



Sprawdzian #2

© PIOTR SIEWNIAK 2023

Zadanie 1

Napisać aplikację pozwalającą na obliczenie pola i obwodu prostokąta dla danych wejściowych (długości boków) wprowadzonych z klawiatury. Wykorzystać strumienie standardowe: wejścia `stdin` i wyjścia `stdout`.

Zadanie 2

Utworzyć serwer webowy (serwer HTTP) pozwalający na przesłanie do klienta bieżącej daty (dzień, miesiąc, rok) w formacie JSON. Ponadto, wspomniana data powinna zostać zapisana w pliku tekstowym w postaci bufora należącego do klasy `Buffer`.

Zadanie 3

Utworzyć serwer HTTP pozwalający na przesłanie do klienta danych samochodu osobowego w formacie JSON. Uwzględnić następujące dane samochodu: marka, model, rok produkcji, cena, data pierwszej rejestracji (reprezentowana za pomocą obiektu) oraz powiaty, w których samochód był dotychczas rejestrowany (zapisane w tablicy o odpowiednim rozmiarze).

Zadanie 4

Utworzyć serwer HTTP pozwalający na zaprezentowanie klientowi zawartości trzech przygotowanych wcześniej plików tekstowych. Każdy z plików ma format JSON i zawiera dane samochodu osobowego: marka, model, rok produkcji, data pierwszej rejestracji, numer rejestracyjny.

Odczyt plików tekstowych zawierających wymienione powyżej dane samochodów zrealizować w sposób asynchroniczny z wykorzystaniem:

- a) funkcji zwrotnych (wariant I);
- b) obietnic (wariant II);
- c) funkcji asynchronicznych `async` (wariant III).

Zaimplementować kompleksową obsługę błędów.

Zadanie 5

Utworzyć serwer HTTP pozwalający na obsługę prostej strony WWW (witryny internetowej), która zawiera kilka podstron. Tematyka strony: samochody ciężarowe – wywrotki o ładowności ponad 10 ton, w latach 70-tych XX wieku w państwach bloku wschodniego.

Kod HTML (wraz z określeniem stylów CSS) odpowiadający stronie głównej i poszczególnym podstronom (wywrotkom) powinien być zapisany w niezależnych, zewnętrznych plikach tekstowych (HTML).

Odczyt plików tekstowych zawierających kod HTML zrealizować w sposób asynchroniczny z wykorzystaniem:

- d) funkcji zwrotnych (*wariant I*);
- e) obietnic (*wariant II*);
- f) funkcji asynchronicznych `async` (*wariant III*).

Zaimplementować kompleksową obsługę błędów.