Csoportok

Eseményvezérelt alkalmazások
IP-18bEVALKEG | 12

Eseményvezérelt alkalmazások
IP-18bEVALKEG | 91

Eseményvezérelt alkalmazások
IP-18bEVALKEG | 92

3. ZH - B csoport

Kategória: Vizsgafeladatok
Elérhető: 2022. 11. 11. 8:20
Pótolható határidő:
Végső határidő: 2022. 11. 11. 8:55
Kiírta: Erdei Zsófia

Leírás:

Funkcionális programozás 3. ZH - B csoport

Előzetes tudnivalók

Használható segédanyagok:

- Haskell könyvtárak dokumentációja,
- Hoogle,
- a tárgy honlapja, és a
- Haskell szintaxis összefoglaló.

Más segítőeszköz nem használható.

Ha bármilyen kérdés, észrevétel felmerül, azt a gyakorlatvezetőnek kell jelezni, **nem** a diáktársaknak!

A feladatsor megoldására 30 perc áll rendelkezésre.

A feladatok tetszőleges sorrendben megoldhatóak. A pontozás szabályai a következők:

- Minden teszten átmenő megoldás ér teljes pontszámot.
- Funkcionálisan hibás (valamelyik teszteseten megbukó) megoldás nem ér pontot.
- Fordítási hibás vagy hiányzó megoldás esetén a teljes megoldás 0 pontos.

Ha hiányos/hibás részek lennének a feltöltött megoldásban, azok kommentben szerepeljenek.

Tekintve, hogy a tesztesetek, bár odafigyelés mellett íródnak, nem fedik le minden esetben a függvény teljes működését, határozottan javasolt még külön próbálgatni a megoldásokat beadás előtt!

Az elméleti kérdésekre adott választ a forráskódban kell elhelyezni, kommentben. Minden függvénynek meg kell adni a típusszignatúráját is. A függvények elvárt neve és típusa zárójelben meg van adva.

`Zarthelyi3` néven kell deklarálni a modult. A `.hs` fájlt `.zip` -be tömörítve kell beadni.

Elméleti kérdések

Adott a következő adattípus:

```
data Allat = Kutya String Int | Macska String Int | Horcsog
```

1. Adj meg olyan kifejezést, amelyet ha paraméterül adunk a `f` függvénynek 2 legyen a visszatérési értéke!

```
f :: [Allat] -> Int
f (_:_:Macska _ 1:_)             = 0
f (_:_:Macska [] _:[])              = 1
f (_:Kutya [_] _:Macska [] _:[xs])  = 2
f (_:Kutya _ _:Macska _ _:xs)       = 3
```

Gyakorlati feladatok

1. Hőmérséklet adattípus (1 pont)

Készíts egy új adattípust `Temperature` néven, aminek két konstruktora van `C` és `F`, mindkettő egy-egy `Double` -t vár paraméterként. Kérd az `Eq` és a `Show` típusosztályok

automatikus példányosítását! A típusban hőmérsékleti adatokat lehet tárolni Celsius illetve Fahrenheit formátumban.

2. Átváltás (1 pont)

Készítsd el a `convertTemp :: Temperature -> Temperature` egy konverziós függvényt, amivel a két reprezentáció között lehet konvertálni.

```
hőmérséklet Fahrenheitben = 1,8*(hőmérséklet Celsiusban) + 32
```

```
convertTemp (C 0) == F 32
convertTemp (F (-40)) == C (-40)
convertTemp (convertTemp (F 0)) == F 0
convertTemp (F 32) == C 0
convertTemp (F 68) == C 20
convertTemp (C 25) == F 77
```

3. Minimum és maximum hőmérséklet (3 pont)

Az előző feladatban definiált `Temperature` típust használva definiáld a `minMaxTemp :: [Temperature] -> (Temperature, Temperature)` függvényt, amely hőmérsékleti adatok alapján kiválasztja a leghidegebb és legmelegebb értéket is és visszaadja az eredményt egy rendezett pár formájában. Az eredményt minden esetben Celsiusban add meg! A listának legalább egy eleme van.

```
minMaxTemp [C 20] == (C 20, C 20)
minMaxTemp [C 10, C 20, C (-10)] == (C (-10), C 20)
minMaxTemp [C 10, C 20, C (-10), F (-40)] == (C (-40), C 20)
minMaxTemp [F 32, F 77, F 12, F 54, F 5] == (C (-15), C 25)
minMaxTemp [F 59, C 15, C 18, F 68] == (C 15, C 20)
```

Megoldás

Letöltés

Név:	Zarthelyi3.zip
Feltöltés ideje:	2022. 11. 11. 8:51
Értékelés:	
Státusz:	Elfogadva
Feltöltések száma:	1
Értékelte:	Erdei Zsófia
Megjegyzések:	0 - 2

Automatikus tesztelés eredményei

```
Valamelyik tesztesetre hibásan futott le a beadott program.

Megbukott tesztek:

## 7.
## Kivétel:
Eval.hs:61:223: error:
    Variable not in scope:
      minMaxTemp :: [Temperature] -> (Temperature, Temperature)
(deferred type error)

## Teszteset:
minMaxTemp [C 20] == (C 20, C 20)


## 8.
## Kivétel:
Eval.hs:61:257: error:
    Variable not in scope:
      minMaxTemp :: [Temperature] -> (Temperature, Temperature)
(deferred type error)

## Teszteset:
minMaxTemp [C 10, C 20, C (-10)] == (C (-10), C 20)


## 9.
## Kivétel:
Eval.hs:61:309: error:
    Variable not in scope:
      minMaxTemp :: [Temperature] -> (Temperature, Temperature)
(deferred type error)

## Teszteset:
minMaxTemp [C 10, C 20, C (-10), F (-40)] == (C (-40), C 20)


## 10.
## Kivétel:
Eval.hs:61:370: error:
    Variable not in scope:
      minMaxTemp :: [Temperature] -> (Temperature, Temperature)
(deferred type error)

## Teszteset:
minMaxTemp [F 32, F 77, F 12, F 54, F 5] == (C (-15), C 25)


## 11.
## Kivétel:
Eval.hs:61:430: error:
    Variable not in scope:
      minMaxTemp :: [Temperature] -> (Temperature, Temperature)
(deferred type error)

## Teszteset:
minMaxTemp [F 59, C 15, C 18, F 68] == (C 15, C 20)
```