**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 1525 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 egyszeri juttatásba (ösztöndíjban) 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 kiírásban a nevek változó adatok!

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!

Bővítse úgy a megfelelő osztályt, hogy az ösztöndíjasok nemét is ismerjük!

Egészítse ki az ösztöndíjjal rendelkező diákok osztályát egy olyan property-vel amely tartalmazza a diák osztályát is! Pl. „9.a”

9. Adja hozza Üveges Tünde és Dolgos Dénes-t a repóhoz!

Üveges Tünde a 9.a osztályba jár és nő.

Dolgos Dánes a 9.a osztályba jár és férfi.

Adja hozzá a következő adatokat a repóhoz:

Szorgalmas Szonja 16 éves, 6527 Ft ösztöndíjat kap havonta, "9.b" osztályba jár és nő!

Mindenttudó Misi 16 éves, 6514 Ft ösztöndíjat kap havonta, "9.b" osztályba jár és férfi!

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!

Fontos tudni, hogy a havi ösztöndíjat a diákok 10 hónapig kapják egy iskolai évben!

9. [Havi ösztöndíjak átlaga]

Tekintse az egy hónapban kifizetett havi ösztöndíjakat (nem egyszeri juttatásokat (ösztöndjakat)! Határozza meg ezen kifizetések havi átlagát két tizedes jegy pontossággal. Az eredményt „A havi ösztöndíjak átlaga xx.xx Ft” formában írja ki a képernyőre!

10. [Nemenként az ösztöndíjasok száma]

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.

11. [Egyszeri juttatások egy évben]

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

12. [Iskolaévben kifizetett ösztöndíjak összege]

Jelenítse meg felhasználóbarát módon a fő programban, hogy egy iskolaévben összesen mennyit kell fizetni az ösztöndíjasoknak (egyszeri juttatás (ösztöndíj) és havi ösztöndíj együtt)!

13. [Iskolai osztályok]

a) Kérje be egy iskolai osztály azonosítóját! Határozza meg, hogy az iskolai osztályban mennyi volt a legnagyobb és legkisebb ösztöndíj!

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!

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

14. [Adóigazolás]

Azoknak, akik már betöltötték a 18. életévüket adóigazolást kell kiadni. Írja ki a képernyőre, a diák nevét 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!

15. [Legfiatalabb diák]

Fejlesszen metódust GetYoungestAwardedStudentName() 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 havi 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), 7527 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 5224 Ft havi ösztöndíjba részesül, és nem rendelkezik törlési joggal. Vezérlési szerkezettel jelenítse meg a fő programban, 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 adminisztrátor ugyan annyi éves!"

A kiírásban a nevek változó adatok!

6. Készítsen az osztályban csak olvasható peropty-t, amely akkor ad igaz értéket, ha az adminisztrátor munkájáért pótlékot nem 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 a metódus kódját úgy, hogy mindegyik teszteset az elvárt eredményt adja!

**Adattárak (repository) fejlesztése**

8. Készítsen el egy új osztály AdministratorTeacherRepo néven, amelyben több adminisztrátor tanár tárolására alkalmas adatstruktúrát használjon!

Vegye fel a repóba a már létrehozott adminisztrátorokat!

Vegye fel „Bizonytalan Béla”, törlési joggal rendelkező adminisztrátort, aki 1997.02.12-én született és 5545 Ft pótlékkal rendelkezik és „Helytáló Hedvig” törlési joggal nem rendelkező adminisztrátort, aki 1999.09.09-én született és 6524 Ft pótlékot kap.

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. [Pótlékok átlaga]

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

11. [Összes kifizetett pótlék]

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. [Törlési joggal rendelkezők és nem rendelkezők száma]

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.

Kérje be egy adminisztrátornak a nevét felhasználóbarát módon. Keresse meg az adminisztrátor születési idejét és írja ki a képernyőre a következő formában!

A megadott X Y adminisztrátor születésé ideje xx.xx.xx.

Ha az adminisztrátor neve nem található, akkor jelenítse meg a következő üzenetet „A megadott nevű adminisztrátor nem található!”

14. [Legnagyobb pótlék]

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

Az eredményt a következő formában jelenítse meg:

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

15. [Legnagyobb és legkisebb pótlék]

Határozza meg a törlési joggal rendelkező és nem rendelkező adminisztrátorok legnagyobb és legkisebb pótlékát!

16. [Februárban született adminisztrátorok]

Írja ki a képernyőre a februárban született adminisztrátorok nevét és születési évét!

17. [Pótlékkal rendelkező és nem rendelkező adminisztrátorok megoszlása]

Határozza meg a pótlékkal rendelkező és nem rendelkező adminisztrátorok megoszlását! (Az adminisztrátorok hány százaléka kap és hány százaléka nem kap pótlékot!). A megoldást felhasználó barát módon írja ki!

18. [Legkisebb pótlékkal rendelkező adminisztrátor neve]

Fejlesszen metódust a legkisebb pótlékkal rendelkező adminisztrátor néven meghatározására amelyet TDD módszerrel fejlesszen ki! Legalább három tesztesetre 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 havi 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 1996.06.06-án született és 8000 Ft pótlékot kap! Nem pótosztályfőnök! Vezérlési szerkezettel jelenítse meg a fő programban, 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 IsBothHeadTeacherAssistant néven, amely megadja, hogy a metódus paraméterében megadott első és második megadott osztályfőnök pótosztályfőnök-e! A kódot TDD módszerben készítse el! A tesztesetek megadására készített táblázatba készítsen legalább három tesztesetet az elvárt kimenetekkel! Készítse el a teszteket! Írja meg a 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!

Egészítse ki a repót a már létrehozott osztályfőnökökkel és pótosztályfőnökökkel!

Vegye fel „Gyermekbarát Gyulát (1994.04.24), 0 Ft pótlék és pótosztályfőnök” valamit „Ideges Ida (1994.05.05), 8400 Ft pótlék és nem pótosztályfőnök” tanárt a repóba!

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 (osztályfőnök és pótosztályfőnökök is) által kapott pótlékok átlagát! A fő programban jelenítse meg az eredményt egész számra kerekítve alábbi formában: „A pótlékok átlaga xxxxx Ft!”

11. Ha a pótlékot az osztályfőnök 10 hónapig kapják egy évben, akkor jelenítse meg, hogy összesen mennyit kell fizetni egy évben az osztályfőnököknek (osztályfőnök és pótosztályfőnökök is) 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.

Kérje be egy osztályfőnök nevét felhasználóbarát módon. Keresse meg az osztályfőnököt és írja ki a képernyőre, hogy ő pótosztályfőnök vagy osztályfőnök!

Ha a tanár neve nem található, akkor jelenítse meg a következő üzenetet „A megadott nevű osztályfőnök nem található!”

14. Vegyen fel a megfelelő osztályban egy property-t, amelyből kiderül a tanár neme is! Határozza meg nemenként a legnagyobb és legkisebb pótlék összegét. A megoldást felhasználó barát módon jelenítse meg!

15. 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>.”

16. Határozza meg a február hónapban született osztályfőnök és pótosztályfőnök nevét és születési évét! A megoldást felhasználóbarát módon jelenítse meg a képernyőn!

17. Határozza meg a pótlékot nem kapó osztályfőnökök nevét és születési évét! A megoldást felhasználó barát módon jelentse meg a képernyőn!

18. Fejlesszen metódust MaxExtraPaiment néven amelyet TDD módszerrel fejlesszen ki. A metódus a legnagyobb pótlékkal rendelkező tanár nevét adja meg, amellyel egy osztályfőnök vagy pótosztályfőnök rendelkezik! Legalább három tesztesetre 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, igazgatói pályázatának dátumát, hány éve van ebben a pozícióban, a pozícióért járó havi 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 (pályázat dátuma: 2010.10.10, 12 éve igazgató, 150000 Ft pótlék). Kinevezett igazgató!"

"Irányító Ida (pályázat dátuma: 2011.11.11, 11 éve igazgató, 120000 Ft pótlék). 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 idősebb, mint Irányító Ida!"

"Irányító Ida idősebb, mint Vezető Vilmos!"

"A két vezető ugyan annyi idős!"

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 (az a nagyobb tapasztalattar rendelkező igazgató, aki több éve 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 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!

Adja hozzá a repóhoz Vezető Vilmos-t és Irányító Idát-t!

Adja hozzá a repóhoz a következő adatokat:

„Átlátó Ákos, pályázat dátuma 2021.10.20, 0 éve igazgató, 60000 Ft pótlék és ideiglenesen kinevezett igazgató!”

„Hajszoló Hedvig, pályázat dátuma 2020.11.01, 2 éve ideiglenesen kinevezett igazgató!”

„Precíz Péter, pályázat dátuma 2015.04.04, 5 éve kinevezett igazgató

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 meghatározza az iskolai igazgatók (igazgatók és ideiglenesen kinevezett igazgatók) igazgatóként eltöltött idejének átlagát! Jelenítse meg az eredményt két tizedes jegy pontossággal az alábbi formában: „Az igazgatóként eltöltött idő x.xx év!”

11. Ha az összes igazgató a fizetését 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 eredményt: „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. Készítsen listát azokról az ideglenesen vagy kinevezett igazgatókról, akik még nem kezdték el a munkájukat vagy még egy egész évet nem dolgoztak kinevezésük óta! A listában az igazgatók neve és pályázatának beadási időpontja jelenjen meg felhasználóbarát módon!

14. Készítsen listát azokról az igazgatókról, akik a pályázatukat októberben vagy novemberben adták le!

15. Egészítse ki a megfelelő osztályt, hogy az iskolaigazgatók nemét is tárolják! Határozza meg nemenként a legnagyobb és legkisebb havi pótlék összegét (kinevezett és ideiglenesen kinevezett igazgatók esetén)! A megoldást felhasználóbarát módon jelenítse meg!

16. Írjon megoldást a kódban, amelyik a legnagyobb havi pótlékkal rendelkező iskolaigazgató nevét adja meg! „A legnagyobb havi pótlékkal rendelkező iskolaigazgató <név>.”

14. Fejlesszen metódust GetTemporarilyAppointedDirectorNumber 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!