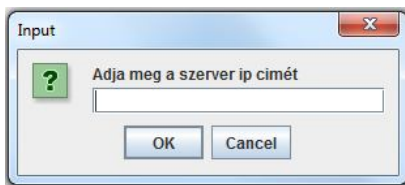


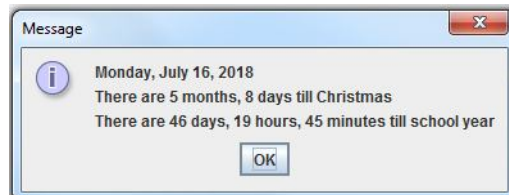
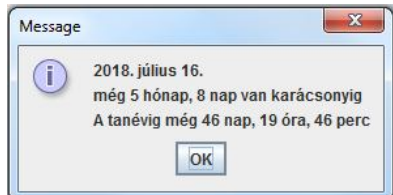
Dátumkezelő kliens-szerver alkalmazás



Összeköttetés alapú szolgáltatást alapul véve oldjuk meg, hogy a világ „bármely” táján meg lehessen kérdezni néhány, dátummal kapcsolatos információt.



A kliens kérje be a fix port-számon futó szerver IP-címét, a szerver pedig válaszként különböző (a kliens által megadott) nyelveken közölje az aktuális dátumot, azt, hogy hány hónap és nap van még hátra karácsonyig, illetve hány nap, óra, perc a tanév kezdetéig.



Használja a `JOptionPane` osztály `showInputDialog()` és `showMessageDialog()` metódusát, illetve a `JDK_8 time` csomagjának szolgáltatásait.

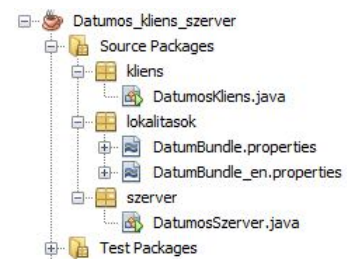
Mielőtt nagyon megfélekedzne róla, írjon JUnit tesztet az idők kiszámításának tesztelésére.

Megoldás-részletek

A dátum részét a `JDK 8 time` csomagjával oldjuk meg, de most megemlítem azt is, hogy korábbi verzióval hogyan lehetne nyelvtől függően megadni a mai dátumot, például így:

```
Locale locale = new Locale(nyelv);  
  
DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.FULL, locale);  
  
String maiKiirtDatum = dateFormat.format(new Date());
```

Ezek után lássuk a konkrét feladat megoldását. Nyilván az lenne logikus, ha külön projektben lenne a kliens és egy másikban a szerver, illetve a lokalitás kezeléséhez szükséges properties fájlok. Most azonban annyira egyszerű a feladat, hogy egyetlen projektbe raktuk őket. Mivel emiatt a projektben most két `main()` metódus is van, ezért célszerű úgy indítani, hogy előbb a `DatumosSzerver` fájlban jobb egérgomb, Run file, majd ugyanezt a `DatumosKliens` fájlban is.



Ha átnézte, hogy hogyan lehet kialakítani az összeköttetés alapú kapcsolatot, és mi a lényege egy kliens-szerver alkalmazásnak, akkor ez a megoldás nem is igényel sok magyarázatot. Most fixen beépítettük a kommunikációt biztosító portot.

A szerver osztály:

```
public class DatumosSzerver {
    private static final int PORT = 44444;
    private static boolean online = true;

    public static void main(String[] args) {
        try {
            new DatumosSzerver().start();
        } catch (IOException ex) {
            Logger.getLogger(DatumosSzerver.class.getName()).log(Level.SEVERE, null,
                ex);
        }
    }

    private void start() throws IOException {
        // A szerver az adott porton várakozik.
        try (ServerSocket serverSocket = new ServerSocket(PORT)) {
            System.out.println("Elindult a szerver");

            LocalDate karacsony = LocalDate.of(2018, 12, 24);
            LocalDateTime tanevkezdodoPont
                = LocalDateTime.of(2018, 9, 1, 8, 00);

            while (online) {
                LocalDate ma = LocalDate.now();

                // fogadja a kliens kapcsolódási kérelmét
                try (Socket socket = serverSocket.accept();
                    BufferedReader input
                        = new BufferedReader(new InputStreamReader(socket.getInputStream()));
                    // Beállítjuk az autoFlush-t is, így nem nekünk kell gondoskodni
                    // arról, hogy az adatok ténylegesen átmenjenek a csatornán.
                    PrintWriter out
                        = new PrintWriter(socket.getOutputStream(), true)) {

                    // Bekéri az aktuális nyelvet.
                    String nyelv = input.readLine();
                    // Beállítjuk a lokalitást
                    Locale locale = new Locale(nyelv);

                    // Meghatározzuk a dátumstring formátumát
                    DateTimeFormatter dtf
                        = DateTimeFormatter.ofLocalizedDate(FormatStyle.FULL);
```

```

dtf = dtf.withLocale(locale);

String maiKiirtDatum = ma.format(dtf);

// A szerver elküldi a mai dátum string alakját
out.println(maiKiirtDatum);

// Megadjuk, hogy hol találja meg a nyelvfüggő szövegeket,
// és azt is, hogy most konkrétan melyik nyelvet akarjuk
// használni.
ResourceBundle bundle
    = ResourceBundle.getBundle("lokalitasok/DatumBundle",
                                locale);

// Meghatározzuk a karácsonyig hátralévő időt.
Period karacsonyig = Period.between(ma, karacsony);

// Mivel a kiírandó szöveg nyelv-függő, ezért azt a megadott
// bundle kulcs alapján találja meg.
String karacsonyigString
    = String.format(bundle.getString("karacsonyig"),
                    karacsonyig.getMonths(),
                    karacsonyig.getDays());
out.println(karacsonyigString);

LocalDateTime mostaniDatumPerc = LocalDateTime.now();

// A temporális adat segítségével számoljuk ki a napok,
// órák, percek értékét.
LocalDateTime tempDateTime
    = LocalDateTime.from(mostaniDatumPerc);

long napok
    = tempDateTime.until(tanevkezdodoPont, ChronoUnit.DAYS);
tempDateTime = tempDateTime.plusDays(napok);

long orak
    = tempDateTime.until(tanevkezdodoPont, ChronoUnit.HOURS);
tempDateTime = tempDateTime.plusHours(orak);

long percek
    = tempDateTime.until(tanevkezdodoPont, ChronoUnit.MINUTES);

String tanevigString
    = String.format(bundle.getString("tanevig"),
                    napok, orak, percek);

out.println(tanevigString);
    }
}
}
}
}

```

A *DatumBundle* *properties* fájl tartalma:

```
karacsonyig=még %d hónap, %d nap van karácsonyig  
tanevig=A tanévig még %d nap, %d óra, %d perc
```

A *DatumBundle_nyelv.properties* fájl kulcsai változatlanok, értékeit értelemszerűen az aktuális nyelvnek megfelelően kell megadni.

A kliens osztály:

```
public class DatumosKliens {  
  
    public static void main(String[] args) {  
        try {  
            new DatumosKliens().start();  
        } catch (IOException ex) {  
            Logger.getLogger(DatumosKliens.class.getName()).log(Level.SEVERE, null, ex);  
        }  
    }  
  
    private void start() throws IOException {  
        // String nyelv = "hu";  
        String nyelv = "en";  
        String szerverCim = JOptionPane.showInputDialog("Adja meg a szerver ip címét");  
        int port = 44444;  
  
        try (Socket socket = new Socket(szerverCim, port);  
            BufferedReader input  
                = new BufferedReader(new InputStreamReader(socket.getInputStream()));  
            PrintWriter out  
                = new PrintWriter(socket.getOutputStream(), true)) {  
  
            out.println(nyelv);  
            String datumString = input.readLine();  
            String karacsonyigString = input.readLine();  
            String tanevigString = input.readLine();  
            JOptionPane.showMessageDialog(null, datumString  
                + "\n" + karacsonyigString  
                + "\n" + tanevigString);  
        }  
    }  
}
```

Megjegyzés: Kliens oldalon is lehetne bundle-t írni az ip-cím bekérésére vonatkozó szöveghez, de ettől most eltekintünk.