

Csoportok

Eseményvezérelt alkalmazások

IP-18bEVALKEG | 12

Eseményvezérelt alkalmazások

IP-18bEVALKEG | 91

Eseményvezérelt alkalmazások

IP-18bEVALKEG | 92

3. ZH - A 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 - A 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 típuszignatúráját 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 Peksuti = Kifli String | Zsemle String Bool | KakaosCsiga Int
```

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

```
f :: [Peksuti] -> Int
f ( _:_:Zsemle _ True:_)           = 0
f ( _:_:Zsemle [] _:[])           = 1
f ( _:Kifli []:Zsemle [] _:[xs])  = 2
f ( _:Kifli _:Zsemle _ _:[xs])    = 3
```

Gyakorlati feladatok

1. Időjárás adattípus (1 pont)

Készíts egy **Weather** adattípust, amely az időjárást reprezentálja! Rendelkezzen a következő adatkonstruktorokkal: **Sunny** , **Cloudy** , **Rainy** . A **Sunny** és a **Cloudy**

adatkonstruktornak legyen egy `Int` paramétere, amely az UV sugárzás erősségét fogja jelölni. Kérd az `Eq` és a `Show` típusosztályok automatikus példányosítását!

2. UV riasztás (1 pont)

Definiáld a `highUVwarning :: Weather -> Bool` függvényt, amely segít eldönteni, hogy UV-riasztás van-e! UV riasztás akkor van érvényben, ha az UV index 5-nél magasabb.

```
highUVwarning Rainy == False
highUVwarning (Sunny 5) == False
highUVwarning (Sunny 6) == True
highUVwarning (Cloudy 5) == False
highUVwarning (Cloudy 6) == True
```

3. Sokáig magas UV (3 pont)

Adott egy lista, amely valahány egymás után következő nap időjárásjelentését tartalmazza. Definiáld a `longtermHighUV :: [Weather] -> Int -> Bool` függvényt, amely eldönti, hogy volt-e olyan egymást követő `n` db nap, amikor UV riasztást rendeltek el!

```
longtermHighUV [Cloudy 1, Rainy, Sunny 1, Sunny 1] 1 == False
longtermHighUV [Cloudy 1, Rainy, Sunny 1, Sunny 6] 1 == True
longtermHighUV [Sunny 6, Cloudy 6, Sunny 1, Sunny 6] 3 == False
longtermHighUV [Sunny 6, Sunny 6, Sunny 1, Sunny 8, Cloudy 6, Sunny 7] 3 == True
longtermHighUV (repeat (Sunny 10)) 5 == True
```