

TUDNIVALÓK:

Most is és a következő gyakorlatokon is – akkor is, ha külön nem emeljük ki – az órán meg nem oldott feladatok HÁZI FELADAT-ként megoldandók!!!

Ez fontos a tárgy sikeres teljesítéséhez!



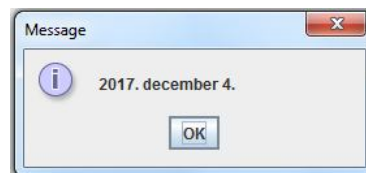
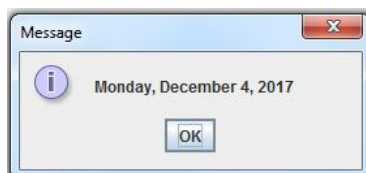
1. feladat:

Összeköttetés alapú szolgáltatást alapul véve oldja meg a következőket:



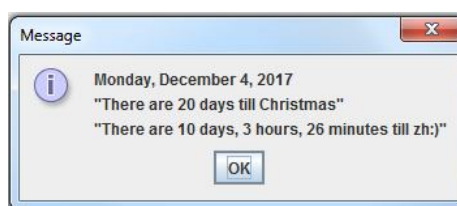
A kliens kérje be a fix port-számon futó szerver IP-címét, a szerver pedig válaszként

a) közölje az aktuális dátumot:



b) most magyarul:

c) majd ezeket is adja meg a választott nyelven:



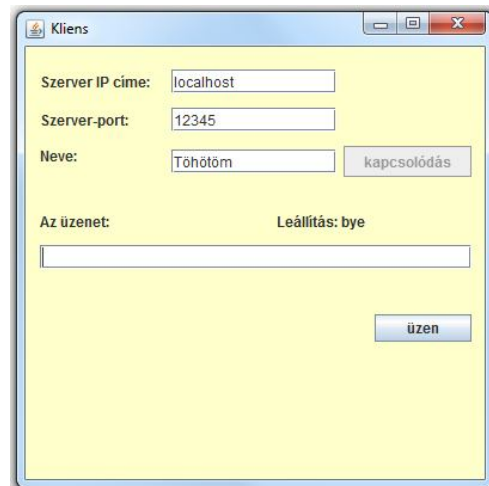
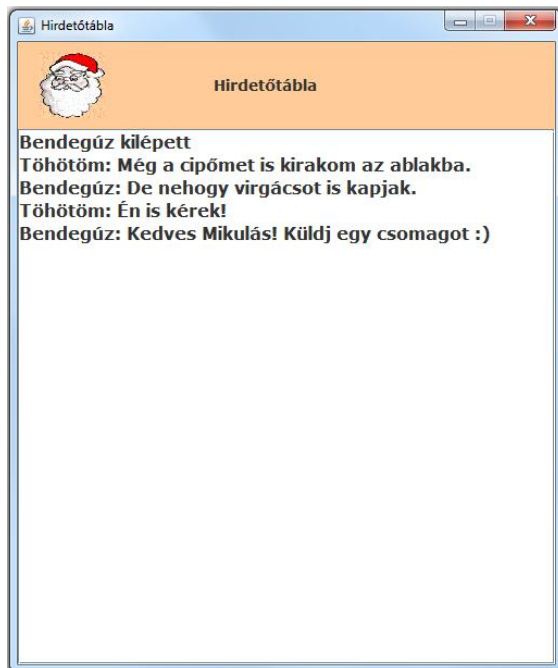
A megoldáshoz használja a `JOptionPane` osztály `showInputDialog()`, illetve `showMessageDialog()` metódusát, illetve a `JDK_8` time csomagjának szolgáltatásait.

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

<http://javarevisited.blogspot.hu/2015/03/20-examples-of-date-and-time-api-from-Java8.html>

2. feladat

A Mikulás a szerver oldalon várja a gyerekek jelentkezését. Írja meg azt a klienst, amelyről be tudunk jelentkezni hozzá, és üzenetet is tudunk hagyni az üzenőfalán.



A gombok értelemszerűen legyenek aktívak/inaktívak, az üzenetet pedig az enter lenyomásával is el lehessen küldeni.

Tudnivalók: Előbb meg kell adni a szerver IP-címét, a port most fixen 12345.

Ugyancsak adja meg a nevét.

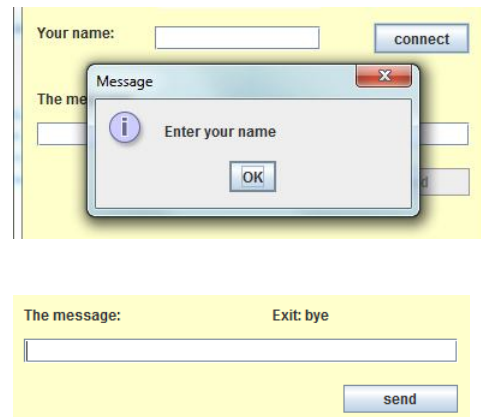
Ezek után a kapcsolódás gomb segítségével kapcsolódjon a szerverhez.

A kapcsolódás gomb hatására jöjjön létre a szerverrel való kapcsolat, ezután a gomb legyen inaktív. A kapcsolat létrejöttkor a szerver fogadja a megadott nevet, majd visszaküldi azt a szót, amelyet megüzenve a szervernek, lezárul a kliens és a szerver kapcsolata. Ez a kapcsolat akkor is záródjon le, ha az egész alkalmazást zárjuk be. A szervertől kapott szó jelenjen is meg a kliens felületén.

Az üzen gomb hatására, illetve az Enter begépelésekor lehet üzenetet küldeni.

A kliensnek az a dolga, hogy egyetlen String-et átküldjön, a szerver majd tudja kezelni azt.

b) Oldja meg a lokalizációt. (Természetesen a többi üzenetet is.)



Bár kódból is megoldható, de egyszerűbb, ha most csak a VM-nek mondjuk meg a nyelvet. Ennek módja: projektnév, jobb egérgomb, Properties; itt Run és a VM Options mezőbe beírni ezt: `-Duser.language=en` (illetve értelemszerűen az adott nyelvre)

Ha esetleg az országra is szükség van, akkor ezt az opciót kell megadni: `-Duser.language=en -Duser.country=GB`

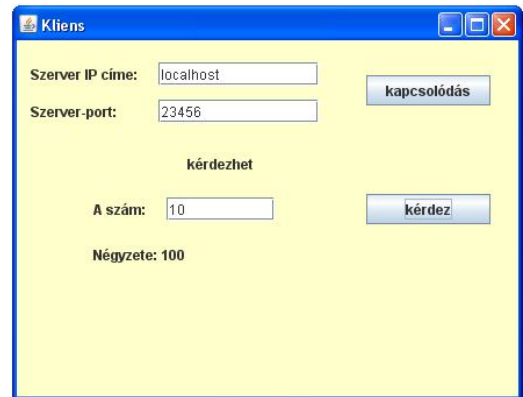
c) Írja meg a szerver oldalt is. Ez már kicsit több utánagondolást igényel, ezért amelyik részre órán nem jut idő, azt szerintem elég, ha csak a zh után próbálja megoldani.

3. feladat

Írjon egy egyszerű kliens-szerver alkalmazást. A kliens grafikus felületű legyen, melyen az alábbi grafikus komponensek szerepeljenek:

- beviteli mező a szerver IP címének a részére
- beviteli mező a szerver port részére
- beviteli mező a szám részére
- címke az üzenetek megjelenítéséhez
- nyomógomb: kapcsolódás és üzenet elküldése a szervernek

A szerver olvassa be a klientsől kapott (egész) számot, és küldje vissza a négyzetét.



Először próbálja úgy megoldani, hogy a szerver csak egyetlen klients tud kiszolgálni, utána úgy, hogy egyszerre többet is. A szervert természetesen nem kell grafikus felületűre írni.

Kliens oldalon figyeljen arra is, hogy ne lehessen kétszer kapcsolódni, ill. hogy jelezze, ha hibás számot (vagy nem számot) írunk be.

4. feladat

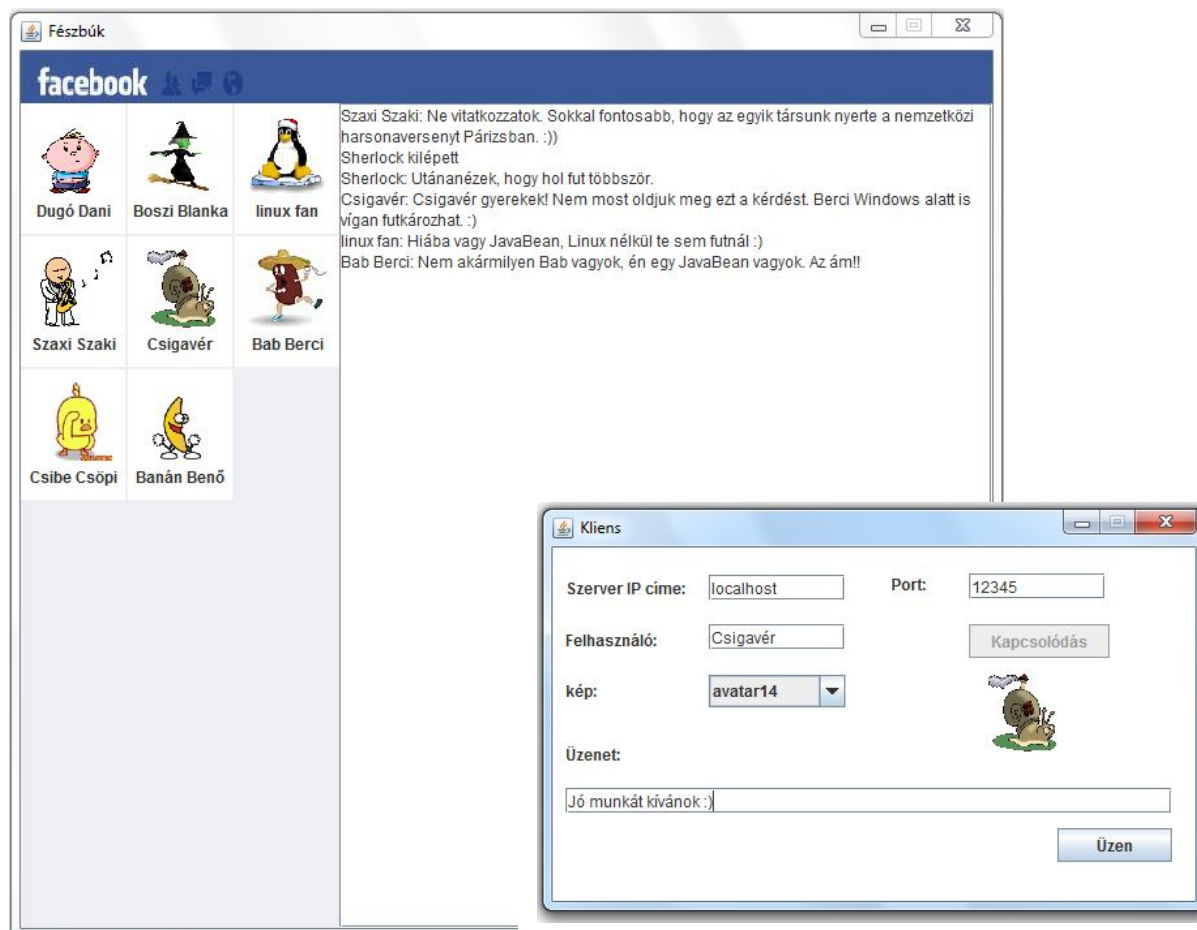
Írjon chat programot!

(De ez is ráér a zh után. ☺)

5. feladat (ez is ráér ☺)

Saját fészbuk készítése ☺:

A szerver lesz a „fészbuk felület”, erre lehet bejelentkezni és üzeneteket küldeni. Az üzenetküldő megadja a becenevét és átküld egy avatárt is. A 750×700-as felület 250 széles baloldali részén látható a bejelentkezettek avatárja, a jobboldalon pedig az üzenetek. Ha valaki kijelentkezik, akkor eltűnik az avatárja, az üzenetek között pedig jelezzük, hogy az illető kilépett.



A kliens-oldal egyszerűbb: egy 500×300-as felületről kapcsolódhat a szerverhez, mégpedig úgy, hogy kapcsolódás előtt kiválaszt egy avatárt, és megadja a becenevét, kapcsolódás után pedig üzengethet.

Segítség a 3. feladathoz:

Értelemszerűen tördelje részekre az adott kliensvázlatot, vagyis:

- A kapcsolódás-gombra kattintva hozza létre a kapcsolatot és az input/output csatornákat. Mivel a szerver első lépésként (és a kapcsolat során csak egyszer) átküldi a „kérdezhet” üzenetet, ezért az üzenetet is itt kell fogadni, és kiírni a megfelelő label-re.
- A kérdez- gombra kattintva lehet kommunikálni a szerverrel, vagyis a szövegmezőben megadott számot itt kell továbbküldeni a szervernek, és a kapott választ itt kell kiírni a megfelelő label-re.

Mivel a protokoll szerint a szerver még vár egy folytatásra vonatkozó üzenetet, ezért még ezt az üzenetet is el kell küldeni.

- Még egy dologra oda kell figyelni: a szerverszál akkor fejezi be a munkáját, ha a folytatásra vonatkozóan nemleges üzenetet küldünk. De ezt nem küldhetjük a kérdez- gomb hatására, hiszen ott még nem tudjuk, hogy mikor akarjuk befejezni a kommunikációt. Ezt az alkalmazás lezárásakor kell megtennünk. Ugyanekkor kell lezárnunk a csatornákat is és a socketet is. Vagyis a frame konstruktorában meg kell adnunk az ablak lezárásához tartozó eseményt:

```
this.addWindowListener(new WindowAdapter() {  
    @Override  
    public void windowClosing(WindowEvent e) { ... }  
})
```

Az esemény megadásakor figyeljen arra is, hogy mikor zárjuk be az ablakot: a kapcsolódás létrejötte előtt vagy után. (Nyilván csak akkor, ha egyáltalán létrejött a kapcsolat.)