

# Prosta aplikacja klient-serwer

## Wstęp

Architektura klient-serwer jest jednym z najpopularniejszych modeli architektonicznych wykorzystywanych w systemach komputerowych. W tej architekturze komunikacja między aplikacjami odbywa się poprzez podział na dwie główne role: klienta i serwera.

Klient to aplikacja lub urządzenie, które inicjuje żądania i korzysta z usług dostarczanych przez serwer. Klient może być programem działającym na komputerze, smartfonie, tablecie lub innym urządzeniu. Przykładowe zadania klienta to wysyłanie żądań do serwera, pobieranie danych, prezentowanie informacji użytkownikowi, interakcja z użytkownikiem itp.

Serwer to program lub urządzenie, które udostępnia określone usługi lub zasoby dla klientów. Serwer jest odpowiedzialny za obsługę żądań klientów, przetwarzanie danych, wykonywanie operacji i udostępnianie wyników klientom. Serwer może działać na dedykowanym sprzęcie lub wirtualnym środowisku, które umożliwia obsługę wielu klientów jednocześnie.

Komunikacja między klientem a serwerem odbywa się za pomocą protokołów sieciowych, takich jak TCP/IP lub HTTP. Klient wysyła żądanie do serwera, a serwer odpowiada na to żądanie, przetwarza je i zwraca odpowiedź do klienta. Przykłady protokołów wykorzystywanych w architekturze klient-serwer to HTTP w przypadku aplikacji internetowych, SMTP w przypadku poczty elektronicznej, FTP w przypadku transferu plików itp.

W architekturze klient-serwer, serwer często pełni rolę centralnego punktu dostępu do zasobów lub usług, a klienci korzystają z tych zasobów lub usług poprzez żądania i odpowiedzi. Ta architektura umożliwia łatwą skalowalność, podział pracy oraz zapewnienie bezpieczeństwa i kontroli dostępu do zasobów.

Architektura klient-serwer jest szeroko stosowana w różnych dziedzinach, takich jak aplikacje internetowe, usługi sieciowe, gry sieciowe, systemy baz danych, systemy zarządzania siecią itp. Zapewnia ona elastyczność, skalowalność i możliwość tworzenia rozbudowanych i złożonych systemów komputerowych.

## Zadanie

Kod przedstawia aplikację typu klient-serwer do czatu. Kod składa się z dwóch części, które podobnie jak poprzednie zadanie ma więcej niż jedną wersję:

Serwer – czeka na połączenia od klientów i odpowiada za obsługę połączeń. Jest wersja w Javie i w



server.py



ChatServer.java

Pythonie. Można je uruchomić jednocześnie tylko na różnych portach.

Klienta – łączy się z serwerami. Jest wersja w Pythonie, w Javie z biblioteką Swing i w Javie z biblioteką



JavaFX.

client.py



ChatClientSwing.java



ChatClientJavaFX.java

Wystarczy wybrać jeden plik kodu serwera i jeden klienta. Klient i serwer nie muszą być w tym samym języku. Analizowanie pozostałych wersji jest opcjonalnie dla chętnych. Jeżeli ktoś woli przepisać sobie

kod na inny język programowania wspierający programowanie obiektowe lub użyć innej biblioteki, to jest taka możliwość.

Tym, którzy nie wiedza jaką wersję plików wybrać polecam serwer napisany w Javie i klienta napisanego z użyciem biblioteki Swing.

Aby wykonać poniższe ćwiczenie należy zastosować wiedzę z poprzednich zajęć, wybrać kod i zrealizować poniższe:

1. Poprawić kod serwera by obsługiwał więcej niż jedno połączenie. Poprawne rozwiązanie powinno pozwalać na uruchomienie na porcie jednego serwera, ale kilku klientów.
2. Poprawić kod tak, aby wyświetlał się użytkownik, który wysłał wiadomość na czat.
3. Dodać możliwość dodania własnej nazwy użytkownika zamiast losowej.
4. Dodać możliwość wyświetlania listy użytkowników na czacie. Może to być wyświetlana lista, do której będą dodawani użytkownicy. Może to być przycisk, który wyświetli listę wszystkich aktualnie obecnych.