

Ćwiczenie 5

Haskell - funkcje anonimowe, funkcje wyższych rzędów, definiowanie typów

1 Wprowadzenie

1.1 Funkcje anonimowe

Notacja lambda pozwala na definiowanie funkcji anonimowych.

```
\ wzorzec -> wyrażenie
```

1.2 Funkcje wyższych rzędów

Można definiować funkcje wyższych rzędów, w których argumentami są inne funkcje.

Przykład funkcji wyższego rzędu:

```
until funkcja1 funkcja2 x
```

(dopóki **funkcja1** stosuj **funkcja2** dla wartości **x**).

1.3 Składanie list

Przykład obliczenia dwukrotności elementów na liście [1,2,3]:

```
[x*2 | x <- [1,2,3]]
```

1.4 Definiowanie typów

Typy danych definiowane są za pomocą słowa kluczowego **data**.

```
data Konstruktor_typu parametr = Konstruktor_danych_1
  | Konstruktor_danych_2
  | ...
  deriving(Show)
```

deriving(Show) - dziedziczenie funkcji **Show** w celu uzyskania możliwości wyświetlania zdefiniowanego typu danych.

2 Zadania

2.1

Zdefiniuj funkcję obliczającą pierwiastek kwadratowy metodą kolejnych przybliżeń. Przetestuj zdefiniowaną funkcję.

2.2

Zdefiniuj funkcje:

- całkowania metodą prostokątów,
- całkowania metodą trapezów,

z argumentami:

- funkcja podcałkowa podawana w notacji lambda,
- dolna granica przedziału całkowania,
- górna granica przedziału całkowania,
- liczba przedziałów, na który podzielony jest przedział całkowania.

Przetestuj zdefiniowane funkcje.

2.3

Wykorzystując rekurencję zdefiniuj typ drzewa binarnego. Następnie zdefiniuj funkcje:

- wyznaczającą rozmiar drzewa,
- wyświetlającą drzewo w wybranej postaci tekstowej,
- wyznaczającą liczbę liści w drzewie,
- przekształcającą drzewo na uporządkowaną listę wartości,
- sprawdzającą czy dany element występuje w drzewie,
- dodającą nowy element do drzewa (tak aby drzewo było binarnym drzewem poszukiwań).

Przetestuj zdefiniowane: typ oraz funkcje.