\_befektetesek.txt*-be.
- Értelem szerűen a *veglegesített\_befektetesek.txt* már nem tartalmazza a hiányos sorokat és a duplikációkat.

e)

- Írja meg a *Main* osztály *listaz* metódusát, amelynek paraméterként egy a billentyűzetről beolvasott értéknél nagyobb lejáratú összegű befektetéseket írja ki a képernyőre.