Java naptár

Készítette: Goldschmidt Balázs, BME IIT, 2019.

A feladat célja, hogy elkészítsünk egy naptár-alkalmazást, amely a szabványos bemeneten fogad parancsokat, és a szabványos kimenetre írja ki az eredményt.

1 Aktuális hónap napjainak kiíratása

Készítsünk egyszerű Java alkalmazást, amely kiíratja naptárformában az aktuális hónapot, pl:

```
May 2019
Mo  Tu  We  Th  Fr  Sa  Su
            1   2   3   4   5
 6   7   8   9  10  11  12
13  14  15  16  17  18  19
20  21  22  23  24  25  26
27  28  29  30  31
```

a) Hozzunk létre egy *Main* osztályt a szokásos *main* metódussal (főprogram)!
b) A főprogramban példányosítsunk egy *LocalDate*-et (*LocalDate.now()*), és írassuk ki az értékét a szabványos kimenetre! Legyen ez a változó a *today*.
c) Hozzunk létre egy metódust (*printMonth()*), ami paraméterül egy *LocalDate*-et kap (*date*). Ez a metódus fogja kiírni a fent megadott módon a hónapot.
d) A metódusban írassuk ki a hónap nevét (*getMonth()*) és az évet (*getYear()*), valamint a fent is látható sort, ami a napok neveit mutatja (minden nap között 2 szóköz van). Ezt a függvényt hívjuk meg a *today* változóval mint paraméterrel.

A további feladatpontok a táblázatos formátum előállítását segítik, és a *printMonth* metódusra vonatkoznak.

e) Állítsuk elő az aktuális hónap első napját! Ehhez a *date*-en hívjuk meg a *minusDays* metódust az aktuális dátumnál (*getDayOfMonth()*) eggyel kisebb értékkel. Pl. ha ma 13-a van, akkor a *minusDays* paramétere legyen 12. Az eredményt tároljuk el a *first* nevű, *LocalDate* típusú változóban.
f) Írassuk ki a hónap első napját helyesen indentálva, vagyis a dátumok a napokhoz illeszkedjenek. Ha hónap 1-je mondjuk szerdára esik, akkor a hétfő és a kedd helyén álljon megfelelő számú szóköz. Segítség: a *LocalDate* osztály *getDayOfWeek()* metódusa visszaadja a napot jelképező objektumot, aminek a *getValue()* metódusa visszaadja a nap sorszámát (az 1-es sorszámú nap a hétfő, a 7-es a vasárnap).
g) Írassuk ki a többi napot is, a *first*-höz adva mindig egy napot (*plusDays(1)*), amíg a hónap végére nem érünk. Ügyeljünk arra is, hogy ha egy dátum egyjegyű, akkor a szám előtt álljon egy extra szóköz! A hónap hosszát a *lengthOfMonth()* metódus adja vissza, de figyelhetjük azt is, hogy a dátum átfordul-e 1-esbe.
A heti bontást úgy valósítsuk meg, hogy megvizsgáljuk, az adott nap vasárnap-e, mert ekkor újsor-jelet írhatunk!

2 Egész éves naptár

Írjunk *printYear* metódust, ami egy *LocalDate*-et kap paraméterül (*date*), és a fenti *printMonth* többszöri meghívásával az egész év naptárját írja ki januártól decemberig! Ehhez használhatjuk a *LocalDate*-ben található *minusMonths()*, *getMonthValue()*, *plusMonths()* metódusokat, illetve az egyes hónapok előállításához akár a *LocalDate* osztály *of* metódusát is, pl. a *date* objektummal azonos évben előforduló április 1-jét így is elő lehet állítani:

```
LocalDate apr1 = LocalDate.of(date.getYear(), 4, 1);
```

A *printYear* metódusnak a segítségével írassuk ki az idei év naptárját!

3 Parancssori felület

A további feladatok egy egyszerű naptáralkalmazás elkészítésén vezetnek végig. A naptárban tároljunk eseményeket! Az egyszerűség kedvéért egy eseményt a nevével adunk meg.

Az eseményeket egy *TreeMap*-ben tároljuk, ezzel a típusdefinícióval: *TreeMap<LocalDate, String>.*

A főprogramban legyen ilyen típusú a statikus változónk (*events*).

> *Szorgalmi feladat: egy dátumhoz lehessen több eseményt is rendelni! Ehhez a TreeMap típusát kicsit összetettebbre kell választani: TreeMap<LocalDate, Collection<String>>. Vagyis egy LocalDate-hez egy olyan Collection fog tartozni, amiben String típusú objektumok lehetnek.*
>
> *Érdemes egy külön metódust írni, ami paraméterül megkapja a LocalDate-et és a tárolandó eseményt (pl.* putEvent(LocalDate datum, String esem)*), és a fenti TreeMap-be beteszi őket. Ha a dátum még nem szerepelt, akkor Collection-ként egy ArrayList-et hoz létre,ebbe teszi az eseményt és ezt teszi a TreeMap-be. Ha a dátum már szerepelt, akkor elkéri a hozzá tartozó Collection-t, és abba teszi az eseményt.*

A korábbi laborokhoz hasonlóan készítsünk parancsfeldolgozást, és valósítsuk meg az alábbi parancsokat:

- **`add`** `<dátum> <név>`
  pl: *add 2019-05-31 Koncert*
  Adott időpontra az adott nevű esemény felvétele.
  Tippek: A dátum létrehozásához használjuk a *LocalDate.parse(String)* metódust ("2019-04-05" formátumú dátumokat fogad el).
- **`list`**
  Kilistázza az összes eseményt .
- **`save`** `<filename>`
  Szerializáltan kiírja a *TreeMap* tartalmát a *filename* nevű fájlba.
- **`load`** `<filename>`
  Beolvassa a szerializált *TreeMap* tartalmát a *filename* nevű fájlból. A *TreeMap* korábbi tartalma elveszik.

4 Naptár és események összekapcsolása

a) Módosítsuk úgy a naptárkiíratást, hogy minden dátum után jelenjen meg egy csillag, ha arra a napra esik esemény. Pl.

```
May 2019
Mo   Tu   We   Th   Fr   Sa   Su
                1    2    3    4    5
 6    7    8    9   10   11   12
13   14   15   16   17   18   19
20   21   22   23   24   25   26
27   28   29   30   31*
```

b) Legyen egy **month** nevű parancsunk, ami a megadott hónapot íratja ki a fenti módon, pl *month 5* kiíratja májust. Ha nem adunk meg paramétert, akkor az aktuális hónapot íratja ki.

c) Legyen egy **year** nevű parancsunk, ami az egész évet kiírja. Ha van opcionális év, akkor a megfelelő évet.

5 Adott nap eseményei

Módosítsuk úgy a **list** parancsot, hogy ha van opcionális év, hónap és nap, akkor az ahhoz tartozó eseményt (eseményeket) íratja ki sorban. Meg lehessen hívni csak évvel, ekkor az egész évben levő eseményeket; évvel és hónappal, ekkor az adott hónapban levő eseményeket; évvel, hónappal, nappal, ekkor az erre a napra eső eseményeket.

6 Események bővebben

Az eseményekhez készítsünk objektumorientált modellt, amihez először az alábbi osztályokra lesz szükségünk:

- *Esemeny*: egy általános bejegyzést ír le, ami minden eseményünk őse. A heterogén kollekciónk absztrakt alaposztálya. Van privát neve (*nev*, *String*), időpontja (*ido*, *LocalTime*), és ki lehet íratni (*toString()*). A név és időpont lekérdezhető (*getter metódusok*).
- *Tanora*: az *Esemeny* leszármazottja. Az örökölt attribútumokon kívül Van helyszíne (*terem*, *String*) és tárgya (*tantargy*, *String*). A kiíratáskor (*toString()*) a név mellett ezek is kiíródnak.
- Koncert: az Esemeny leszármazottja. Az örökölt attribútumokon kívül van helyszíne (*cim*, *String*).

Az eseményeket a fenti *TreeMap*-ben tároljuk, de most a második sablonparaméter legyen az Esemeny típus: *TreeMap<LocalDate, Esemeny>*.

7 Parancssori felület módosítása

A korábban készített parancsokat az alábbiak szerint módosítsuk:

- **add** <dátum> <típus>
  pl: *add 2019-05-31 Koncert*
  Adott típusú esemény felvétele. A parancs kiadása után a végrehajtó függvény kérje be a szabványos bemenetről soronként az adott típusú esemény adatait. Pl. Koncert esetén:
  ```
  Kérem, adja meg a Koncert nevét>
  Kérem, adja meg a Koncert időpontját>
  Kérem, adja meg a Koncert helyszínét>
  ```

  Az adatbekérést valósítsuk meg az adott típus egy statikus metódusával (*readFrom*), ami egy *Scanner*-t kap paraméterül, felteszi a szükséges kérdéseket, a válaszok alapján pedig létrehozza és visszaadja az objektumot.

  Tippek: A dátum létrehozásához használjuk a *LocalDate.parse(String)* metódust ("2019-04-05" formátumú dátumokhoz), az időponthoz a LocalTime.parse(String) metódust ("20:05" formátumhoz).
  A típus alapján döntsük el, hogy milyen kérdéseket kell feltenni. Az adatok bekérése után hozzuk csak létre az új eseményt.

- **list**
  Kilistázza az összes eseményt . Ha van opcionális év, hónap és nap, akkor az ahhoz tartozó eseményt (eseményeket) íratja ki sorban. Meg lehessen hívni csak évvel, ekkor az egész évben levő eseményeket; évvel és hónappal , ekkor az adott hónapban levő eseményeket; évvel, hónappal, nappal, ekkor az erre a napra eső eseményeket.
- **save** <filename>
  Szerializáltan kiírja a *TreeMap* tartalmát a *filename* nevű fájlba. Ehhez az *Esemeny* ősosztálynak meg kell valósítania a *Serializable* interfészt.
- **load** <filename>
  Beolvassa a szerializált *TreeMap* tartalmát a *filename* nevű fájlból. A *TreeMap* korábbi tartalma elveszik.