

A Százholdas Pagony új élményfürdőjében az alábbi árak érvényesek:

Jegytípus	Alapár	Százholdas Pagony lakóinak*
Felnőtt jegy	2 900 Ft	2 100 Ft
Gyermekjegy 3 éves korig	100 Ft	100 Ft
Gyermekjegy 3-6 éves korig	1 900 Ft	1 600 Ft
Diákjegy	2 400 Ft	1 900 Ft
Nyugdíjas jegy	2 400 Ft	1 900 Ft
Délutáni belépőjegy 16:00 után	1 900 Ft	1 600 Ft
Úszójegy (07:00 és 09:00 közötti, valamint 18:30 és 20:00 közötti belépés esetén)	1 000 Ft	1 000 Ft

*azoknak, akiknek a cím attribútum értéke „Százholdas Pagony”

Létezik **családi jegy** is, amely az *úszójegy kivételével* minden jegytípus árából kedvezményt kínál a családok számára. Családi jegyet olyan csoportok (családok) válthatnak, amelynek tagjai között van legalább két felnőtt jegyre, vagy egy-egy felnőtt és nyugdíjas jegyre jogosult személy (de összesen legfeljebb négyen közülük), valamint legalább egy gyermek- vagy diákjegyre jogosult személy. Az azonos kategóriába tartozó gyermek- és diákjegyek esetén csak az első kettőért kell fizetni, a harmadik (és onnantól az összes többi) ingyen van (pl. három diák esetén 2 x 2 400 Ft csak a számítás alapja az alapár figyelembe vételével, de két 3-6 év közti gyermek és egy diák esetén mindhárom jegyet figyelembe vesszük). A családi jegy ára a családtagokra vonatkozó jegyárak összegének a 75%-a, száz forintokra **kerekítve**.

1. példa: egy, a Százholdas Pagonyon kívül lakó, két felnőttet és egy-egy 3 év alatti és 3 és 6 év közötti gyermeket számláló négytagú család esetén:

2 db felnőtt jegy: $2 \times 2\,900 \text{ Ft} = 5\,800 \text{ Ft}$
 1 db gyermekjegy 3 éves korig: 100 Ft
 1 db gyermekjegy 3-6 éves korig: 1 900 Ft
 Összesen: 7 800 Ft
 Családi jegy ára: $7\,800 \text{ Ft} \times 75\% (= 5\,850 \text{ Ft})$, százásokra kerekítve: 5 900 Ft

2. példa: egy, a 8 éves diák gyermekével a Százholdas Pagonyban élő anyuka a gyerekekkel és a nyugdíjas nagymamával (aki a Pagonyon kívül él) látogat el az élményfürdőbe:

1 db felnőtt jegy a Százholdas Pagony lakóinak: 2 100 Ft
 1 db nyugdíjas jegy: 2 400 Ft
 1 db diákjegy a Százholdas Pagony lakóinak: 1 900 Ft
 Összesen: 6 400 Ft
 Családi jegy ára: $6\,400 \text{ Ft} \times 75\% = 4\,800 \text{ Ft}$ (eleve százásokra kerekített).

3. példa: egy hattagú család (2 felnőtt plusz 4 gyerek: 5, 7, 11 és 13 évesek) du. 16 óra után érkeznek, és nem a Pagonyban laknak:

2 db felnőtt jegy 16:00 után: $2 \times 1\,900 \text{ Ft} = 3\,800 \text{ Ft}$
 1 db gyermekjegy 3-6 éves korig: 1 900 Ft
 3 db diákjegy: mivel a 16:00 utáni jegyár kedvezőbb, ezért diákjegy helyett olyat kapnak, és mivel 3 azonos kategóriájú jegyre van szükség, ezért csak az első kettőért kell fizetni, a többi (jelen esetben a harmadik) ingyen van, ezért $2 \times 1\,900 \text{ Ft} = 3\,800 \text{ Ft}$
 Összesen: 9 500 Ft
 Családi jegy ára: $9\,500 \text{ Ft} \times 75\% (= 7\,125 \text{ Ft})$, százásokra kerekítve: 7 100 Ft.

0. Nevezze át a projektet úgy, hogy annak neve a neptunkódjával egyezzen meg! Nézze át a Személy osztály és a NotNull annotáció deklarációját is! Előbbiben a cím gyakorlatilag egy települést takar, míg utóbbi egy adattagokra előírható megszorítást definiál, mely szerint az ezzel ellátott adattag értéke nem lehet null referencia.
Készítsen egy ReflectionException nevű nem ellenőrzött kivételt! E kivétel feladata az lesz, hogy mintegy csomagolóként funkcionáljon olyan, a különféle reflection műveletek során bekövetkező kivételek számára, mint a NoSuchMethodException, a SecurityException, az IllegalAccessException, az IllegalArgumentException, vagy éppen az _InvocationTargetException. Épp ezért adjon az elkészített kivételosztálynak olyan konstruktort is, amellyel a hiba kiváltó okát megjelölhetjük!
1. Valósítsa meg a NotNullService interfész metódusait! Az egyes metódusokban bekövetkező, a reflektív hívások eredményeként létrejövő kivételeket úgy kezelje, hogy dobjon egy ReflectionException kivételt, amelyben megjelöli a tényleges kivételokat is. Az osztályt ugyanabban a csomagban hozza létre, amelyben a Személy osztály is van!
- a. `List<Field> getNotNullMegtörtAdattagok(Object o);`
Meghatározza, hogy melyek egy objektum azon adattagjai, amelyek megsértik a NotNull megszorítást és az objektum futásidejű osztályában kerültek deklarálásra.
- b. `void correctNotNullIntekre(Field f, int x, Object o);`
Az Integer típusú f adattag értékét állítsa x-re úgy, hogy meghívja az adott o objektum megfelelő adattagjának a beállító metódusát! Például "életkor" adattag esetén az o objektum `setÉletkor(x)`, "ár" adattag esetén a `setÁr(x)`, stb. metódusának meghívását végzi. (Feltételezzük, hogy a "name" nevű adattaghoz mindig tartozik egy létező void `setName(Integer)` metódus.)
- c. `void checkNotNullMegtörtések(Object o);`
Ellenőrzi, hogy a paraméterül kapott objektumnak van-e deklarált adattagja, amelyre bár elő van írva a NotNull annotáció, az adattag értéke mégis null.
Az ilyenek közül az Integer típusú adattagokhoz az 50 értéket rendelje!
Ha vannak a megszorítást sértő nem Integer adattagok, akkor a metódus az `IllegalStateException` kivételt dobja, amelynek a szövegében szerepeljen az objektum minden olyan adattagjának neve is, amely a megszorítást sérti!
2. Valósítsa meg a SzemélyDAO interfész metódusait úgy, hogy az implementációs osztály konstruktorparamétereként kapott nevű XML-dokumentumból olvassa ki a szükséges adatokat!
- a. `Személy getSzemélyById(String id);`
Visszaadja az adott azonosítójú személyt.
- b. `List<Személy> getÖsszesSzemély();`
Az adatforrás összes személy objektumát visszaadja.
- c. `List<Személy> getÖsszesSzemélyAdottCímről(String cím);`
Visszaadja az adott címen lakó összes személyt.
3. Implementálja az ÉlmenyfürdőService interfészt a fentebb leírt árképzési megoldások tükrében!
- a. `int getJegyar(Személy sz, String belépésiIdő);`
Meghatározza a Személy által fizetendő jegyárát, figyelembe véve a belépés idejét valamint a Személy jellemzőit (címét, életkorát, diák ill. nyugdíjas voltát, hogy azok alapján jogosult-e valamilyen kedvezményre). Egy személyt akkor tekintünk diáknak illetve nyugdíjasnak,

ha a diákigazolványszám ill. nyugdíjasigazolványszám adattagja nem null értékű (feltételezzük, hogy senkinek sem nem null mindkettő egyszerre). A belépés időpontja MI:SS formátumban (perc:másodperc, mindkettő két karakteren, pl. 07:05 vagy 16:02) van megadva.

b. `boolean jogosultakCsaládiJegyre(List<Személy> személyek);`
Meghatározza, hogy a kollekcióban szereplő személyek így együtt jogosultak-e családi jegyre.

c. `int getCsaládiJegyár(List<Személy> személyek);`
Meghatározza a család által fizetendő jegyárát. Feltételezzük, hogy a személyek így együtt jogosultak a családi jegyre (vagyis `jogosultakCsaládiJegyre(személyek) == true`).

4. Készítsen grafikus felhasználói felületet, amelyen két szövegmezőben egy személy azonosítóját és egy belépési időpontot megadva, a „Jegyár számítása” gombra kattintva a felületen jelenjen meg a személy által fizetendő jegyár! (A megvalósításhoz használja a 2. és 3. feladatokban megvalósított alkalmas műveleteket!) A személy adatait a `people.xml` fájlból olvassa! Hiba esetén (pl. adott azonosítóval nincs személy, vagy a belépési időpont nem MI:SS formátumú), a felhasználót egy értelmes hibaüzenettel egy dialógusablakban tájékoztassa!
5. Csomagolja össze a teljes projektjét egy neptunkódjáról elnevezett archívumba!