

Zajęcia nr 2: Java RMI ciąg dalszy.

Przydatne materiały i oprogramowanie:

- Kompilator i maszyna wirtualna języka Java w wersji 1.5 lub wyższej
- Specyfikacje i pomoc dotycząca technologii Java dostępna pod adresem: <http://download.java.net/jdk7/docs/>
- Dowolny analizator ruchu sieciowego (może być *wireshark*)
- Dowolny serwer WWW (może być *apache* albo *tomcat*)

Uwagi:

- Wśród zadań przykładowych będą pojawiały się zadania oznaczone (*). Jest to informacja o tym, że zadanie takie jest trudniejsze, czy to ze względu na tematykę, czy też pracochłonność rozwiązania. Realizacja takich zadań powinna być punktowana dodatkowo.
- Studenci mogą korzystać ze zintegrowanych środowisk programistycznych (IDE) w celu wspomaganie programowania. Nie wolno jednak korzystać z kreatorów dostępnych w takich środowiskach - proces tworzenia aplikacji powinien być w całości kontrolowany (i rozumiany) przez studenta.

Przykładowy zestaw zadań:

1. Zaimplementuj grę "kółko i krzyżyk" wykorzystując implementację Java RMI oraz następujące wskazówki:
 - Implementacja składa się z obiektów dwóch rodzajów - KKSerwer (obiekt zdalny zarządzający grą) oraz KKKlient (obiekt gracza wywołujący metody z KKSerwer).
 - Nie zwracamy uwagi na wizualną stronę gry - interesują nas zdalne wywołania (nie trać czasu na pisanie niepotrzebnego kodu)
2. W napisanej grze "kółko i krzyżyk" zabezpiecz transport danych w wywołaniach RMI korzystając z mechanizmu SSL
3. W implementacji kalkulatora z poprzednich zajęć dodaj iloczyn skalarny wektorów, przy czym:
 - Wektory reprezentowane są jako klasa **Wektor** i tym samym przekazywane do wywołania jako obiekty (zwróć uwagę na to, co jest potrzebne do dystrybucji takiej aplikacji)Użyj analizatora sieciowego do obserwowania wymiany danych przy takim wywołaniu.
4. Do aplikacji kalkulatora dodaj możliwość transportu kodu klasy wektor przez HTTP
5. (*) Napisz prostą książkę telefoniczną przechowującą dane postaci: imię, nazwisko, telefon oraz posiadającą jedną operację - "szukajTelefonu" (pobierającą imię i nazwisko a zwracającą numer). Obiekt obsługujący takie wywołania powinien być ładowany do pamięci za pomocą mechanizmu aktywacji implementacji.