



Szoftverfejlesztés
gyakorlat – 10. óra

November 8

2018

1. Feladat – Könyv

Készíts egy `Könyv` osztályt, melyből hozz majd létre egy objektum példányt a kedvenc könyveddel.

Az osztályban hozd létre a következő rejtett adattagokat:

- `oldalszam`
- `aktualisOldalszam`
- `nyitva`

Készíts konstruktort a `Könyv` osztályba, ami a következő elv alapján működik:

- A könyvnek létrehozásakor véletlen szám sorsolással határozd meg az oldalszámát [50;250].
- A könyv alaphelyzetben zárva legyen, ezért az aktuális oldalszám legyen 0.

Készítsd el a következő metódusokat, mellyel a könyvet kezelni tudod:

- `kinyit()` – egy véletlen számú oldalon nyissa ki a könyvet, ha még nem volt nyitva, különböző hibaüzenetet írjon ki a program.
- `bezár()` – ha nyitva van, akkor zárja be a könyvet, különben hibaüzenet jelenjen meg.
- `lapoz()` – lehessen egyesével lapozni a könyvet az általad megadott irányba (ha a könyv elejére vagy végére érsz, akkor az utolsó lapozás zárja be a könyvet)
- `porget()` – lehessen lapozni a könyvet véletlenszerű lapszámmal az általad megadott irányba (ilyenkor nem zárható be)
- `nez()` – rá lehessen nézni a könyvre, hogy éppen milyen állapotban van (hol van nyitva, zárva van)

Gondoskodj a megfelelő hibakezelésekről és üzenetekről. A létrehozott `Könyv` példányt teszteld a megfelelő metódusokkal.

2. Nagykövetség és diplomata szimulátor

A feladatban egy ország nagykövetségét és az ott dolgozó diplomatákat fogjuk ábrázolni. A diplomatákról számon tartjuk nevüket és az általuk beszélt idegen nyelveket. A diplomaták neveit és a beszélt nyelveket `String`-ként ábrázoljuk.

A feladatban a `government.Diplomat` és `government.Embassy` osztályokat fogjuk több lépésben megvalósítani. Az előbbi egy nagykövetségen dolgozó diplomatát, az utóbbi egy nagykövetséget ábrázol.

government.Diplomat osztály:

Szerepeljenek benne az alábbi rejtett adattagok.

- Egy `String` típusú név.
- Egy `String` tömb, mely a beszélt nyelveket tárolja.

Az osztályhoz készítsen egy rejtett konstruktort, mely paraméterül várja a két adattag kezdeti értékét, és inicializálja az adattagokat.

Valósítson meg egy osztályszintű **make()** nyilvános metódust, mely ugyanazokat a paramétereket várja, mint a konstruktor, és egy `Diplomat` objektumot ad vissza. A metódus elvégzi az alábbi ellenőrzéseket, majd létrehozza és visszaadja az objektumot, ha minden feltétel teljesül, különben egy null referenciát ad vissza.

- A név nem egy üres szöveg.
- A név pontosan két részből áll, szóközzel elválasztva.
- A két rész egyike sem üres szöveg, mindkettő egy-egy nagy kezdőbetűvel kezdődik. Ennek leellenőrzéséhez használjuk a `Character.toUpperCase()` metódust.
- Figyeljünk arra, hogy a létrehozott objektum belső állapota ne szivároгjon ki.

Készítsen egy **getName()** paraméter nélküli metódust, mely visszaadja a diplomata nevét.

Készíts egy osztályszintű, nyilvános `JAMES_BOND` adattagot, mely egy diplomatát ábrázol, akinek a neve James Bond, és beszéli a következő nyelveket: „English”, „German”, „French”.

Emellett valósítsuk meg az alábbi nyilvános metódusokat is:

- Egy **speaks()** metódust, mely egy nyelvet vár paraméterül, és logikai értékben adja vissza, hogy a diplomata beszéli-e a nyelvet. Feltesszük, hogy a paraméter nem null.
- Egy **getLanguages()** paraméter nélküli metódust, mely visszaadja a beszélt nyelvek tömbjét. Figyeljünk arra, hogy a metódus ne szivároгtassa ki a belső állapotot.
- Egy **toString()** metódust, mely visszaadja a diplomata szöveges ábrázolását a következő alakban: `Diplomat(<név>, <nyelvek>)`. Például `Diplomat(Colin Powell, [English, French])`. A nyelvek formátuma tetszőleges, azonban legalább egy szóközzel legyenek elválasztva.

government.Embassy osztály:

Vegyük fel az alábbi rejtett adattagokat:

- Egy String típusú országnévet, a nagykövetség országát.
- `Diplomat` objektumok sorozatát, mely a nagykövetségen dolgozó diplomatákat tárolja.
- Valósítsuk meg az alábbi metódusokat is a `government.Embassy` osztályban.

Egy konstruktort, mely paraméterül várja a nagykövetség országának nevét és egy `Diplomat` objektumokból álló tömböt, mely utóbbi a nagykövetségen dolgozó diplomatákat tárolja. A konstruktor beállítja az adattagok kezdeti értékét. Ügyeljünk arra, hogy a létrehozott objektum belső állapota ne szivároгjon ki.

Egy **spokenLanguages()** paraméter nélküli metódust, mely visszaadja a nagykövetségen dolgozó diplomaták által beszélt nyelvek listáját, ismétlés nélkül.

Egy **spokenBy()** metódust, mely egy nyelvet vár paraméterül, és visszaadja azon `Diplomat` objektumok listáját, melyek beszélik a nyelvet.

Egy **delegation()** metódust, mely nyelvek tömbjét várja paraméterül, és összeállít egy delegációt egy országba, ahol a hivatalos nyelveket a paraméter tárolja. A delegációt úgy kell összeállítani, hogy minden nyelvet beszélje legalább egy ember. Nem kell törekedni a legkisebb méretű delegáció összeállítására, azonban kerüljük el az ismétléseket. A delegációt nevek listájaként adjuk vissza. Feltesszük, hogy minden nyelvet beszél legalább egy diplomata, és a paraméter legalább egy nyelvet tartalmaz.

3. Feladat – Kávéfőző

A **kávéfőző** a következő metódusokkal rendelkezzen:

- **feltolt(mivel, mennyit)** - A kávéfőzőbe maximum 10 adag vizet, és 6 adag kávé lehet tölteni. Vizet bármikor utánatölthetsz, kávé csak akkor, ha nem használt. A kávéfőzőt fel lehessen tölteni vízzel és kávéval a kapacitásnak megfelelően.

- kiurit(mit) - A kávéfőzőből ki lehessen üríteni a használt kávé vagy vizet.
- foz() - A feltöltött kávéfőzővel lehessen kávé főzni. A főzéskor mindig annyi vizet használ fel, ahány adag kávé van benne. A kávéfőző rendelkezzen egy edénnyel, ami főzéskor a kész kávé tárolja. Főzéskor írd ki, hány adag kávé főztél.

Az **edény** a következő metódusokkal rendelkezzen:

- kostol() – megadja, hogy milyen kávé van benne. Ha főzéskor a víz kevesebb, mint amennyi adag kávé van benne, a lefőtt kávé legyen túl erős. Ha a kávé használt volt, akkor a kávé legyen pocsék.
- kiont() – kiönti a kész kávé

Ha az edényt nem ürítet ki, akkor a benne lévő jó kávé el tudja rontani az esetlegesen “ráfőzött” rossz, vagyis ha nem ürítetted ki a használt kávé, vagy túl kevés vízzel főzted, valamint a kész kávé mennyisége is legyen több. Ha jó kávéra jót főztél, akkor az edényben is jó marad a kávé, csak a mennyisége legyen több.

A kávéfőzőre rá lehessen nézni, ilyenkor írd ki, hogy mivel és mennyire van feltöltve, használt kávé van-e benne, valamint éppen milyen és mennyi kávé van az edényben.

A vezérlőprogramban metódus hívásokkal üzemeltesd a kávéfőzőt.

```
kv.feltolt("hasznalt_kave",5);
kv.feltolt("kave",5);
kv.feltolt("kave",3);
kv.feltolt("viz",10);
kv.feltolt("viz",2);
kv.foz();
kv.foz();
kv.nez();
edeny.kostol();
edeny.kiont();
kv.nez();
```

Megfelelő hibakezeléssel biztosítsd, hogy ne fordulhassanak elő hibák. A hibákról szöveges üzenetet is írnak ki:

- negatív vízmennyiség
- túltöltés (csak akkor tölthess bele, ha belefér)
- üresen főzni
- edényből kétszer kiöntés, stb.