

Korszerű Programozási Technikák

NagyZH – 2017.12.05.

Név:

Neptun kód:

A feladat egy vállalat fizetéseket meghatározó programjának kiegészítése úgy hogy a különböző területekhez tartozó alkalmazottak éves fizetései a megfelelő módokon legyenek kiszámolva. Adottak az alábbi osztályok:

- **Employee**: egy alkalmazottat reprezentáló osztály, a következő adattagokkal: *id* (azonosító, **int**), *name* (név, **string**), *division* (terület, **string**: **WORKERS/SUPERVISORS/MANAGEMENT**), *salary* (havi fizetés, **int**), *share* (részesedés százalékban, **int**), *performance* (teljesítmény százalékban, **int**), *bonus* (bónusz fizetés, **int**), *taxBenefit* (adó jóváírás százalékban, **int**). Az adattagok konstruktoron keresztül érhetők el, illetve getterekkel kérhetők le,
- **BasicSalaryCalculator**: a vállalat alkalmazottainak alap fizetéseit meghatározó osztály, tároljuk a vállalat éves nyereségét (*companyRevenue*, **int**), amit a konstruktorral állítunk be és getterrel lekérhető. A *calculateSalary* metódus az alkalmazott havi fizetése alapján meghatározza az alap éves fizetést (havi fizetés * 12),
- **SalaryCalculatorInterface**: a programban ezt az interfészt használjuk az egyes alkalmazottak végső éves fizetésének kiszámolására. Egy **BasicSalaryCalculator*** adattagja van, amit a konstruktor állít be, valamint egy virtuális *calculateSalary* függvény, ami a paraméterben megkapott **Employee** éves fizetését adja vissza,
- **Company**: a vállalatot reprezentáló osztály a következő adattagokkal: *name* (név, **string**), *revenue* (éves nyereség, **int**), *bc* (az alapfizetést kiszámító **BasicSalaryCalculator*** példány), az adattagok konstruktorral állíthatók be.

Rekordok beolvasása

Az első részfeladat az **employees.txt** fájlban megadott alkalmazott rekordok beolvasása és tárolása a **Company** osztályban. A fájlban minden sor egy alkalmazottat reprezentál a következő adatokkal: azonosító, keresztnév, vezetéknev, terület (elfogadott értékek: **MANAGEMENT/WORKERS/SUPERVISORS**), havi fizetés, részesedés (csak **MANAGEMENT** sorokban), teljesítmény (%), csak **SUPERVISORS** sorokban), bónusz fizetés (csak **SUPERVISORS** sorokban), adó jóváírás (%), csak **WORKERS** sorokban). A fájlban minden sor megfelelő adatot tartalmaz, nem kell ellenőrizni.

1. A **Company** osztályba hozz létre egy listát, ami majd a beolvasott **Employee** példányokat tárolja, valamint készíts egy *readEmployees* metódust, ami egy fájlnevet vár paraméterként és az alábbiakban részletezett módon beolvassa a fájlból a rekordokat és eltárolja a listában, (3 pont)
 - a. az **Employee** osztály legyen egy *employee* nevű névtérben (ennek megfelelően egészítsd ki a meglévő és a létrehozott osztályokat, hogy használni tudják az **Employee** osztályt), (2 pont)
 - b. azok az adatok, amik nem léteznek minden típusnál (részesedés, teljesítmény, bónusz, adó jóváírás) és hiányoznak az adott sorból 0 értékkel szerepeljenek, (1 pont)
 - c. a név adattagot a keresztnév + vezetéknev összefűzésével add meg, (1 pont)
 - d. csak a **MANAGEMENT/WORKERS/SUPERVISORS** értékek elfogadottak a terület adattagnál, ha nem ilyen van megadva, akkor írd ki egy hibaüzenetet, de a beolvasás folytatódjon, (1 pont)
 - e. + a fenti hibaüzenet mellett egy a **Company** osztályban létrehozott saját kivétel osztályt is használj úgy hogy a beolvasás során egy listába gyűjtsd össze a hibás sorok azonosítóit és a beolvasás végén ezt a listát add át a kivétel osztálynak és dobd el a kivételt (amennyiben volt hiba), ami a példakimenetben megadott módon írja ki a hibát. A *main* függvényben kezeld az esetleges kivételt a *readEmployees* függvényhívás körül. (3 pont)

Fizetések meghatározása

A programban a fizetéseket a **SalaryCalculatorInterface** segítségével szeretnénk meghatározni, de úgy hogy használjuk a már létező **BasicSalaryCalculator**-t is. A feladat az **Adapter** mintának megfelelően létrehozni mind a három terület típushoz egy-egy adapter osztályt, ami megvalósítja a **SalaryCalculatorInterface** *calculateSalary* metódusát és a számolás során használja a **BasicSalaryCalculator** adattagot (az adapter osztályok konstruktorában állítsd be az interfész ősosztály segítségével). Mindegyik esetben az éves alapfizetést a **BasicSalaryCalculator** segítségével számold ki.

2. Az alábbi adapter osztályokat készítsd el:

- a. **WorkerSalaryCalculatorAdapter**: a **WORKERS** területhez tartozó alkalmazottak éves fizetését számolja ki. Az éves alapfizetésnek számold ki az adó jóváírásnak megfelelő részét és ezt még add hozzá az éves alapfizetéshez (alapfizetés + alapfizetés * adó jóváírás %), (2 pont)
 - b. **SupervisorSalaryCalculatorAdapter**: a **SUPERVISORS** területhez tartozó alkalmazottak éves fizetését számolja ki. Az éves alapfizetéshez add hozzá a bónusz fizetés értékét, akkor ha a teljesítmény legalább 70% különben marad az alap, (2 pont)
 - c. **ManagerSalaryCalculatorAdapter**: a **MANAGEMENT** területhez tartozó alkalmazottak éves fizetését számolja ki. Az éves alapfizetéshez add hozzá a vállalat éves nyereségének a részesedés adattagnak megfelelő részét (alapfizetés + éves nyereség * részesedés %) (2 pont)
3. A **Company** osztályba hozz létre egy asszociatív tároló adattagot, aminek **string** típusú a kulcsa és **SalaryCalculatorInterface*** típusú az értéke. A **Company** konstruktorában hozz létre mindegyik típusú adapterből egy-egy mutató példányt és tárol el ezeket az asszociatív tömbben a típusnak megfelelő kulccsal (**MANAGEMENT/WORKERS/SUPERVISORS**), (3 pont)
4. A **Company** osztály destruktoraiban gondoskodj arról, hogy a dinamikusan foglalt adatok (adapterek, **BasicSalaryCalculator**) törlése megtörténjen. (1 pont)

Fizetések kiszámítása és kiírása

5. A **Company** osztályba készíts egy paraméter nélküli, **void calculateSalaries** metódust, ami az alábbi módon kiszámolja a vállalat alkalmazottainak éves fizetését az adapterek segítségével.
- a. Menj végig az alkalmazottak listáján és mindegyik esetben az alkalmazott **division** adattagjának megfelelő adapter (asszociatív tömbből lekérve) segítségével számold ki és írd ki az alkalmazott azonosítóját és az éves fizetését, (2 pont)
 - b. + a függvényen belül hozz létre három listát a terület típusoknak (**MANAGEMENT/WORKERS/SUPERVISORS**) megfelelően és az alkalmazottakat válogasd szét ezekbe a listákba a terület (**division**) adattagjuk alapján, (2 pont)
 - c. + a redundáns adattárolás elkerülése érdekében valamilyen hivatkozást (mutató, iterátor) tárolj a fenti listákban, (1 pont)
 - d. + készíts egy lambda függvényt, ami paraméterként megkap egy alkalmazott listát és egy adapter mutatót (**SalaryCalculatorInterface***), a függvénytörzsben pedig végigmegy a listán és kiírja az alkalmazott azonosítóját és az adapter segítségével kiszámolt éves fizetését, továbbá minden iterációkor aludjon a szál 100 milliszekundumot, (2 pont)
 - e. + indíts el három szálát, amelyek a fenti lambda függvényt futtatják, paraméterként mindegyik megkap egy-egy szétválogatott alkalmazott listát, illetve a megfelelő adapter mutatót az asszociatív tömbből lekérve, figyelj a szálak megfelelő visszacsatolására a fő szálhoz, (2 pont)
 - f. + kölcsönös kizárás segítségével oldd meg, hogy a három szál **cout**-ra történő kiírásai szinkronizáltan történjenek (ne keveredjenek a sorok). (2 pont)