

# Paradygmaty programowania

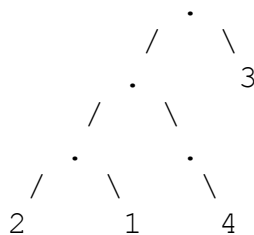
## Lista 6

**Zad. 1. [2p]** Zdefiniować typ `dane_slow` określający następujące cechy słowa zależne od jego długości: puste, krótkie (do 10 znaków), średnie (od 11 do 20 znaków) i długie (powyżej 20 znaków). Dla słów niepustych typ ma dodatkowo zawierać informację określoną przez typ `litera` o zawieraniu zadanej litery. Napisać w OCaml funkcję typu `string list -> char -> dane_slow list`, która zmieni listę słów na listę informacji o słowie. Zastosować funkcje `contains` i `length` z pakietu `String`.

**Zad. 2. [3p]** Zdefiniować w Scala typ `Samochod` będący aliasem do krotki zawierającej dane o marce, modelu i roczniku samochodu a następnie alias do listy elementów typu `Samochod` o nazwie `Samochody`. Dla elementu typu `Samochody` zawierającego wartości (Opel, astra, 1999), (Renault, megane, 2004), (Opel, corsa, 2009), (Nissan, micra, 2004), (Opel, corsa, 2009), (Nissan, micra, 2003), wykorzystując odpowiedni funkcjonal, napisać funkcję zwracającą listę par (marka, liczba wystąpień marki).

**Zad. 3. [5p]** Zdefiniować w Scala drzewo elementów dowolnego typu, w którym występują tylko węzły i wartości (wartość oznacza koniec gałęzi drzewa) a następnie napisać funkcję, która przepisuje w porządku postfiksowym zbudowane drzewo do listy heterogenicznej elementów: para("Dana", etykieta) i łańcuch znaków charakteryzujący rodzaj węzła np. "Wezel (lewo, element)", gdzie etykieta oznacza pamiętaną w drzewie wartość.

Przykład drzewa do zadania:



Rozwiązanie dla powyższego drzewa:

```
List(P((Dana,2)), P((Dana,1)), S(Wezel (element, element)),
      P((Dana,4)), S(Wezel (element)), S(Wezel (lewo,prawo)),
      P((Dana,3)), S(Wezel (lewo,element)))
```