

Csoportok

Eseményvezérelt alkalmazások

IP-18bEVALKEG | 12

Eseményvezérelt alkalmazások

IP-18bEVALKEG | 91

Eseményvezérelt alkalmazások

IP-18bEVALKEG | 92

Pót ZH péntek 8-10

Kategória:	Vizsgafeladatok
Elérhető:	2022. 12. 02. 9:00
Pótolható határidő:	
Végső határidő:	2022. 12. 02. 9:35
Kiírta:	Erdei Zsófia

Leírás:

Pót zárthelyi

Elméleti kérdése

1. elméleti kérdés

Mi a különbség az `x`, az `xs` és az `[xs]` minták közt?

2. elméleti kérdés

Miért nem fordul az alábbi kód?

```
g = map [1,2,0] (1+)
```

Gyakorlati feladatok

Munka tervezése (2 pont)

Adottak munkák, melyeket egymás után kell elvégezni. Adott egy függvény, amely minden munkára megadja, hogy hány nap alatt végezhető el az adott feladat. Implementáld a `schedule :: [a] -> (a -> Int) -> Int` függvényt, amely megadja hogy az adott munkák mennyi idő alatt végezhetőek el leghamarabb!

```
schedule [] (\ x -> 5) == 0
schedule ["design", "implementation", "testing", "maintenance"] (\ x -> if x == "design" then 3 else if x == "implementation" then 6 else if x == "testing" then 1 else if x == "maintenance" then 3) == 13
schedule ["implementation", "testing", "maintenance"] (\ x -> if x == "design" then 3 else if x == "implementation" then 6 else if x == "testing" then 1 else if x == "maintenance" then 3) == 13
schedule ["implementation", "maintenance"] (\ x -> if x == "design" then 3 else if x == "implementation" then 6 else if x == "testing" then 1 else if x == "maintenance" then 3) == 13
schedule ["implementation"] (\ x -> if x == "design" then 3 else if x == "implementation" then 6 else if x == "testing" then 1 else if x == "maintenance" then 3) == 6
schedule [1,2,3] (\ x -> if x == 1 then 3 else if x == 2 then 6 else if x == 3 then 1) == 10
```

Számítógép

Típusdeklaráció (1 pont)

Definiáld a `Computer` típust, amelyhez a `Comp` konstruktor tartozik, amely két paramétert vár, a RAM mennyiségét MiB-ban (`Integer`) és a processzor sebességét MHz-ben (`Integer`). Kérjük a fordítótól az `Eq` automatikus példányosítását!

Létezzen a `Computers` típuszsinoníma, amely `Computer` -ek listájával ekvivalens.

```
(([(Comp 1024 2400) :: Computer, (Comp 2048 3600) :: Computer] :: Computers) == [(Comp 1024 2400) :: Computer, (Comp 2048 3600) :: Computer]) == True
not ((([(Comp 1024 2400) :: Computer, (Comp 2048 3600) :: Computer] :: Computers) == [(Comp 1024 3600) :: Computer, (Comp 1024 3600) :: Computer]) == True)
Comp 1024 3600 == Comp 1024 3600
not (Comp 1024 3600 == Comp 1023 3600)
not (Comp 1024 3600 == Comp 1024 3599)
not (Comp 1024 3600 == Comp 1023 3601)
```

Google Chrome (1 pont)

A kedvenc böngésződnek legalább 16 GiB (16384 MiB) RAM szükséges. Definiáld a `goodEnoughComputer` függvényt, amely megadja, hogy `Computer` -ek listájában hány olyan számítógép van, amelyen képes a kedvenc böngésző futni! (`goodEnoughComputer :: Computers -> Int`)

```
goodEnoughComputer [] == 0
goodEnoughComputer [Comp 16384 2400] == 1
goodEnoughComputer [Comp 16384 2400, Comp 32768 3600] == 2
goodEnoughComputer [Comp 16383 2400, Comp 32768 3600] == 1
goodEnoughComputer [Comp 16384 128, Comp 32768 3600] == 2
goodEnoughComputer (concat (((replicate 10 [Comp 16384 128])++(replicate 5 [Comp 16384 128])++(replicate 5 [Comp 16384 128])++(replicate 5 [Comp 16384 128])++(replicate 5 [Comp 16384 128])
```