

**Домашна работа № 2 по Функционално програмиране
специалност „Информационни системи“, I курс, I група
2024/2025 учебна година**

Решенията трябва да са готови за автоматично тестване. Важно е програмният код да бъде добре форматиран. Предайте решенията на всички задачи в **един** файл с име `hw2_<FN>.hs`, където `<FN>` е Вашият факултетен номер.

Домашните работи се предават като изпълнение на съответното задание в курса по ФП в Moodle (<https://learn.fmi.uni-sofia.bg/mod/assign/view.php?id=341690>) най-късно до **10.04.2025 г. (четвъртък), 23:55**. Решения няма да могат да се предават след крайния срок. Ще се оценяват само файлове с разширение `hs`. Решения предадени като архив **няма** да се оценяват.

Решенията ще бъдат оценявани след успешна защита, която ще се проведе по време на упражнението на първа група на 11.04.2025 г. (петък).

Приятна работа и успех!

Вашата задача е да добавите функционалност към система за управление на зоопарк, реализирана на Haskell. В зоопарка се управляват животни и техните хранителни нужди. Всяко животно се описва със следните характеристики: име на вида (низ), уникален идентификатор (цяло число), тип на хранене (растителноядно, месоядно или всеядно), ниво на глад (цяло число).

Типът `Diet` представя типа на хранене (диетата) на животните, а типът `Animal` - животните.

```
type Species = String
type AnimalId = Int
data Diet = Herbivore | Carnivore | Omnivore deriving (Show, Eq)
data Animal = Animal { species :: Species
                      , animalId :: AnimalId
                      , diet :: Diet
                      , hungerLevel :: Int
                      } deriving Eq
```

Да се дефинират на езика Haskell:

1. екземпляр на класа `Show` за типа `Animal`, чрез който на всяка стойност от вида `(Animal <s> <id> <d> <h>)` да се съпоставя низ от вида `"<s>(<id>,<d>,<h>)"`.
2. функция `updateDiet :: [Animal] -> AnimalId -> Diet -> [Animal]`, която позволява промяна на диетата на животно по неговия идентификатор. Функцията получава списък с всички животни, идентификатор на животно и

конструктор на типа `Diet`. Функцията връща списък със животни. **За решението да се използва** `List Comprehension`.

3. функция `feedAnimal :: [Animal] -> Int -> AnimalId -> [Animal]`, която получава списък от животни `zoo`, число `k` и идентификатор на животно `id`. Функцията трябва да върне нов списък с всички животни, но в който към нивото на глада на животното с идентификатор `id` в `zoo` е добавено числото `k`. **За решението да се използва** `map`.
4. функция `feedIfHungry :: [Animal] -> Int -> [AnimalId] -> [Animal]`, която получава списък със животни `zoo`, число `k` и списък с идентификатори на животни `ids`. Функцията трябва да върне нов списък с всички животни, но в който към нивото на глада на всички *гладни* животни с идентификатори от `ids` е добавена стойността `k`. *Гладни* са тези животни с ниво на глада по-малко от 5. **За решението да се използва** `foldr`.
5. функция `findAnimalById :: [Animal] -> AnimalId -> Maybe Animal`, която позволява търсене на животно по неговия уникален идентификатор. Функцията трябва да връща `Nothing`, ако животното не е намерено или да върне *подходяща стойност от тип* `Maybe Animal`, ако е намерено. **За решението да се използва рекурсия**.

Примери:

```
zoo = [Animal "Lion" 1 Carnivore 3, Animal "Elephant" 2 Herbivore 3,
       Animal "Bear" 3 Omnivore 6, Animal "Turtle" 4 Herbivore 2]

show (Animal "Lion" 1 Carnivore 3) -> "Lion(1,Carnivore,3)"

updateDiet zoo 3 Herbivore -> [Lion(1,Carnivore,3),
                               Elephant(2,Herbivore,3),Bear(3,Herbivore,6),Turtle(4,Herbivore,2)]

feedAnimal zoo 3 1 -> [Lion(1,Carnivore,6),
                      Elephant(2,Herbivore,3),Bear(3,Omnivore,6),Turtle(4,Herbivore,2)]

feedIfHungry zoo 3 [1,2,3,4] -> [Lion(1,Carnivore,6),
                                 Elephant(2,Herbivore,6),Bear(3,Omnivore,6),Turtle(4,Herbivore,5)]

feedIfHungry zoo 3 [3] -> [Lion(1,Carnivore,3),
                           Elephant(2,Herbivore,3),Bear(3,Omnivore,6),Turtle(4,Herbivore,2)]

findAnimalById zoo 10 -> Nothing
```