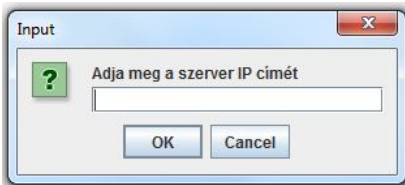


A dátumkezelő kliens-szerver alkalmazás megoldása

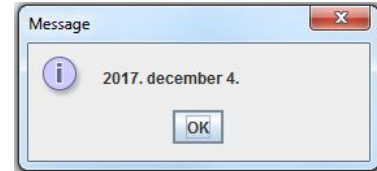
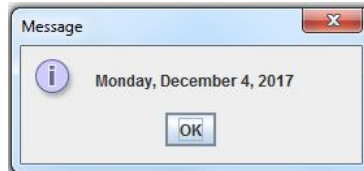
Feladat

Összeköttetés alapú szolgáltatást alapul véve oldja meg a következőket:



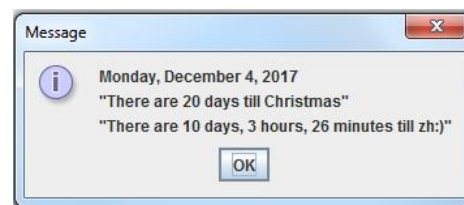
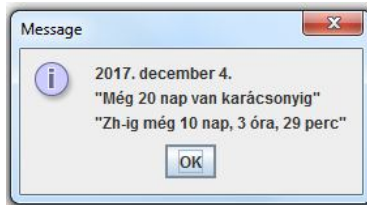
A kliens kérje be a fix port-számon futó szerver IP-címét, a szerver pedig válaszként

a) közölje az aktuális dátumot:



b) most magyarul:

c) majd ezeket is adja meg a választott nyelven:



A megoldáshoz használja a `JOptionPane` osztály `showInputDialog()`, illetve `showMessageDialog()` metódusát, illetve a `JDK_8` `time` csomagjának szolgáltatásait.

Mielőtt nagyon megfélemlene rőla, írjon JUnit tesztet az idők kiszámításának tesztelésére.

Megoldás-részletek

Nem szándékszem különösebb magyarázatot fűzni hozzá, szerintem most elég beszédes a kommentezett kód.

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());
```

A szerver osztály:

```

public class DatumSzerver {

    public static final int port = 44444;
    private static boolean online = true;

    public static void main(String[] args) {

        try {
            new DatumSzerver().start();
        } catch (IOException ex) {
            Logger.getLogger(DatumSzerver.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    private void start() throws IOException {

        String maiKiirtDatum;
        DateTimeFormatter dtf;

        LocalDate ma = LocalDate.now();
        LocalDate karacsony = LocalDate.of(2017, 12, 24);

        LocalDateTime mostanDatumiPerc;
        LocalDateTime zhIdoPont = LocalDateTime.of(2017, 12, 14, 12, 55);
        // Ez a mostani és a zh közötti idő kiszámításához kell
        LocalDateTime tempDateTime;

        Period karacsonyig;

        String karacsonyigString;
        String zhigString;

        // Ez a más nyelv használatához kell.
        ResourceBundle bundle;

        // A szerver az adott porton vár.
        try (ServerSocket serverSocket = new ServerSocket(port)) {
            while (online) {
                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!!
                    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 formázott dátum-stringet.
dtf = DateTimeFormatter.ofLocalizedDate(FormatStyle.FULL);
dtf = dtf.withLocale(locale);
maiKiirtDatum = ma.format(dtf);
// elküldi a szerver
out.println(maiKiirtDatum);

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

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

// karacsonyigString =
//      String.format("Még %d nap van karácsonyig",
//                    karacsonyig.getDays());
// Ezt helyettesítjük a bundle-ben megadott stringgel, amit
// a megadott kulcs alapján talál meg.
karacsonyigString =
    String.format(bundle.getString("karacsonyig"),
                  karacsonyig.getDays());
out.println(karacsonyigString);

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

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

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

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

// zhigString =
//      String.format("Zh-ig még %d nap, %d óra, %d perc",
//                    napok, orak, percek);
zhigString = String.format(bundle.getString("zhig"),
                           napok, orak, percek);
out.println(zhigString);
    }
}
}
}
}

```

A bundle beállításához ld. az ide vonatkozó előadás-anyagot, vagy pl. ezt az oldalt:
<https://netbeans.org/kb/73/java/gui-automatic-i18n.html>

A kliens osztály:

```
public class DatumKliens {

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

    private void start() throws Exception {
        int port = 44444;
        String nyelv = "en";
        // String nyelv = "hu";
        String szerverCim = JOptionPane
            .showInputDialog("Adja meg a szerver IP címét\n");

        // A kliens kapcsolódni próbál a szerverhez.
        try (Socket socket = new Socket(szerverCim, port);
            BufferedReader input = new BufferedReader(new InputStreamReader(
                socket.getInputStream()));
            //Beállítjuk az autoFlush-t is!!
            PrintWriter out = new PrintWriter(socket.getOutputStream(),
                true)) {

            // Elküldi a nyelvet.
            out.println(nyelv);
            // Beolvassa a dátumstring-et
            String datumString = input.readLine();
            // A karácsonyig tartó időszak stringét
            String karacsonyigString = input.readLine();
            // A zhig tartó időszak stringjét.
            String zhigString = input.readLine();
            // Kiírja a kapott értékeket.
            JOptionPane.showMessageDialog(null, datumString +
                "\n" + karacsonyigString +
                "\n" + zhigString);
        }
    }
}
```

A DatumBundle properties fájl tartalma:

```
karacsonyig=még %d nap van karácsonyig
zhig=Zh-ig még %d nap, %d óra, %d perc
```

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