

Programiranje I — 4. domača naloga

Rok za oddajo: nedelja, 2. decembra 2018, ob 23:55

Bankomat

Naloga

V skladu z navodili v nadaljevanju napišite razreda **Datum** in **Bankomat**. Pri vsakem elementu (konstruktorju oz. metodi) so v oglatih oklepajih navedeni testni razredi, v katerih se kliče ta element. Če ste, denimo, zadovoljni s 30% točk, zadošča, da napišete in oddate razred **Datum**.

Po potrebi lahko v razreda dodate še svoje lastne konstruktorje, metode in attribute.

Razred Datum

Objekt razreda **Datum** predstavlja nek datum. Razred naj vsebuje sledeče elemente:

- `public Datum(int dan, int mesec, int leto)` [J1–J10, S1–S50]

Ustvari objekt, ki predstavlja datum s podanim dnevom, mesecem in letom. Parameter `dan` se v vseh testnih primerih nahaja znotraj intervala [1, 31], parameter `mesec` znotraj intervala [1, 12], parameter `leto` pa znotraj intervala [1901, 2099]. Trojica (`dan`, `mesec`, `leto`) v vseh testnih primerih predstavlja veljaven datum.

- `public int vrniDan()` [J1–J10, S1–S50]

Vrne dan datuma `this` (tj. zaporedno številko dneva v mesecu).

- `public int vrniMesec()` [J1–J10, S1–S50]

Vrne mesec datuma `this` (tj. zaporedno številko meseca v letu).

- `public int vrniLeto()` [J1–J10, S1–S50]

Vrne leto datuma `this`.

- `public String toString()` [J2–J10, S6–S50]

Vrne niz oblike `DD.MM.LLLL`, kjer `DD` predstavlja dan, `MM` mesec, `LLLL` pa leto datuma `this`. Če je dan oziroma mesec enomestno število, naj bo zapisan z vodilno ničlo (npr. `06.09.2018` za šesti september 2018).

- `public boolean jeEnakKot(Datum datum)` [J3–J10, S11–S50]

Vrne `true` natanko v primeru, če objekt `this` predstavlja isti datum kot objekt `datum`.

Razred Bankomat

Objekt razreda **Bankomat** predstavlja bankomat, ki sprejema in izdaja bankovce za 5 dinarjev, 2 dinarja in 1 dinar. Razred naj vsebuje sledeče elemente:

- `public Bankomat()` [J4–J10, S16–S50]
Ustvari objekt, ki predstavlja bankomat na začetku delovanja. Bankomat je tedaj prazen.
- `public int vrniN5()` [J4–J10, S16–S50]
Vrne trenutno število bankovcev za 5 dinarjev v bankomatu `this`.
- `public int vrniN2()` [J4–J10, S16–S50]
Vrne trenutno število bankovcev za 2 dinarja v bankomatu `this`.
- `public int vrniN1()` [J4–J10, S16–S50]
Vrne trenutno število bankovcev za 1 dinar v bankomatu `this`.
- `public void nalozi(int k5, int k2, int k1)` [J4–J10, S16–S50]
Posodobi stanje bankomata `this`, potem ko tehnik vanj vstavi `k5` bankovcev po 5 dinarjev, `k2` bankovcev za 2 dinarja in `k1` bankovcev za 1 dinar.
- `public void izpisi()` [J5–J10, S21–S50]
Izpiše niz oblike `n5|n2|n1`, kjer `n5`, `n2` in `n1` predstavljajo število bankovcev po 5 dinarjev, 2 dinarja oziroma 1 dinar, ki jih bankomat `this` trenutno vsebuje. Izpis naj se zaključi s skokom v naslednjo vrstico. Na primer, če bankomat vsebuje 7 bankovcev po 5 dinarjev, 11 bankovcev po 2 dinarja in 6 bankovcev po 1 dinar, naj metoda izpiše
`7 | 11 | 6`
- `public int kolicinaDenarja()` [J5–J10, S21–S50]
Vrne trenutno količino denarja v bankomatu `this`.
- `public boolean dvigni(int dvig, Datum datum)` [J6–J10, S26–S50]
Simulira dogodek, ko stranka poskuša na datum `datum` iz bankomata `this` dvigniti `dvig` dinarjev. Če bankomat zneska ne more izplačati, naj se njegovo stanje ne spremeni, metoda pa naj vrne `false`. V nasprotnem primeru naj pri izplačevanju zelenega zneska bankomat izda karseda veliko 5-dinarskih bankovcev, morebitni preostanek pa naj izplača s karseda velikim številom 2-dinarskih bankovcev.

Datum si v zaporednih klicih metode `dvigni` sledijo v kronološkem vrstnem redu. Nikoli se torej ne zgodi, da bi datum v nekem klicu metode `dvigni` nastopil kasneje kot datum v nekem poznejšem klicu metode `dvigni`.

V primerih J6 in S26–S30 velja `dvig < 5`. V vseh primerih velja `dvig > 0`.
- `public int najDvig()` [J8–J10, S36–S50]
Vrne največjo količino denarja, ki je bila iz bankomata `this` dvignjena v enem samem uspešnem dvigu. Če ni bilo še nobenega uspešnega dviga, naj metoda vrne 0.
- `public Datum najDatum()` [J9–J10, S41–S50]
Vrne datum, ko je bilo iz bankomata `this` dvignjenega največ denarja, pri čemer se vnovič upoštevajo samo uspešni dvigi. Če je takih datumov več, naj metoda vrne prvega (torej najzgodnejšega) med njimi. Če ni bilo še nobenega uspešnega dviga, naj metoda vrne `null`.

Testni primer J9

Testni razred (in pripadajoči izhod):

```
public class Test09 {

    public static void main(String[] args) {
        Datum datum = new Datum(15, 3, 2018);

        System.out.println(datum.vrniDan());           // 15
        System.out.println(datum.vrniMesec());         // 3
        System.out.println(datum.vrniLeto());          // 2018
        System.out.println(datum.toString());          // 15.03.2018
        System.out.println(datum.jeEnakKot(datum));    // true
        System.out.println(datum.jeEnakKot(new Datum(15, 3, 2018))); // true
        System.out.println(datum.jeEnakKot(new Datum(15, 3, 2019))); // false

        System.out.println("-----");
        Bankomat bankomat = new Bankomat();
        bankomat.nalozi(2, 5, 0);
        bankomat.nalozi(6, 2, 1);
        System.out.println(bankomat.vrniN5());         // 8
        System.out.println(bankomat.vrniN2());         // 7
        System.out.println(bankomat.vrniN1());         // 1
        bankomat.izpisi();                             // 8 | 7 | 1
        System.out.println(bankomat.kolicinaDenarja()); // 55

        System.out.println("-----");
        Datum datum2 = new Datum(20, 1, 2019);
        System.out.println(bankomat.dvigni(9, datum)); // true
        bankomat.izpisi();                             // 7 | 5 | 1
        System.out.println(bankomat.dvigni(13, datum)); // true
        bankomat.izpisi();                             // 5 | 4 | 0
        System.out.println(bankomat.dvigni(13, datum2)); // true
        bankomat.izpisi();                             // 4 | 0 | 0
        System.out.println(bankomat.dvigni(17, datum2)); // false
        bankomat.izpisi();                             // 4 | 0 | 0
        System.out.println(bankomat.dvigni(5, datum2)); // true
        bankomat.izpisi();                             // 3 | 0 | 0

        System.out.println("-----");
        bankomat.nalozi(10, 10, 10);
        bankomat.dvigni(12, new Datum(3, 2, 2019));
        bankomat.dvigni(11, new Datum(3, 2, 2019));
        bankomat.dvigni(10, new Datum(4, 2, 2019));
        bankomat.dvigni(8, new Datum(4, 2, 2019));
        bankomat.dvigni(5, new Datum(4, 2, 2019));
        bankomat.izpisi();                             // 5 | 8 | 8
        System.out.println(bankomat.kolicinaDenarja()); // 49

        System.out.println("-----");
        System.out.println(bankomat.najDvig());        // 13
        System.out.println(bankomat.najDatum().toString()); // 03.02.2019
    }
}
```

Oddaja naloge

Oddajte datoteki z nazivoma `Datum.java` in `Bankomat.java`. V prvi vrstici obeh datotek v komentarju navedite svojo vpisno številko. Če je, denimo, vaša vpisna številka enaka 63180999, morata datoteki izgledati takole:

```
// 63180999
```

```
public class Datum {  
    ...  
}
```

```
// 63180999
```

```
public class Bankomat {  
    ...  
}
```

Oddajte **ločeni** datoteki `Datum.java` in `Bankomat.java`. **Ne »zipajte«!**

Testiranje

Program `tj.exe` boste tokrat pognali takole:

```
tj.exe <mapa_z_vasim_razredom> <mapa_s_testnimi_razredi> <mapa_za_rezultate>
```

Če si želite postopek testiranja karseda poenostaviti, postavite datoteki `Datum.java` in `Bankomat.java` v mapo, kjer se nahajajo testni razredi. Znotraj te mape boste namreč lahko program `tj.exe` pognali preprosto takole:

```
tj.exe
```

To je okrajšava za ukaz

```
tj.exe . . .
```

kar pomeni, da se vse, tudi bodoči rezultati, nahaja v trenutni mapi. Če se vaš program nahaja v isti mapi kot testni razredi, boste testne razrede lahko prevajali in poganjali tudi ročno (npr. `javac Test01.java` in `java Test01` za prvi testni razred).