# 6 Klasa LocalDateTime

Predstavljanje datuma i vremena u računaru je relativno složen problem iz nekoliko razloga. Prvo, datumi vreme sadrže veliku količinu podataka (dan, mesec, godina, sat, minut, sekunda, stotinka...) pa se ne mogu predstaviti korišćenjem nekog prostog tipa podatka. Drugo, u svetu postoji više tipova kalendara tj. načina za računanje vremena od kojih neki, na primer, imaju 13 meseci, a ne 12. Konačno, manipulacija podacima, ispis datuma i računanje razlike u vremenu su relativno složene operacije pa bi bilo korisno imati neke već gotove metode koje pružaju ove funkcionalnosti.

Zbog navedenih razloga, u Javi postoji **predefinisana klasa LocalDateTime**. Ona kombinuje informacije o datumu (pomoću unutrašnje klase LocalDate) i vremenu (pomoću unutrašnje klase LocalTime) u jedan objekat koji se može koristiti za manipulisanje vremenom i datumom na različite načine. Klasa LocalDateTime ne poseduje informacije o vremenskoj zoni, ali se za te potrebe može kombinovati sa klasom TimeZone.

Objekti klase LocalDateTime su nepromenljivi. Iz tog razloga svaka „izmena“ na nekom objektu zapravo zahteva kreiranje novog objekta i njegovo smeštanje u postojeću promenljivu.

Prvo pitanje koje se postavlja je: kako saznati trenutni datum i vreme? Klasa LocalDateTime sadrži **statičku metodu „now“** koja vraća novu instancu ove klase sa trenutnim datumom i vremenom preuzetim iz sistema.

```
LocalDateTime datum = LocalDateTime.now();
```

Međutim, pre nego što se uopšte može koristiti ova klasa potrebno je dodati odgovarajuću „import” naredbu pre definicije klase u kojoj se koristi. Razlog je taj što se klasa LocalDateTime nalazi u posebnoj biblioteci „java.time” pa je potrebno naglasiti da se koristi ta biblioteka. Objašnjenje „import” naredbe se može pročitati u poglavlju o paketima i nivoima pristupa.

Najjednostavniji način da se **ispiše datum i vreme je pozivanjem „toString” metode**. Ova metoda vraća String u formatu „yyyy-MM-dd hh:mm:ss.nnnn“.

Za ispis LocalDateTime objekta (datuma i vremena) u drugačijem obliku može se koristiti klasa „DateTimeFormatter“.

***Primer 62***

*Potrebno je kreirati klasu **Datumi** koja ima:*

- *Statičku metodu koja na ekranu ispisuje trenutni datum i vreme.*

*Napraviti klasu **TestDatumi** koja poziva metodu klase Datumi za ispisivanje trenutnog datuma i vremena na ekranu.*

```
import java.time.LocalDateTime;

class Datumi {
      static void ispisiTrenutniDatumIVreme(){
            LocalDateTime datum = LocalDateTime.now();
            System.out.println(datum.toString());
      }
}
class TestDatumi {
```

```
      public static void main(String[] args) {
            Datumi.ispisiTrenutniDatumIVreme();
      }
}
```

*U metodi „ispisiTrenutniDatumIVreme" se pozivom statičke metode „**now**" dobija novi objekat klase LocalDateTime , koji sadrži trenutni datum i vreme. Zatim se poziva metoda „toString" i njen izlaz se ispisuje na ekranu.*

Naravno, objekti klase LocalDateTime mogu da predstavljaju i neki drugi datum i vreme - nečiji datum rođenja, datum prodaje itd. **Ručno postavljanje datuma i vremena se može vršiti na dva načina:**

- Prilikom inicijalizacije, korišćenjem metode „of". Ova statička metoda postoji u nekoliko varijanti. Najčešće se koristi verzija sa 6 parametara (godina, mesec, dan, sat, minut, sekund). Ona vraća novi objekat klase LocalDateTime inicijalizovan u skladu sa prosleđenim vremenskim parametrima.
- Korišćenjem „with" grupe metoda. U ovoj grupi nalaze se metode: „**withYear**", „**withMonth**", „**withDayOfMonth**", „**withHour**", „**withMinute**", „**withSecond**". Kako su objekti klase LocalDateTime nepromenljivi, ove metode takođe vraćaju novi objekat, pa se njihova povratna vrednost mora prihvatiti u promenljivu tipa LocalDateTime kako bi se postigla željena promena.

Bitno je primetiti da nam prvi pristup omogućava da postavimo više elemenata datuma (ili sve) pomoću samo jednog poziva metode „of". To znači da u cilju izmene više paramatera datuma kreiramo samo **jedan objekat** sa svim prosleđenim vrednostima. Da bismo postigli isti rezultat korišćenjem „with" grupemetoda, bilo bi neophodno kreirati nekoliko ulančanih poziva (onoliko koliko parametara želimo da postavimo), **svaki od kojih kreira novi objekat klase LocalDateTime** i time smanjuje efikanost programa.

Iz tog razloga se prvi pristup (metoda „of") preporučuje u situacijama u kojima je neophodno istovremeno **postaviti više elemenata datuma i vremena**.

***Primer 63***

*Dopuniti klasu Datumi tako da ima i:*

- *Statičku metodu koja pravi objekat klase LocalDateTime i postavlja da njegov datum bude: 01.01.2000. Metoda vraća ovaj objekat kao povratnu vrednost. Koristiti „with" grupu metoda.*
- *Statičku metodu koja pravi objekat klase LocalDateTime i postavlja da njegovi datum i vreme budu: 31.12.1999. 23:59:59. Metoda vraća ovaj objekat kao povratnu vrednost. Koristiti „with" grupu metoda.*

```
static LocalDateTime promeniDatumWith(){
      LocalDateTime datum = LocalDateTime.now();
      datum = datum.withYear(2000).withMonth(1).withDayOfMonth(1);
      //Obratiti pažnju da svaka metoda iz „with" grupe metoda kreira novi
      //objekat, tako da se u ovom slučaju zapravo kreira 3 nova objekta,
      //od kojih se poslednji vraća kao povratna vrednost metode.
      return datum;
}
static LocalDateTime promeniDatumIVremeWith(){
      LocalDateTime datum = LocalDateTime.now();
      datum = datum.withYear(1999).withMonth(12).withDayOfMonth(31);
      datum = datum.withHour(23).withMinute(59).withSecond(59);
      return datum;
}
```

*Ove metode je moguće (i preporučuje se) implementirati i korišćenjem pomenute „of" metode klase LocalDateTime:*

```
static LocalDateTime promeniDatumOf(){
      LocalDateTime datum = LocalDateTime.of(2000,1,1,0,0,0);
```

```
      //Jedan poziv metode „of“ kreira samo jedan LocalDateTime objekat.
      return datum;
}

static LocalDateTime promeniDatumIVremeOf(){
      LocalDateTime datum = LocalDateTime.of(1999,12,31,23,59,59);
      return datum;
}
```

Klasa LocalDateTime poseduje i nekoliko metoda **„plus“** grupe. Ove metode omogućavaju jednostavno dodavanje ili oduzimanje određenih vremenskih perioda od objekta tipa LocalDateTime. „Plus“ metode takođe kreiraju nove objekte pa se i njihova povratna vrednost mora prihvatiti u promenljivu tipa LocalDateTime kako bi se postigla željena promena. Metode ove grupe su:

```
plusYears(long years)
plusMonths(long months)
plusWeeks(long weeks)
plusDays(long days)
plusHours(long hours)
plusMinutes(long minutes)
plusSeconds(long seconds)
```

***Primer 64***

*Dopuniti klasu Datumi tako da ima i:*

- *Statičku metodu koja kao parametar prima objekat klase LocalDateTime. Ova metoda vraća datum i vreme koje je tačno 3 nedelje **nakon** prosleđenog datuma i vremena.*
- *Statičku metodu koja kao parametar prima objekat klase LocalDateTime. Ova metoda vraća datum i vreme koje je tačno 2 sata i 15 minuta **pre** prosleđenog datuma i vremena.*

```
static LocalDateTime dodajNedelje(LocalDateTime datum){
      datum = datum.plusWeeks(3);
      return datum;
}

static LocalDateTime vratiVreme(LocalDateTime datum){
      datum = datum.plusHours(-2).plusMinutes(-15);
      return datum;
}
```

U nekim situacijama je potrebno preuzeti samo neke elemente datuma, ali ne sve. U tu svrhu se koriste naredne metode koje vraćaju elemente datuma u formi celog broja:

```
getYear()
getMonthValue()
getDayOfMonth()
getHour()
getMinute()
getSecond()
```

Često je potrebno uporediti dva datuma i utvrditi koji od njih se odnosi na raniji trenutak. Klasa LocalDateTime ima dve predefinisane metode koje se koriste u ovu svrhu - **„isBefore”** i **„isAfter”**. Metoda „isBefore” poredi dva datuma i vraća „true” ako se prvi datum odnosi na raniji trenutak, a „false” ako to nije slučaj. Metoda „isAfter” radi upravo suprotnu stvar.

***Primer 65***

*Dopuniti klasu Datumi tako da ima i:*

- *Statičku metodu koja kao parametar dobija objekat klase LocalDateTime i na ekranu ispisuje godinu, mesec, dan, sat, minut i sekundu u formatu „DD.MM.GGGG SS:MI:SE".*
- *Statičku metodu koja kao parametre prima datume rođenja dve osobe i na ekranu ispisuje koja osoba je starija.*

```
static void ispisiPosebno(LocalDateTime datum){
      int godina = datum.getYear();
      int mesec = datum.getMonthValue();
      int dan = datum.getDayOfMonth();
      int sat = datum.getHour();
      int minute = datum.getMinute();
      int sekunde = datum.getSecond();

      System.out.println(dan+"."+mesec+"."+godina+
                  ". "+sat+":"+minute+":"+sekunde);
}

static void starijaOsoba (LocalDateTime datRodj1,
                                    LocalDateTime datRodj2){
      if (datRodj1.isBefore(datRodj2))
            System.out.println("Starija je prva osoba");
      else
            System.out.println("Starija je druga osoba");
}
```

## *Zadaci*

### *Zadatak 1*

*Napraviti klasu **IstorijskiDogadjaj**. Ova klasa bi trebalo da ima:*

- *Atribut nazivDogadjaja koji je tipa String.*
- *Atribut datumDogadjaja koji je tipa LocalDateTime.*
- *Konstruktor koji prima četiri parametra - String i tri cela broja. Prvi parametar predstavlja naziv događaja dok naredna tri predstavljaju godinu, mesec i dan kada se događaj odigrao. Ovaj konstruktor postavlja vrednosti odgovarajućih atributa klase.*
- *Metodu koja na ekranu ispisuje u kom godišnjem dobu se desio događaj (proleće 22.3.-21.6., leto 22.6-22.9, jesen 23.9. - 21-12. i zima 22.12.-21.3.).*
- *Metodu koja kao parametar prima objekat klase IstorijskiDogadjaj i na ekranu ispisuje kojidogađaj se desio pre.*
- *Metodu koja na ekranu ispisuje naziv događaja i datum kada se desio u formatu „Dana DD:MM:GGGG se desio dogadjaj: NAZIV_DOGADJAJA"*

*Napraviti klasu **TestIstorijskiDogadjaj** koja kreira dva objekta klase IstorijskiDogadjaj: „Bombardovanje Beograda u II svetskom ratu" 6.4.1941 i „NATO Bombardovanje Beograda", 24.3.1999. Potrebno je ispisati na ekranu godišnje doba u kojem su se odvila oba događaja kao i to koji događaj se desio pre.*

### *Rešenje*

```
import java.time.LocalDateTime;

class IstorijskiDogadjaj {

      String nazivDogadjaja;
      LocalDateTime datumDogadjaja;

      IstorijskiDogadjaj(String naziv, int godina, int mesec, int dan){
```

```
            nazivDogadjaja = naziv;
            datumDogadjaja = LocalDateTime.of(godina, mesec, dan, 0, 0, 0);
      }

      void ispisiGodisnjeDoba(){
            int mesec = datumDogadjaja.getMonthValue();
            int dan = datumDogadjaja.getDayOfMonth();

            if (mesec == 4 || mesec == 5 ||
                        (mesec == 6 && dan < 22)||
                        (mesec == 3 && dan >= 22))
                  System.out.println("Prolece");
            if (mesec == 7 || mesec == 8 ||
                        (mesec == 6 && dan >= 22)||
                        (mesec == 9 && dan <= 22))
                  System.out.println("Leto");
             if (mesec == 10 || mesec == 11 ||
                        (mesec == 12 && dan < 22)||
                        (mesec == 9 && dan > 22))
                  System.out.println("Jesen");
            if (mesec == 1 || mesec == 2 ||
                        (mesec == 12 && dan >= 22)||
                        (mesec == 3 && dan < 22))
                  System.out.println("Zima");
      }

      void preDogadjaja(IstorijskiDogadjaj id2){
            if(datumDogadjaja.isBefore(id2.datumDogadjaja))
                  System.out.println("Desio se pre unetog dogadjaja");
            else  System.out.println("Desio se posle unetog dogadjaja");
      }

      void ispisi(){
            int godina = datumDogadjaja.getYear();
            int mesec = datumDogadjaja.getMonthValue();
            int dan = datumDogadjaja.getDayOfMonth();

            System.out.println("Dana "+dan+"."+mesec+"."+godina+
                        " se desio dogadjaj: "+nazivDogadjaja);
      }
}

class TestIstorijskiDogadjaj {

      public static void main(String[] args) {
            IstorijskiDogadjaj id1 =
                  new IstorijskiDogadjaj(
                              "Bombardovanje Beograda u II svetskom ratu",
                              1941,4,6);
            IstorijskiDogadjaj id2 =
                  new IstorijskiDogadjaj("NATO Bombardovanje Beograda",
                              1999,3,24);

            id1.ispisiGodisnjeDoba();
            id2.ispisiGodisnjeDoba();
            id1.preDogadjaja(id2);
      }
}
```