

Przed przystąpieniem do pisania programu koniecznie zapoznaj się z całą treścią zadania.

Do kolokwium jest dołączony plik Client.java, który stanowi kompletną aplikację klienta, działającą z serwerem, który należy napisać. Po napisaniu kolokwium, katalog ze źródłami należy spakować do formatu zip lub tar.gz. Spakowany plik należy wysłać na kampus. Po zakończeniu kolokwium proszę nie wyłączać komputera.

Podczas kolokwium można korzystać ze wszystkich zasobów internetu, w tym samodzielnie przygotowanych. Zabroniona jest wzajemna komunikacja oraz publikowanie i dostęp do kodu stworzonego na potrzeby kolokwium. Wszystkie działania wykonywane na ekranie komputera są rejestrowane i mogą zostać odtworzone przez prowadzących. Zabronione jest korzystanie z innych urządzeń niż udostępniony komputer.

Napisz serwer, który w jednej chwili będzie akceptował połączenie tylko od jednego klienta. Po obsłudze jednego klienta, serwer powinien móc obsłużyć kolejnego. Obsługa każdego z klientów powinna wyglądać w ten sam sposób:

1. Otrzymanie od klienta pliku PNG.
2. Zapis pliku.
3. Odczytanie liczby całkowitej z interfejsu użytkownika.
4. Przekształcenie obrazu algorytmem box blur.
5. Zapisanie podsumowania przekształcenia do bazy danych.
6. Odesłanie pliku z przekształconym obrazem klientowi.

Poniżej omówione są wymienione funkcjonalności:

Ad 1.

Serwer powinien oczekiwać na przesłanie przez użytkownika pliku graficznego png. Zakładamy, że użytkownik prześle wyłącznie taki plik. Sposób przesłania pliku jest dowolny - można zmodyfikować kod klienta, choć nie jest to konieczne.

Ad 2.

Po otrzymaniu pliku należy zapisać go w katalogu images znajdującym się w ścieżce uruchomienia programu. Jeżeli katalogu nie ma, program powinien go utworzyć. W nazwie pliku powinien znaleźć się znacznik czasowy w dowolnej postaci. Zapisany plik może mieć nazwę na przykład 2023-06-16_12:50:35.jpg.

Ad 3.

Serwer powinien posiadać zaimplementowany interfejs użytkownika służący do ustawienia rozmiaru jądra splotu. Interfejs powinien posiadać dwa elementy:

- suwak (slider) pozwalający modyfikować promień.
- etykieta (label) wyświetlająca promień filtra.

Promień filtra powinien przyjmować wyłącznie wartości nieparzyste z zakresu 1 - 15.

Ad 4.

Przesłany obraz powinien zostać przekształcony z wykorzystaniem algorytmu box blur (https://en.wikipedia.org/wiki/Box_blur) z równymi wagami.

Algorytm polega na zapisaniu do zmodyfikowanego piksela, średniej arytmetycznej pikseli znajdujących się w jego sąsiedztwie. Rozmiar sąsiedztwa odpowiada rozmiarowi jądra splotu. Rozmiar jądra (w przykładzie jest to 3) powinien być definiowany w interfejsie użytkownika. Algorytm powinien przekształcić niezależnie trzy kanały tego obrazu (RGB). Algorytm należy zrównoleglić tak, aby obszar obrazu został podzielony równomiernie lub prawie równomiernie pomiędzy wątki. Liczba wątków powinna być równa liczbie rdzeni w procesorze. Sposób przekształcenia pikseli na krawędziach obrazu jest dowolny.

Ad 5.

Jeżeli w katalogu images nie znajduje się plik index.db, należy utworzyć taki plik i założyć w nim bazę danych SQLite. Można to zrobić na przykład w czasie uruchomienia programu. Baza powinna zawierać jedną tabelę składającą się z kolumn:

- id (autoinkrementowany integer),
- path (tekst), zawierający ścieżkę kopii pliku graficznego zapisanego na serwerze,
- size (integer), rozmiar filtra odczytany z interfejsu użytkownika,
- delay (integer), czas działania algorytmu box blur w milisekundach.

Jeżeli istnieje plik z bazą, zakładamy, że znajduje się w nim poprawna tabela. Dla każdego konwertowanego obrazu należy dodać wpis do tej tabeli.

Ad 6.

Po zakończeniu wcześniejszych działań, należy odesłać klientowi przekształcony plik, zakończyć połączenie z klientem i oczekiwać na kolejne.