

Typy danych

zadania

Zadanie 1

Zdefiniować typ

```
data Student = Student {imie :: String,  
                        nazwisko :: String,  
                        nrAlbumu :: Int  
                        }
```

jako instancję Show w taki sposób, aby kolejne wartości pól były wypisywane po dwukropkach, w kolejnych liniach.

Zadanie 2

Zdefiniować typ `Calkowite` przez `Zero`, następnik całkowitej oraz poprzednik całkowitej (podobnie, jak `Naturalne` na slajdach).
Napisać funkcję, która konwertuje `Integer` do typu `Calkowite` oraz drugą funkcję, które konwertuje `Calkowite` na `Integer`.

Zadanie 3

Dla typu

```
data Tree a = Empty | Node a (Tree a) (Tree a)
```

Oraz drzewa z liczbami typu Integer zdefiniować funkcję, która sprawdza, czy wszystkie wierzchołki drzewa mają liczby parzyste.

Zadanie 4

Dla typu

```
data Tree a = Empty | Node a (Tree a) (Tree a)
```

zdefiniować trzy funkcje, zwracające listę jego elementów w porządku odpowiednio preorder, inorder oraz postorder.

Zadanie 5

Zdefiniować typ odpowiadający liczbom całkowitym Gaussa (liczby zespolone, których część rzeczywista jest całkowita oraz część urojona jest całkowita). Uczynić typ instancją Show oraz Num. Ponieważ w Num trzeba zdefiniować wartość bezwzględną oraz znak, zdefiniować pierwszą jako identyczność, a drugą jako stale równą jeden. Funkcję show zdefiniować tak, aby liczba $a + bi$ została wypisana jako $a+bi$

Zadanie 6

Zdefiniować typ odpowiadający macierzom 2×2 o współczynnikach całkowitych, zaimplementować dodawanie, mnożenie i wypisywanie macierzy (wypisywanie w możliwie czytelny sposób).