

Ćwiczenie 7.

Niniejsze ćwiczenie służy poznaniu technik programowania rozproszonego, opartych na RMI.

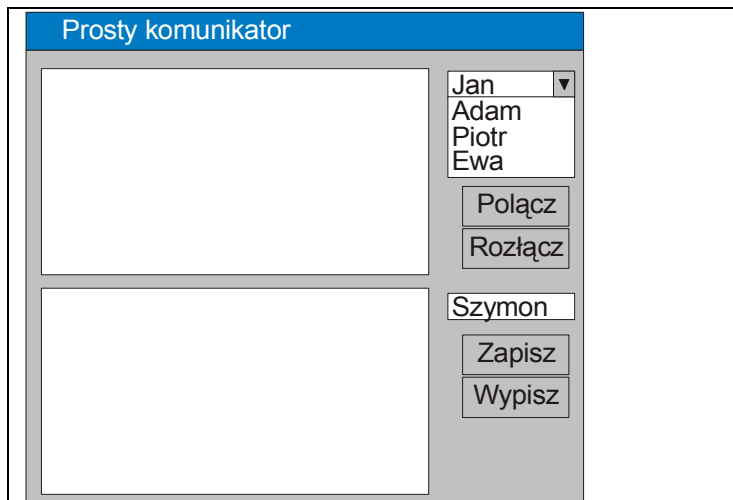
Zad. 1

Napisz program komunikatora, którego działanie podobne będzie do działania programu talk (UNIX). Dodatkowo uruchom serwis zapisów (czyli napisz program pomocniczy), którego zadaniem będzie rejestracja i wyrejestrowywanie namiastek potencjalnych uczestników dialogu. Sam serwis powinien być zarejestrowany w rejestrze RMI, aby wszyscy użytkownicy uruchamiający swoje komunikatory mogli z niego skorzystać w sposób zdalny.

Program komunikatora powinien:

- wyświetlać dwa pola tekstowe: jedno dla zdalnego odbiorcy, jedno dla lokalnego nadawcy. Tekst pisany w polu tekstowym nadawcy powinien pojawiać się w polu tekstowym odbiorcy.
- wyświetlać w rozwijalnej liście wyboru wszystkich zarejestrowanych użytkowników komunikatorów (lista ta powinna być dostarczona przez serwis zapisów),
- pozwaląć na nawiązanie połączenia z wybranym użytkownikiem (tj. odbiorcą) i na zamknięcie takiego połączenia (przyciski Połącz i Rozłącz),
- pozwaląć na zarejestrowanie i wyrejestrowanie użytkownika komunikatora pod wybraną nazwą (przyciski Zapisz i Wypisz).

Propozycja interfejsu graficznego komunikatora może być następująca:



Implementacja serwisu zapisów i komunikatora powinna korzystać z następującej specyfikacji interfejsów zdalnych:

serwis zapisów	komunikator
<pre>import java.rmi.Remote; import java.rmi.RemoteException; public interface SInterface extends Remote { int zapisz (KInterface u, String n) throws RemoteException; int wypisz(String n) throws RemoteException; String[] listuj() throws RemoteException; KInterface pobierz(String n) throws RemoteException; }</pre>	<pre>import java.rmi.Remote; import java.rmi.RemoteException; public interface KInterface extends Remote { void pisz (String s) throws RemoteException; }</pre>

Objaśnienia:

int zapisz (KInterface u, String n) throws RemoteException;
służy do rejestrowania namiastki u pod nazwą n w serwisie zapisów
jeśli operacja powiodła się to zwraca 1, w przeciwnym przypadku zwraca 0 (gdy np. w serwisie zarejestrowana jest już jakaś namiastka z nazwą n).

int wypisz(String n) throws RemoteException;
służy do wyrejestrowania namiastki zarejestrowanej pod nazwą n z serwisu zapisów
jeśli operacja powiodła się to zwraca 1, w przeciwnym przypadku zwraca 0 (gdy np. w serwisie nie ma namiastki zarejestrowanej pod nazwą n).

String[] listuj() throws RemoteException;
służy do pobrania z serwisu tablicy nazw wszystkich zarejestrowanych namiastek.

KInterface pobierz(String n) throws RemoteException;
służy do pobrania z serwisu namiastki zarejestrowanej pod nazwą n.

void pisz (String s) throws RemoteException;
służy do przesłania tekstu s do komunikatora odbiorcy.

UWAGA: `pobierz()` może zwrócić namiastkę, którą za chwilę miał właśnie wyrejestrować jej właściciel (albo może zwrócić błędną wartość, jeśli podana zostanie w argumencie nazwa niezarejestrowanej namiastki).