

# Csoportok

## Eseményvezérelt alkalmazások

IP-18bEVALKEG | 12

## Eseményvezérelt alkalmazások

IP-18bEVALKEG | 91

## Eseményvezérelt alkalmazások

IP-18bEVALKEG | 92

### 4. házi feladat

Kategória:	Házi feladatok
Elérhető:	2022. 10. 06. 23:59
Pótolható határidő:	2022. 10. 14. 23:59
Végső határidő:	2022. 10. 16. 23:59
Kiírta:	Erdei Zsófia

Leírás:

### Házi feladat

A házi feladatot egy Homework4 nevű modulként kell beadni. Minden definiálandó függvényhez adjuk meg a hozzá tartozó típuszignatúrát is!

**Fontos:** a feltöltött tömörített zip állományban egyetlen .hs kiterjesztésű fájl szerepeljen (*hazi.txt*, *Homework.hs.txt* nem felel meg)!

**Tipp:** Ha szükség van rá, a feladatokat bontsd fel részfeladatokra és definiáljatok segédfüggvényeket!

### Lista dekonstrukció

Adjuk meg azt a függvényt, amely egy rendezett párt állít elő egy listából. A pár első eleme a lista első eleme, a pár második eleme pedig a lista farok része legyen. Feltételezhetjük, hogy a lista nem üres.

headTail :: [a] -> (a, [a])

headTail [1..10] == (1,[2,3,4,5,6,7,8,9,10])

Adjuk meg azt a függvényt, amely kap két listát, és egy rendezett párt állít elő, amelynek első eleme az első lista első eleme, a második eleme a második lista első eleme! Feltételezhetjük, hogy a listák közül egyik sem üres.

doubleHead :: [a] -> [b] -> (a, b)

doubleHead [1,2,3] [3,2,1] == (1,3)

### Rekurzív függvények

Adjuk meg azt a függvényt, amely eldönti egy listáról, hogy szerepel-e benne a 0 !

hasZero :: [Int] -> Bool

hasZero [1..10] == False  
hasZero [1,1,0,1] == True

Adjuk meg azt a függvényt, amely eldönti egy listáról, hogy szerepel-e benne az üres lista!

hasEmpty :: [[a]] -> Bool

hasEmpty [] == False  
hasEmpty [[]] == True  
hasEmpty [[1..10],[],[42]] == True

Adjuk meg azt a függvényt, amely egy lista összes elemét megszorozza kettővel!

doubleAll :: [Int] -> [Int]

doubleAll [] == []  
doubleAll [1..10] == [2,4,6,8,10,12,14,16,18,20]

Definiáljuk újra a last' függvényt, amely visszaadja egy lista utolsó elemét! (A

megoldáshoz *ne* használjátok a Prelude-beli `last` függvényt!)

```
last' :: [a]{-nemüres-} -> a
```

```
last' [1] == 1
last' [1..10] == 10
```

Definiáljuk újra az `init'` függvényt, amely egy lista összes elemét visszaadja, az utolsót kivéve! (A megoldáshoz *ne* használjátok a Prelude-beli `init` függvényt!)

```
init' :: [a]{-nemüres-} -> [a]
```

```
init' [1] == []
init' [1..10] == [1,2,3,4,5,6,7,8,9]
```

Megoldás

 Letöltés

Név:	Homework4.zip
Feltöltés ideje:	2022. 10. 14. 22:44
Értékelés:	
Státusz:	Elfogadva
Feltöltések száma:	2
Értékelte:	Erdei Zsófia
Megjegyzések:	

Automatikus tesztelés eredményei

A megoldás átment a teszteken