**AwardedStudent feladat**

**Osztályok fejlesztése**

1. Hozzon létre a modell rétegben osztályt AwardedStudent (tanulmányi eredményeik miatt alapítványi támogatásban, ösztöndíjban részesített diákok) néven azon diákok számára, akik ösztöndíjat kapnak!

2. Az ösztöndíjas diákoknak tudjuk a nevét, életkorátt és az ösztöndíj értékét (Ft-ban) és hogy az ösztöndíj egyszeri juttatás vagy havonta járó juttatás!

3. Készítse el a szükséges konstruktorokat!

4. Készítse el az osztályt szöveggé alakító metódusát, úgy, hogy példányosítás után az adatok megjelenítésekor a pontosan következő szöveg jelenjen meg:

"Ügyes Tünde 17 éves és 15000 Ft ösztöndíjat nyert! Az ösztöndíj egyszeri juttatás!"

Jelenítse meg a fő programmal a minta szerint a diák adatait! Ha az ösztöndíj havi juttatás, akkor a szöveg vége „Az ösztöndíj havi juttatás!” legyen.

5. Hozzon létre még egy díjazott diákot aki „Dolgos Dénes, 18 éves és 5000 Ft havi ösztöndíjba részesül. Vezérlési szerkezettel jelenítse meg a fő programban, hogy melyik diák az idősebb! A kód kimenetei a következők közül lehetnek:

"Ügyes Tünde idősebb, mint Dolgos Dénes!"

"Dolgos Dénes idősebb, mint Ügyes Tünde!"

"A két diák ugyan annyi éves!"

A kiírás akkor is helyes legyen, ha más nevű diákokról van szó!

6. Készítsen az osztályban csak olvasható peropty-t, amely akkor ad igaz értéket, ha a támogatott diák már betöltötte a 18. életévét!

7. Készítsen az osztályban statikus metódust IsYounger néven, amely megadja, hogy a metódus paraméterében megadott első diák fiatalabb-e a paraméterben megadott második diáknál! A kódot TDD módszerben készítse el! A tesztesetek megadására készített táblázatba, amely tartalmaz legalább három tesztesetet az elvárt kimenetelekkel! Készítse el a teszteket! Írja meg az IsYounger metódus kódját úgy, hogy egyetlen teszteset se legyen hibás!

**Adattárak (repository) fejlesztése**

8. Készítsen el egy új osztály AwardedStudent Repo néven, amelyben több ösztöndíjjal rendelkező diák tárolására alkalmas adatstruktúrát használjon! A fő programban adjon hozzá legalább három minden adatával rendelkező diákot a repóhoz! A feladatok megoldása közben, ha szükségét érzi, újabb adatokat adhat a repóhoz! A repository kódját úgy készítse el, hogy alkalmazza a réteges programozás elvét, vagyis annak adataihoz, csak a publikus metódusin keresztül lehessen hozzáférni. Azon feladatok megoldásai, amelyek sértik ezt az elvet, nem kerülnek pontozásra!

9. Írjon metódust, amely megadja a repóban lévő diákok ösztöndíjainak átlagát! A fő programban jelenítse meg az eredményt két tizedes jegy pontossággal az alábbi formában: „Az átlagos ösztöndíj x.xx Ft!”

11. Egészítse ki az ösztöndíjjal rendelkező diákok osztályát egy olyan property-vel amely tartalmazza a diák osztályát is! A pótlékot a diákok 10 hónapig kapják egy évben!

a) Jelenítse meg a fő programban felhasználóbarát módon, hogy egyszeri jutatásokból (ösztöndíjból) mennyi kifizetés történt egy évben!

b) Jelenítse meg felhasználóbarát módon a fő programban, hogy összesen mennyit kell fizetni egy évben az ösztöndíjasoknak!

c) Jelenítse meg a fő programban felhasználóbarát módon, hogy osztályonként összesen mennyit ösztöndíj kerül kifizetésre egy évben!

d) Kérje be egy iskolai osztály azonosítóját! Határozza meg, hogy az iskolai osztályban mennyi volt a legnagyobb és legkisebb *havi* ösztöndíj (nem egyseri kifizetés)!

A <x> osztályban a legnagyobb ösztöndíj < x> Ft, a legkisebb ösztöndíj <x> Ft.

Ha az osztály nem létezik, írja ki „Az osztály nem létezik!” üzenetet!

12. Bővítse úgy a megfelelő osztályt, hogy az ösztöndíjasok nemét is ismerjük! Írjon megoldást a kódban, amely meghatározza nemenként az ösztöndíjasok számát. Ha lehetséges a feladatot egy metódus segítségével oldja meg! Az eredményt a következő formátumba jelenítse meg:

Ösztöndíjasok megoszlása nemenként:

A fiú diákok száma x fő.

A lány diákok száma x fő.

13. Azoknak, akik már betöltötték a 18. életévüket adóigazolást kell kiadni. Írja ki a képernyőre, kik ezek a diákok és mennyi az ösztöndíjük! Az adatokat a képernyőn a fő program segítségével írja ki! A metódust úgy írja meg, hogy az ne string visszatérési értékkel rendelkezzen!

14. Írjon megoldást a kódban, amely nemek és iskolai osztályok megoszlásában kiírja a *havi* (nem egyseri kifizetés) legnagyobb és legkisebb ösztöndíj értékét!

15. Fejlesszen metódust Youngest AwardedStudent () néven amelyet TDD módszerrel fejlesszen ki. A metódus a legfiatalabb ösztöndíjas nevét adja vissza! Legalább három tesztesetet esetén készítse el a tesztesetek táblázatát! Írja meg a metódus kódját úgy, hogy a tesztesetek között ne legyen olyan, amelyik elbukik!

**AdministratorTeacher feladat**

**Osztályok fejlesztése**

1. Hozzon létre osztály AdministratorTeacher néven azon tanárok számára, akik adminisztrátori jogosultsággal kezelik az alkalmazást!

2. Az ilyen adminisztrátor tanárok esetén tudjuk a tanár nevét, születésének idejét, a pótlék összegét, amiért ezt a tevékenységet végzi, és azt hogy rendelkezik-e törlés joggal a rendszerben!

3. Készítse el a szükséges konstruktorokat!

4. Készítse el az osztályt szöveggé alakító metódusát, úgy, hogy példányosítás után az adatok megjelenítésekor a pontosan következő szöveg jelenjen meg:

"Admin Antal (1995.05.05), 7000 Ft pótlékot kap. Rendelkezik törlés joggal!"

Jelenítse meg a fő programban a minta szerint a tanár adatait! (Ha nem rendelkezik a tanár törlés joggal, akkor a „Nem rendelkezik törlés joggal!” legyen a szöveg vége.)

5. Hozzon létre még egy adminisztrátor tanárt aki „Vezető Vilmos, (1992.02.02-én született) és 5000 Ft havi ösztöndíjba részesül, és nem rendelkezik törlési joggal. Vezérlési szerkezettel jelenítse meg a főprogramban, hogy melyik tanár az idősebb! A kód kimenetei a következők közül lehetnek:

"Admin Antal idősebb, mint Vezető Vilmos!"

"Vezető Vilmos idősebb, mint Admin Antal!"

"A két tanár ugyan annyi éves!"

A kiírás akkor is helyes legyen, ha más nevű adminisztrátor tanárról van szó!

6. Készítsen az osztályban csak olvasható peropty-t, amely akkor ad igaz értéket, ha az adminisztrátor törlési joggal nem rendelkező, aki a munkájáért pótlékot sem kap!

7. Készítsen statikus metódust IsBetterPaid amely megadja, hogy a metódus paraméterében megadott első tanár jobban fizetett-e a második tanárnál! A kódot TDD módszerben készítse el! A tesztesetek megadására készített táblázatba készítsen legalább három tesztesetet az elvárt kimenetekkel! Készítse el a teszteket! Írja meg az IsBetterPaid metódus kódját úgy, hogy a megírt tesztesetek mindegyike hibátlan legyen!

**Adattárak (repository) fejlesztése**

8. Készítsen el egy új osztály AdministratorTeacherRepo néven, amelyben több adminisztrátor tanár tárolására alkalmas adatstruktúrát használjon! A fő programban adjon hozzá legalább három minden adatával rendelkező adminisztrátor tanárt a repóhoz! A feladatok megoldása közben, ha szükségét érzi, újabb adatokat adhat a repóhoz! A repository-t úgy készítse el, hogy alkalmazza a réteges programozás elvét, vagyis annak adataihoz, csak a publikus metódusin keresztül lehessen hozzáférni. Azon feladatok megoldásai, amelyek sértik ezt az elvet, nem kerülnek pontozásra!

9. Írjon metódust, amely megadja a repóban lévő tanárok pótlékainak átlagát! A fő porgamban jelenítse meg az eredményt két tizedes jegy pontossággal az alábbi formában: „A átlagos pótlék x.xx Ft!”

11. Ha a pótlékot az adminisztrátor tanárok 11 hónapig kapják egy évben, akkor jelenítse meg, hogy összesen mennyit kell fizetni egy évben összesen az adminisztrátoroknak a pótlékok kifizetésekor! Írjon metódust a repository rétegben a feladat megoldására! A megoldást a következő formátumban jelenítse meg: „Az összes kifizetett pótlék egy évben x Ft.”

12. Írjon megoldást a kódban az törlési joggal rendelkező és nem rendelkező adminisztrátorok számának meghatározására. Ha lehetséges a feladatot egy metódus segítségével oldja meg! Az eredményt a következő formátumba jelenítse meg:

Adminisztrátorok száma x fő.

Törlési joggal rendelkező adminisztrátorok száma x fő.

13. Írjon megoldást a kódban, amelyik a legnagyobb pótlékkal rendelkező törlési joggal rendelkező adminisztrátor nevét adja meg! Ha lehetséges a megoldást egy új metódus megírásával és felhasználásával oldja meg!

Az eredményt a következő formában jelenítse meg:

„A legnagyobb pótlékkal rendelkező törlési joggal rendelkező adminisztrátor <név>.”

14. Fejlesszen metódust OldestAdministrator() néven amelyet TDD módszerrel fejlesszen ki. A metódus a legidősebb törlési joggal rendelkező vagy nem rendelkező adminisztrátor tanár nevét adja vissza! Legalább három tesztesetet esetén készítse el a tesztesetek táblázatát! Írja meg a metódus kódját úgy, hogy a tesztesetek között ne legyen olyan, amelyik elbukik!

**HeadTeacher feladat**

**Osztályok fejlesztése**

1. Hozzon létre osztály HeadTeacher néven azon tanárok számára, akik osztályfőnöki jogosultsággal kezelik az alkalmazást!

2. Az ilyen adminisztrátor tanárok esetén tudjuk az osztályfőnök nevét, születésének idejét, a pótlék összegét, amiért ezt a tevékenységet végzi, és azt hogy ő pótosztályfőnök-e!

3. Készítse el a szükséges konstruktorokat és az osztályt szöveggé alakító metódust!

4. Készítse el az osztályt szöveggé alakító metódusát, úgy, hogy példányosítás után az adatok megjelenítésekor a pontosan következő szöveg jelenjen meg:

"Oszi Ottó (1994.04.04), 5000 Ft pótlékot kap. Pótosztályfőnök!"

Jelenítse meg a fő programban a minta szerint az osztályfőnök adatait! (Ha a tanár nem pótosztályfőnök, akkor a „Nem pótosztályfőnök!” legyen a szöveg vége.

5. Hozzon létre még egy osztályfőnököt aki „Határozott Hedvig, 5000, 1996.06.06-án született és 8000 Ft pótlékot kap! Nem pótosztályfőnök! Vezérlési szerkezettel jelenítse meg a főprogramban, hogy melyik osztályfőnök az idősebb!

"Oszi Ottó idősebb, mint Határozott Hedvig!"

"Határozott Hedvig idősebb, mint Oszi Ottó!"

"A két tanár ugyan annyi éves!"

A kiírás akkor is helyes legyen, ha más nevű osztályfőnökről van szó!

6. Készítsen olvasható peropty-t az osztályban, amely akkor ad igaz értéket, ha a tanár pótlékot nem kap és pótosztályfőnök!

7. Készítsen az osztályban statikus metódust IsOlder amely megadja, hogy a metódus paraméterében megadott első osztályfőnök idősebb-e a második paramméterben megadott osztályfőnöknél! A kódot TDD módszerben készítse el! A tesztesetek megadására készített táblázatba készítsen legalább három tesztesetet az elvárt kimenetekkel! Készítse el a teszteket! Írja meg az IsOlder metódus kódját úgy, hogy a megírt tesztesetek mindegyike hibátlan legyen!

**Adattárak (repository) fejlesztése**

8. Készítsen el egy új osztály HeadTeacherRepo néven, amelyben több osztályfőnök tárolására alkalmas adatstruktúrát használjon! A fő programban adjon hozzá legalább három minden adatával rendelkező osztályfőnököt a repóhoz! A feladatok megoldása közben, ha szükségét érzi, újabb adatokat adhat a repóhoz! A repository-t úgy készítse el, hogy alkalmazza a réteges programozás elvét, vagyis annak adataihoz, csak a publikus metódusin keresztül lehessen hozzáférni. Azon feladatok megoldásai, amelyek sértik ezt az elvet, nem kerülnek pontozásra!

9. Írjon metódust, amely megadja a repóban lévő osztályfőnökök által kapott pótlékok átlagát! A fő porgamban jelenítse meg az eredményt két tizedes jegy pontossággal az alábbi formában: „A átlagos pótlék x.xx Ft!”

11. Ha a pótlékot az osztályfőnök 10 hónapig kapják egy évben, akkor jelenítse meg, hogy összesen mennyit kell fizetni egy évben összesen az osztályfőnököknek a pótlékok kifizetésekor! Írjon metódust a repository rétegben a feladat megoldására! A megoldást a következő formátumban jelenítse meg: „Az összes kifizetett pótlék egy évben x Ft.”

12. Írjon megoldást a kódban az osztályfőnökök és pótosztályfőnökök számának meghatározására. Ha lehetséges a feladatot egy metódus segítségével oldja meg! Az eredményt a következő formátumba jelenítse meg:

Osztályfőnökök száma x fő.

Pótosztályfőnök száma x fő.

13. Írjon megoldást a kódban, amelyik a legfiatalabb osztályfőnök (nem pótosztályfőnök) nevét adja meg! Ha lehetséges a megoldást egy új metódus megírásával és felhasználásával oldja meg!

Az eredményt a következő formában jelenítse meg:

„A legfiatalabb osztályfőnök <név>.”

14. Fejlesszen metódust MaxExtraPaiment néven amelyet TDD módszerrel fejlesszen ki. A metódus a legnagyobb pótlékot adja meg, amellyel egy osztályfőnök vagy pótosztályfőnök rendelkezik! Legalább három tesztesetet esetén készítse el a tesztesetek táblázatát! Írja meg a metódus kódját úgy, hogy a tesztesetek között ne legyen olyan, amelyik elbukik!

**SchoolDirector feladat**

**Osztályok fejlesztése**

1. Hozzon létre osztály SchoolDirector néven!

2. Az osztály tárolja az iskolai igazgató nevét, igazgatóvá való kinevezésének dátumát és azt, és a pozícióéért járó plusz juttatás összegét illetve azt, hogy kinevezett vagy ideiglenes státuszban végzi ezt a munkát!

3. Készítse el a szükséges konstruktorokat és az osztályt szöveggé alakító metódust!

4. Példányosítson két iskolai igazgatót úgy, hogy az adatainak megjelenítésekor a következő szöveg jelenjen meg:

"Vezető Vilmos (kinevezés: 2010.10.10, 12 éve igazgató). Kinevezett igazgató!"

"Irányító Ida (kinevezés: 2011.11.11, 11 éve igazgató). Ideiglenesen kinevezett igazgató!”

Jelenítse meg a fő programban az adatokat!

5. Vezérlési szerkezettel jelenítse meg a fő programban, hogy melyik igazgató az idősebb! A kimenetek a következők közül lehetnek:

"Vezető Vilmos hosszabb időt volt tanár, mint Irányító Ida!"

"Irányító Ida hosszabb időt volt tanár, mint Vezető Vilmos!"

"A két vezető ugyan annyi időt volt tanár!"

A kiírás akkor is helyes legyen, ha más nevű igazgatóról van szó!

6. Készítsen csak olvasható peropty-t az osztályban, amely akkor ad igaz értéket, ha az iskola vezető még nem kezdte el a munkáját vagy még egy egész évet nem dolgozott és ideiglenesen kinevezett igazgató!

7. Készítsen az osztályban statikus metódust IsMoreExperienceDirector amely megadja, hogy a metódus paraméterében megadott első igazgató nagyobb tapasztalattal rendelkező igazgató a második paramméterben megadottnál! A kódot TDD módszerben készítse el! A tesztesetek megadására készített táblázatba készítsen legalább három tesztesetet az elvárt kimenetekkel! Készítse el a teszteket! Írja meg az IsMoreExperienceDirector metódus kódját úgy, hogy a megírt tesztesetek mindegyike hibátlan legyen!

**Adattárak (repository) fejlesztése**

8. Készítsen el egy új osztály SchoolDirectorRepo néven, amelyben több iskolai vezető tárolására alkalmas adatstruktúrát használjon! A fő programban adjon hozzá legalább három havi fizetéssel rendelkező vezetőt a repóhoz! A feladatok megoldása közben, ha szükségét érzi, újabb adatokat adhat a repóhoz! A repository-t úgy készítse el, hogy alkalmazza a réteges programozás elvét, vagyis annak adataihoz, csak a publikus metódusin keresztül lehessen hozzáférni. Azon feladatok megoldásai, amelyek sértik ezt az elvet, nem kerülnek pontozásra!

9. Egészítse ki a SchoolDirector osztályt egy olyan tulajdonsággal, amely a vezetők havi fizetését tárolja. Írjon metódust AverageSalary() néven, amely megadja a repóban lévő igazgatók átlagfizetését! Jelenítse meg az eredményt két tizedes jegy pontossággal az alábbi formában: „Az átlagfizetés x.xx Ft!”

11. Ha a fizetésüket az iskolaigazgatók 12 hónapig kapják egy évben, akkor jelenítse meg, hogy összesen mennyit kell fizetni egy évben összesen az összes igazgatónak! Írjon metódust a repository rétegben a feladat megoldására! A megoldást a következő formátumban jelenítse meg: „Az összes kifizetett bér egy évben x Ft.”

12. Írjon megoldást a kódban a kinevezett és ideiglenesen kinevezett igazgatók legnagyobb fizetésének meghatározására. Ha lehetséges a feladatot egy metódus segítségével oldja meg! Az eredményt a következő formátumba jelenítse meg:

Legnagyobb fizetések

Kinevezett igazgatók között: x Ft.

Ideiglenesen kinevezett igazgatók között: x Ft.

13. Írjon megoldást a kódban, amelyik a legidősebb iskolaigazgató nevét adja meg! Ha lehetséges a megoldást egy új metódus megírásával és felhasználásával oldja meg! Az eredményt a következő formátumban jelenítse meg:

„A legidősebb iskolaigazgató <név>.”

14. Fejlesszen metódust GetTemporarily AppointedDirectorNumber néven TDD módszerrel fejlesszen ki! A metódus az ideiglenesen kinevezett igazgatók számát adja meg! Legalább három tesztesetet esetén készítse el a tesztesetek táblázatát! Írja meg a metódus kódját úgy, hogy a tesztesetek között ne legyen olyan, amelyik elbukik!

SpecialNeedsTeacher feladat

Az iskolában gyógypedagógusok is dolgoznak. Készítsen az előző feladatokhoz hasonló kódot egy város összes iskolájában dolgozó gyógypedagógusok kezelésére!