

Paradygmaty programowania - studia stacjonarne, lista 2, 16.10.2018, godzina 09:15
Czas na rozwiązanie: 70 min.

WAŻNE1: Do każdego zadania przygotuj 3-5 testów sprawdzających poprawność działania!

WAŻNE2: Nie wolno stosować funkcji bibliotecznych i konstrukcji innych niż te przedstawione na wykładzie 1.

1. Napisz funkcję, która na wejściu przyjmuje parę dowolnych elementów i zamienia ją miejscami.

Przykład:

wywołanie: `zamien(3., 'a')`

wynik : `('a', 3.)`.

Punkty: 5 (język Ocaml)

2. Napisz funkcję, która na wejściu przyjmuje dwa parametry: listę liczb rzeczywistych i funkcję na liczbach rzeczywistych zwracającą liczbę rzeczywistą - *func*. Napisz funkcję, która zwróci listę elementów po przekształceniu elementów z listy liczb rzeczywistych za pomocą funkcji *func*. Elementy powinny być przekształcone w kolejności w jakiej występują na liście wejściowej. Przygotować przykłady dwóch wersji funkcji *func*:

- Kwadrat wartość
- Podwojona wartość

Podpowiedź: Użyj typu `Function[Double, Double]` do definicji typu funkcyjnego dla *func*

Przykład:

wywołanie `fun_kwad 2;;`

wynik : `4`

wywołanie: `przekształc [5.;3.;2.] fun_kwad;;`

wynik : `[25.,9.,4.]`”.

Punkty: 9 (język Scala)

3. Napisz funkcję, która na wejściu przyjmuje dwa parametry: listę napisów oraz napis będący separatorem. Na wyjściu zwraca pojedynczy napis będący połączeniem napisów wejściowych oddzielonych od siebie separatorem.

Przykład:

wywołanie: `polacz ["To"; "jest"; "napis"] "-";;`

wynik : "To-jest-napis"

Punkty: 6 (język Ocaml)