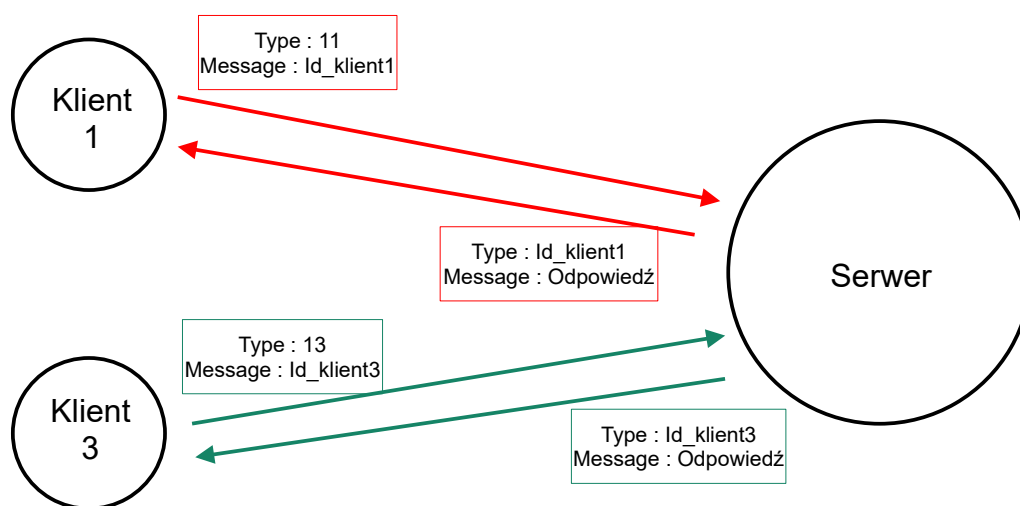


PROTOCOL

Protokół komunikacji między komponentami projektu (opis używanych struktur) :

Strukturę naszego komunikatora postanowiliśmy oprzeć na jednej kolejce komunikatów, do której podłączają się wszyscy użytkownicy. Nasz komunikator posiada tylko jeden serwer, który obsługuje wszystkie żądania. Użytkownik wysyła do serwera żądanie o pewnym typie oznaczającym pewną funkcję w komunikatorze, natomiast w wiadomości wysyła swój indywidualny numer Id (który jest jednocześnie jego nazwą). Serwer odbiera dany komunikat i odsyła odpowiedź na żądanie o unikatowym typie znanym tylko przez danego użytkownika (jego numerze Id). Klient posiada dwa oddzielne procesy. Jeden odpowiada za wykonywanie różnych funkcji, natomiast drugi za odbieranie wiadomości dla niego przeznaczonych (o typie takim samym jak jego numer Id). Pozwala to uniknąć zablokowania procesu, któregoś z klientów (np. ciągłego oczekiwania na wiadomość).



Schemat funkcjonowania komunikacji
klient-serwer

Po włączeniu aplikacji klientowi pojawia się komunikat pozwalający skorzystać mu z dwóch opcji : zalogować się do komunikatora lub opuścić komunikator. Jeżeli klient zdecyduje się na zalogowanie, zostaje zapytany o 4 cyfrowy numer, który stanie się jego identyfikatorem. Po podaniu prawidłowego numeru Id, wysyłany jest komunikat typu 11, zawierający w wiadomości identyfikator. Serwer odbiera taki sygnał i sprawdza czy nie ma już zalogowanego użytkownika o takim samym numerze. W zależności od dostępności odsyła on do klienta informację (komunikatem typu 12 na który czeka klient) o tym czy udało się dołączyć i go rejestruje (bądź nie) na serwerze.

Po prawidłowym zalogowaniu się do serwera, klient ma możliwość wykonania kilku opcji w aplikacji konsolowej :

- Wyświetlanie wszystkich użytkowników zalogowanych na serwerze : Po wybraniu tej opcji do serwera wysyłane jest zapytanie typu 13. Serwer odczytuje ten sygnał

jako prośbę o przesłanie identyfikatorów wszystkich zarejestrowanych użytkowników i taką informację odsyła w wiadomości typu „numer Id klienta”.

- Wyświetlanie wszystkich dostępnych pokoi: Serwer po otrzymaniu zapytania tego typu (16), przesyła listę pokoi obecnych na serwerze.
- Wyświetlanie wszystkich pokoi z użytkownikami: Serwer po otrzymaniu zapytania tego typu (17), przesyła listę użytkowników obecnych w każdym z pokoi.
- Zarejestrowanie się do danego pokoju: Wybierając tą opcję użytkownik wysyła serwerowi wiadomość typu 15 oznaczającą chęć rejestracji. Serwer w zamian wysyła zapytanie o numer pokoju, do którego użytkownik chce dołączyć. Po otrzymaniu informacji zwrotnej użytkownik dopisywany jest do odpowiedniego pokoju. Jeżeli użytkownik znajduje się już w jakimś pokoju i chce przenieść się do innego, wystarczy że też skorzysta z tej funkcji. Serwer zapewnia najpierw wyrejestrowanie klienta z pokoju w którym się znajduje w danej chwili, a dopiero później zapisuje do innego pokoju. Po dołączeniu do pokoju klientowi przekazywane jest ostatnie 10 wiadomości wysłanych w danym pokoju.
- Wysyłanie wiadomości do wybranego użytkownika: Decydując się na tą funkcję, użytkownik zostaje od razu zapytany o numer Id użytkownika do którego ma zostać wysłana wiadomość oraz o tekst tej wiadomości. Po dostarczeniu takich informacji, są one wysyłane do serwera. Serwer odbiera łącznie 3 komunikaty. Pierwszy zawierający informację o tym jaka funkcja ma zostać wykonana (typ 19), oraz kto ją zleca w wiadomości. Drugi zawierający numer Id osoby która ma otrzymać wiadomość, oraz trzeci z wiadomością. Dzięki tym 3 informacjom, serwer przeszukuje wszystkie pokoje w celu znalezienia, w którym pokoju znajduje się osoba wysyłająca wiadomość. Po jej znalezieniu, sprawdza listę osób w tym pokoju i jeśli odbiorca również się w nim znajduje, zostaje wysłana wiadomość o typie „Id odbiorcy” (taka wiadomość może być odebrana tylko przez odbiorcę, ponieważ numer Id jest unikatowy).
- Wysyłanie wiadomości do wszystkich użytkowników w pokoju: Użycie tej funkcji jest bardzo zbliżone do wysyłania wiadomości do danego użytkownika. Klient tym razem wysyła jednak tylko dwa komunikaty. Pierwszy zawierający informację co należy wykonać (typ 18) oraz kto wysyła żądanie (zawartość message). Drugi zawierający wiadomość, do wysłania do całego pokoju. Serwer znajduje pokój klienta i wysyła wiadomość do wszystkich numerów Id przypisanych do tego pokoju.
- Opuszczanie serwera: Serwer otrzymuje komunikat typu 14 i usuwa klienta, który wysłał komunikat ze wszystkich rejestrów.

Po stronie serwera, wykorzystaliśmy nieskończoną pętlę while odpowiadającą za oczekiwanie na nowe zapytania do momentu wyłączenia aplikacji. Serwer w pętli czeka na wiadomość o dowolnym typie, a w momencie jej otrzymania przechodzi do wykonywania odpowiedniego żądania (danego typu). Po jego wykonaniu wraca na początek pętli i znowu czeka na nowe żądanie od klientów. W międzyczasie wyświetla pewne informacje informujące nas na jakim etapie pracy jest oraz co wykonał do tej pory.