

Zadanie 1: Wielki Turniej SKJ (2018)

Zarys zadania

W sieci lokalnej trwa nieustający turniej. Każdy z graczy (agentów) musi rozegrać z każdym z pozostałych wersję gry “w marynarza”. Dla danej pary agentów A i B polega to na tym, że obaj podają równocześnie liczby n_A i n_B ; zwycięzcę wskazuje się przy pomocy odliczenia od 1 do $n_A + n_B$, rozpoczynając od jednego z graczy.

Nowy gracz może dołączyć do gry w **każdej** chwili. W dowolnej chwili może też ją opuścić, jeśli tylko rozegrał partie z **każdym** z obecnych w sieci.

Dokładniejszy opis

Gra rozgrywa się w sieci lokalnej, pomiędzy niezależnymi procesami, które potencjalnie mogą być uruchomione na różnych maszynach. Jediną dozwoloną formą komunikacji pomiędzy nimi jest przesyłanie wiadomości poprzez **nawiązane połączenia TCP**.

Początek

Każdy z graczy identyfikowany jest poprzez swoją **nazwę**, **numer IP** i **numer portu**. Aby wziąć udział w grze, należy znać **jednego aktywnego agenta wprowadzającego**. (Pierwszy z graczy może być swoim własnym agentem wprowadzającym.) Rozpoczęcie rozgrywki wygląda tak:

1. Agent A chce dołączyć do gry w której uczestniczy I . W tym celu agent A wysyła do agenta I komunikat **JOIN**.
2. Nowo dodany agent A pozyskuje (w sposób wybrany przez rozwiązującego zadanie) wiedzę o obecnych w sieci graczach.

Pojedynki

Po dołączeniu do gry wykonane zostają następujące kroki:

3. Dla każdego aktywnego gracza B (poza sobą samym), agent A rozgrywa prostą grę, w której:

- (a) Agent A przygotowuje liczbę n_A .
- (b) Agent B przygotowuje liczbę n_B .
- (c) Ustalone zostaje, od kogo zaczyna się odliczanie.
- (d) Odliczenie 1 od $n_A + n_B$ do wskazuje, kto wygrał.

Np. jeśli A wybrał 3 a B wybrał 7 , oraz ustalone zostało, że odliczanie zaczyna się od agenta B , to A wygrywa.

Zakończenie

Każdy z agentów może zakończyć pracę w **dowolnym momencie**, jeśli tylko **rozegrał gry z każdym ze znanych mu przeciwników**. Konkretniej:

- 4. Agent w każdym momencie może otrzymać komunikat **QUIT**, będący sygnałem zakończenia. Upewnia się wówczas, że **nie jest i nie będzie stroną w żadnej z rozgrywek**.
- 5. Agent kończący grę powinien poinformować o tym wszystkich graczy, którzy **usuną go, wraz z wynikami rozegranych z nim partii** ze swojej pamięci.
- 6. Agent kończy pracę wypisując podsumowanie rozegranych partii.

Uczciwość gry

Przebieg gry, opisany w punkcie (3) powyżej, **nie gwarantuje jej uczciwości**. W szczególności, nieuczciwy jest protokół w którym najpierw zostaje ustalone, że odliczanie zaczyna się od agenta A a następnie agent B wysyła liczbę n_B do agenta A . W tym bowiem przypadku agent A może wybrać n_A tak, by zagwarantować sobie wygraną.

Rozwiązujący zadanie powinien zaimplementować przebieg pojedynczej rozgrywki tak, by **szanse na zwycięstwo każdego z graczy były równe**. Akceptowalne rozwiązania obejmują zastosowania prostej kryptografii, budowę zewnętrznego źródła losowości i inne.

Dodatkowa funkcjonalność - monitor HTTP

Jako zadanie dodatkowe proponowana jest implementacja **monitora** - prostego serwera HTTP, prezentującego w czasie rzeczywistym wyniki partii rozegranych pomiędzy obecnymi w grze. (Agenci, którzy zrezygnowali z gry usuwani są również z bazy danych monitora.)

Specyfikacja rozwiązania

Program

Każdy z agentów wywoływany jest z następującymi parametrami:

- NAME: nazwa agenta.
- PORT: numer portu komunikacyjno-nasłuchowego.
- IIP: adres IP agenta wprowadzającego.
- IPORT: port agenta wprowadzającego.

Raport

Rozwiązanie powinno zostać opisane w dokumencie zapisanym w formacie **pdf**, zawierającym następujące podpunkty:

1. Ogólny opis rozwiązania

Musi zawierać przykład uruchomienia kilku agentów w danej grze. Może zawierać krótki opis rozwiązania i użytych technik.

2. Szczegółowy opis rozwiązania

Musi zawierać (1) opis tego, jak przechowywane są dane n/t sieci; (2) jak zaimplementowano rozgrywki po dołączeniu nowego gracza; (3) jak zapewniono uczciwość.

3. Obserwacje, eksperymenty i wnioski

Jak w tytule. W tym miejscu można opisać rozwiązania na które zabrakło czasu, wyniki skalowalnych eksperymentów, itp.

Rozwiązania bez dołączonego raportu nie będą przyjmowane.

Zadanie

1. Poprawny i pełny projekt wart jest 5 punktów (+1 dodatkowy):

- Za funkcjonalność obejmującą poprawne połączenie z siecią graczy można otrzymać **1 punkt**.
- Za poprawne i efektywne rozwiązanie problemu rozegrania przez dołączającego gracza gier ze wszystkimi obecnymi agentami otrzymać można **2 punkty**.
- Za zapewnienie uczciwości gry otrzymać można **2 punkty**.
- Za budowę dodatkowego monitora można otrzymać **1 punkt** (ale jedynie, jeśli poprzednie funkcjonalności zostały zaimplementowane).

2. Aplikację agenta piszemy w języku Java zgodnie ze standardem Java 8 (JDK 1.8). Do komunikacji przez sieć można wykorzystać jedynie podstawowe klasy do komunikacji z wykorzystaniem protokołu TCP (ServerSocket, Socket).
3. Projekty powinny zostać zapisane do odpowiednich katalogów w systemie EDUX w nieprzekraczalnym terminie 25.11.2018 (termin może zostać zmieniony przez prowadzącego grupę).
4. Spakowany plik projektu powinien obejmować:
 - Pełen opis rozwiązania zgodny z zamieszczonym wyżej formatem raportu.
 - Pliki źródłowe (dla JDK 1.8) (włącznie z wszelkimi bibliotekami nie należącymi do standardowej instalacji Javy, których autor użył) - aplikacja musi dać się bez problemu skompilować i uruchomić na komputerach w laboratorium w PJA.
5. Prowadzący oceniać będą w pierwszym rzędzie organizację komunikacji sieciowej i poprawność utrzymywanej wiedzy globalnej, ale na ocenę wpływać będzie także zgodność wytworzonego oprogramowania z zasadami inżynierii oprogramowania i jakość implementacji.
6. JEŚLI NIE WYSZCZEGÓLNIŁO INACZEJ, WSZYSTKIE NIEJASNOŚCI NALEŻY PRZEDYSKUTOWAĆ Z PROWADZĄCYM ZAJĘCIA POD GROŻBĄ NIEZALICZENIA PROGRAMU W PRZYPADKU ICH NIEWŁAŚCIWEJ INTERPRETACJI.