

Temat: serwer i klient do transferu plików pomiędzy dwoma komputerami

1. Zadanie polega na napisaniu dwóch aplikacji konsolowych działających w modelu klient-serwer:
 - a. Klient przesyłający pliki
 - b. Serwer odbierający pliki
2. Aplikacje będą służyły do przesyłania pliku od klienta do serwera i zapisywania ich po stronie serwera.
 - a. Klient:
 - i. umożliwi podanie adresu IP i numeru portu serwera,
 - ii. umożliwi wskazanie katalogu na dysku, którego zawartość (wraz z podkatalogami) ma zostać przesłana do serwera,
 - iii. wyświetli na bieżąco informację o tym, który plik jest aktualnie przesyłany i z jaką szybkością (wyrażoną w kilobitach na sekundę),
 - iv. po przesłaniu wszystkich plików wyświetli potwierdzenie tego, że wszystkie pliki dotarły do serwera i zostały przez niego poprawnie zapisane (np. serwerowi mogło się skończyć miejsce na dysku, albo jakiś plik mógł ulec zmianie podczas transmisji),
 - b. Serwer:
 - i. umożliwi podanie portu na którym działa, a także listy adresów IP z których będzie przyjmował transmisję (dopuszcza się użycie „wildcard”),
 - ii. umożliwi wskazanie katalogu w którym wszystkie odebrane pliki mają być zapisane,
 - iii. odbierze przesłane przez klienta pliki i zapisze je na dysku we wskazanym miejscu zachowując strukturę katalogów i podkatalogów,
 - iv. sprawdzi czy dane nie zostały zmodyfikowane po drodze i czy zostały poprawnie zapisane na dysku i poinformuje o tym klienta,
3. Wszystkie aplikacje piszemy w języku JAVA zgodnie ze standardem Java7 (JDK 1.7), a do komunikacji pomiędzy nimi wykorzystujemy standardowe gniazda omawiane na zajęciach (czyli klasy Socket, ServerSocket itp...).
4. Wszystkie pisane aplikacje powinny logować swoje działanie na konsoli lub do pliku – pozwoli to łatwiej i szybciej naprawiać ewentualne błędy, zwłaszcza na poziomie komunikacji Serwer ↔ Klient.
5. Za poprawną i pełną implementację zakresu opisanego w punkcie 2 projektu student może otrzymać do 4 punktów.
6. Projekty powinny zostać zapisane do odpowiednich katalogów w systemie EDUX do 16.11.2015 (termin może zostać zmieniony przez prowadzącego grupę).
7. Spakowany plik projektu powinien obejmować:
 - a. Pliki źródłowe (dla JDK 1.7)
 - b. Pliki binarne (class)

- c. Jeśli może być to pomocne (np. w przypadku konieczności podawania wielu parametrów podczas uruchomienia), skrypty dla systemu Linux (i/lub Windows zależy od indywidualnych ustaleń z prowadzącymi zajęcia) umożliwiające:
 - i. uruchomienie serwera,
 - ii. uruchomienie klienta.
 - d. Plik Readme.txt z opisem i uwagami autora (co nie działa lub jak uruchomić to co działa)
 - e. Jeśli aplikacja korzysta z dodatkowych bibliotek niedostępnych w standardzie JDK 1.7, biblioteki te muszą zostać dołączone do projektu wraz z odpowiednimi skryptami umożliwiającymi zarówno skompilowanie plików źródłowych projektu (nie plików źródłowych bibliotek) oraz uruchomienie obydwu aplikacji.
8. Prowadzący oceniać będą w pierwszym rzędzie organizację komunikacji po sieci, ale na ocenę wpływać będzie także zgodność wytworzonego oprogramowania z zasadami inżynierii oprogramowania i jakość implementacji (np. „wszystko w main”).
9. **JEŚLI NIE WYSZCZEGÓLNIŁO INACZEJ, WSZYSTKIE NIEJASNOŚCI NALEŻY PRZEDYSKUTOWAĆ Z PROWADZĄCYM ZAJĘCIA POD GROŻBĄ NIEZALICZENIA PROGRAMU W PRZYPADKU ICH NIEWŁAŚCIWEJ INTERPRETACJI.**