

LISTA 6: WĄTKI

Wybierz jedno z poniższych zadań. 3 punkty obowiązkowe. Możliwe punkty bonusowe za pełne rozwiązania.

Ćwiczenie 1. Dodaj współbieżność / zrównoleglenie do dowolnego wcześniej zrobionego przez Ciebie na kurs Pascala programu.

Ćwiczenie 2. Napisz moduł implementujący *futures*, konkretnie moduł `futures` z klasą generyczną `generic Future<T>`. Konstruktor tej klasy bierze argument typu proceduralnego / wskaźnik na metodę / obiekt, mające policzyć wartość typu `T` w osobnym wątku, i rejestrują pracę (w prostszym wariacie: startują wątek) do obliczenia tej wartości. `Future<T>` ma metodę `Get`, czekającą na policzenie i zwracającą policzoną wartość. Ma też metodę `Forget`, która anuluje liczenie wartości oraz zwalnia obiekt `self` (być może od razu, a może w momencie kiedy miałyby się zacząć liczyć wartość). Być może wykorzystaj menedżera prac z programów do wykładu.

Ćwiczenie 3. * Napisz moduł implementujący *Map-Reduce*, konkretnie moduł `map_reduce` z klasą generyczną `generic MapReduce<D,K,V,R>`. Konstruktor bierze dwie funkcje / metody: pierwszą `Map` biorącą argument typu `D` i liczącą parę: klucz typu `K` oraz listę / iterator wartości typu `V`. Drugi argument konstruktora to funkcja / metoda `Reduce` biorąca argument typu `V` i argument typu `R`, i produkująca zaktualizowany rezultat typu `R`. Konstruktor `MapReduce<D,K,V,R>` bierze też początkową wartość typu `R`. Klasa `MapReduce<D,K,V,R>` ma też metody: `procedure AddData (Data: D); function Query (Key: K) : R; procedure WaitForResults`; `AddData` wrzuca do policzenia na wątkach roboczych następujący proces: z danej `Data` policz przez `Map` klucz `Key` oraz listę / iterator wartości typu `V`; a następnie krok po kroku aktualizuj wartość typu `R` przypisaną do `Key` metodą `Reduce` na kolejnych wartościach typu `V`. `Query` zwraca wartość typu `R` aktualnie przypisaną do `Key`. `WaitForResults` czeka aż wszystkie wrzucone dane zostaną przetworzone, i pośrednie wartości zredukowane.

(Hmm... To zadanie jest chyba bardzo pracochłonne.)