

[Табло](#) / [Моите курсове](#) / [Функционално програмиране \(И, КН2, избираем\), зимен семестър 2020/2021](#) / Поправителна сесия 2020/21  
/ [Ликвидационна сесия – практически \(14.09.2021 г.\)](#)

|                       |                                     |
|-----------------------|-------------------------------------|
| <b>Започнат на</b>    | вторник, 14 септември 2021, 09:00   |
| <b>Състояние</b>      | Завършен                            |
| <b>Приключен на</b>   | вторник, 14 септември 2021, 10:45   |
| <b>Изминало време</b> | 1 час 44 мин.                       |
| <b>Оценка</b>         | <b>6,00</b> от 6,00 ( <b>100%</b> ) |

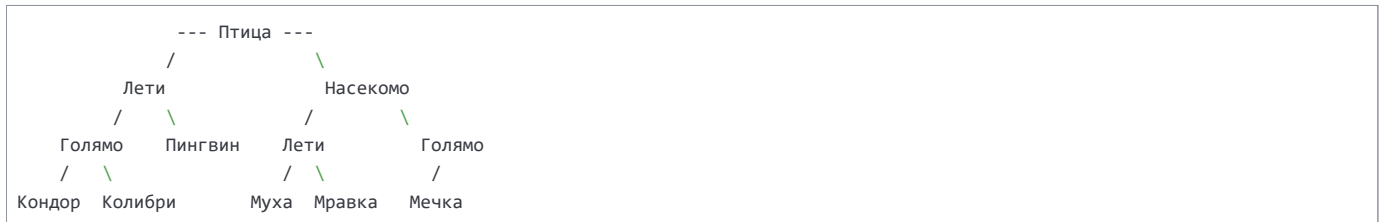
Въпрос 1

Отговорен

6,00 от максимално 6,00 точки

- Решете задачата на езика Scheme.
- Спазвайте добрите практики на функционалния стил.

В задачата ще работим с двоичните дървета познати ви от домашната работа. В едно такова дърво се представя информация за животни и техните свойства. Листата на дървото съдържат непразни символни низове -- имена на животни. Възлите, които не са листа също съдържат непразни низове -- свойства на животните. Вляво от всеки такъв възел се намират животни, които удовлетворяват свойството, а вдясно -- такива, които не го удовлетворяват. По-долу е даден пример за такова дърво. Забележете, че някои от възлите в такова дърво могат да имат само по един наследник.



За да бъде валидно едно такова дърво, трябва да бъдат едновременно изпълнени:

- във всяко листо има точно по едно животно (не може да има листо с две животни в него).
- едно животно не може да се среща едновременно в две различни листа на дървото (например не може да има две различни листа с текст "мечка").
- по пътя от корена към дадено листо не може два пъти да се срещне едно и също свойство.

"Характеристика" на животно представено в дървото ще наричаме асоциативен списък от всички негови свойства. Ключове в този списък ще бъдат свойствата, а с тях ще бъдат свързани булеви стойности, които показват дали свойството е в сила. Например характеристиката на мечката от примерното дърво ще бъде:

```
( ("птица" #f) ("насекомо" #f) ("голямо" #t) )
```

А) Помислете как ще представите в Scheme дърво от така описания вид. В коментар, в началото на файла с решението, опишете това представяне.

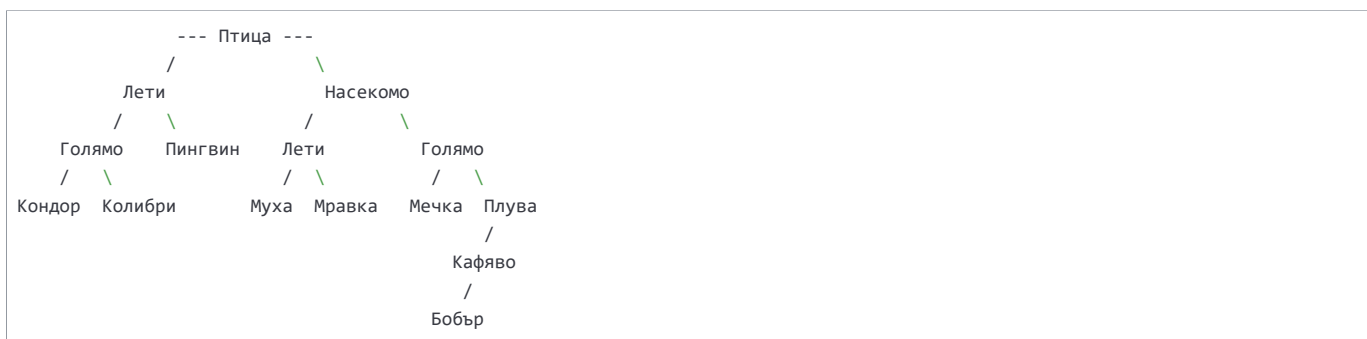
Б) Напишете функция `all-with`, която получава дърво от посочения от вас тип и списък от едно или повече свойства. Функцията да върне списък от имената на всички животни в дървото, които имат тези свойства. Например ако за дървото от примера поискаме всички животни със свойството "голямо", трябва да получим "Кондор" и "Мечка". Ако потърсим за "Птица" и "Лети", трябва да получим "Кондор" и "Колибри". Ако няма такива животни, да се върне празният списък. Например ако потърсим за "Птица", "Лети" и "Бързо", трябва да се върне празният списък. Такъв трябва да се върне и ако потърсим за "Насекомо" и "Голямо".

В) Напишете функция `add-animal`, която ви позволява да добавите ново животно в дървото. Функцията трябва да получи името на животното, което искате да добавите и списък от неговите свойства. Няма ограничение върху реда, в който да се подредят свойствата в него. Функцията трябва да гарантира, че в резултат от операцията отново ще се получи валидно дърво. Ако това не е възможно, да се върне `#f`.

Например, ако към дървото от примера се опитае да добавим животно "бобър" със следните характеристики:

```
( ("плува" #t) ("голямо" #f) ("птица" #f) ("кафяво" #t) ("насекомо" #f) )
```

може да се получи например следното дърво:



Ако опитаме да добавим отново животно "мечка" ще се получи резултат #f.

Ако опитаме да добавим животно "щраус" с характеристики (( "птица" #t) ("лети" #f)) също ще получим #f.

 [45557.rkt](#)

Коментар:

[◀ Теоретичен тест \(23.08.2021 г.\)](#)

Отиди на ...

[Ликвидационна сесия - теория \(14.09.2021 г.\) ▶](#)