

Τρίτη εργαστηριακή άσκηση στο μάθημα Αρχές Γλωσσών Προγραμματισμού

Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Πανεπιστημίου Ιωαννίνων
Γκόγκος Χρήστος

Άρτα, 26 Απριλίου 2025

Εισαγωγή

Η εργαστηριακή άσκηση εξετάζει βασικές έννοιες του συναρτησιακού προγραμματισμού με την Haskell όπως οι συναρτήσεις, η αναδρομή, το ταίριασμα προτύπων, οι συναρτήσεις υψηλής τάξης, τα comprehensions και το currying.

Επιπλέον, ζητείται η υλοποίηση ισοδύναμου κώδικα στη Python, έτσι ώστε αφενός να γίνει κατανοητή η διαφορετική προσέγγιση που ακολουθούν οι δυο γλώσσες, αλλά και το ότι ιδέες από το συναρτησιακό προγραμματισμό έχουν επηρεάσει και χρησιμοποιούνται σε δημοφιλείς προστακτικές, κατά βάση γλώσσες, όπως η Python.

Προετοιμασία για την εργασία

Εγκαταστήστε την Haskell ακολουθώντας της οδηγίες που θα εντοπίσετε στο <https://www.haskell.org/get-started/> για το GHCup. Στη συνέχεια εγκαταστήστε με το cabal τη βιβλιοθήκη HUnit με την ακόλουθη εντολή:

```
$ cabal install --lib HUnit
```

Ερωτήματα στην Haskell 70%

Σε ένα αρχείο με όνομα Assignment.hs και αρχικά περιεχόμενα τα ακόλουθα:

```
module Assignment where
```

```
-- Ερώτημα 1: Υπολογισμός εμβαδού κύκλου
```

```
piVal :: Double
```

```
...
```

```
circleArea :: Double -> Double
```

```
...
```

```
-- Ερώτημα 2: Παραγοντικό
```

```
factorial :: Integer -> Integer
```

```
...
```

```
-- Ερώτημα 3: Υπολογισμός μήκους λίστας (χωρίς length)
```

```
myLength :: [a] -> Int
```

```
...
```

```
-- Ερώτημα 4: Εφαρμογή ανώνυμης συνάρτησης σε λίστα
```

```
applyToList :: (Int -> Int) -> [Int] -> [Int]
```

```
....
```

```

-- Ερώτημα 5: Αθροισμα ζεύγους τιμών
addPair :: (Int, Int) -> Int
...

-- Ερώτημα 6: Τετράγωνα άρτιων τιμών
squaredEvens :: [Int] -> [Int]
...

-- Ερώτημα 7: Τετράγωνα άρτιων τιμών μέχρι μια τιμή (με comprehension)
evenSquares :: Int -> [Int]
...

-- Ερώτημα 8: Μερική αποτίμηση
greaterThan :: Int -> [Int] -> [Int]
greaterThan n xs = filter (> n) xs

-- Μερικώς εφαρμοσμένες συναρτήσεις
gt5 :: [Int] -> [Int]
...

gt10 :: [Int] -> [Int]
...

```

1: Το αρχείο Assignment.hs

συμπληρώστε τον απαιτούμενο κώδικα, στα σημεία όπου υπάρχουν 3 τελείες, έτσι ώστε τα ακόλουθα unit tests να επιτυγχάνουν:

```

import Assignment
import Test.HUnit

tests :: Test
tests = TestList
  [ "piVal" ~: piVal ~?= 3.14159
  , "circleArea" ~: abs (circleArea 5 - 78.53975) < 1e-4 ~?= True
  , "factorial" ~: factorial 5 ~?= 120
  , "myLength" ~: myLength [1,2,3,4] ~?= 4
  , "applyToList" ~: applyToList (*2) [1,2,3] ~?= [2,4,6]
  , "addPair" ~: addPair (3,4) ~?= 7
  , "squaredEvens" ~: squaredEvens [1..5] ~?= [4, 16]
  , "evenSquares" ~: evenSquares 10 ~?= [4,16,36,64,100]
  , "gt5" ~: gt5 [1..10] ~?= [6,7,8,9,10]
  , "gt10" ~: gt10 [1..20] ~?= [11..20]
  ]

main :: IO ()
main = do
  _ <- runTestTT tests
  return ()

```

2: Το αρχείο AssignmentTest.hs

δηλαδή να εμφανίζονται τα ακόλουθα αποτελέσματα:

```

$ runghc AssignmentTest.hs
Cases: 10   Tried: 10   Errors: 0   Failures: 0

```

Στη συνέχεια παρατίθενται τα επιμέρους ερωτήματα. Παρατηρήστε ότι για κάθε συνάρτηση δίνεται ο τύπος της, ενώ εσείς πρέπει να αντικαταστήσετε τις 3 τελείες σε κάθε περίπτωση με κατάλληλο κώδικα.

Ερώτημα 1

Να οριστεί η σταθερά `piVal` στην τιμή 3.14159 και η συνάρτηση `circleArea`, που να δέχεται την ακτίνα ενός κύκλου ως πραγματικό αριθμό και να επιστρέφει το εμβαδόν του κύκλου (χρησιμοποιώντας τον τύπο $E = \pi r^2$).

Ερώτημα 2

Να οριστεί η συνάρτηση `factorial` που να δέχεται ως όρισμα έναν μη αρνητικό ακέραιο και να επιστρέφει το παραγοντικό του. Η λύση να υλοποιηθεί με `pattern matching`.

Ερώτημα 3

Να οριστεί η συνάρτηση `myLength`, που να δέχεται μια λίστα και να επιστρέφει το πλήθος των στοιχείων της, χωρίς να χρησιμοποιηθεί η συνάρτηση `length`.

Ερώτημα 4

Να οριστεί η συνάρτηση ανώτερης τάξης `applyToList` που να δέχεται ως ορίσματα μια συνάρτηση και μια λίστα ακεραίων και να εφαρμόζει τη συνάρτηση σε κάθε στοιχείο της λίστας ακεραίων.

Ερώτημα 5

Να οριστεί η συνάρτηση `addPair`, σύμφωνα με την γραμμή τύπου που δίνεται, που να επιστρέφει το άθροισμα των 2 αριθμητικών τιμών που υπάρχουν στα ορίσματά της.

Ερώτημα 6

Να οριστεί η συνάρτηση `squaredEven` που να δέχεται μια λίστα ακεραίων και να επιστρέφει τα τετράγωνα των άρτιων αριθμών της λίστας. Να χρησιμοποιηθούν οι ενσωματωμένες συναρτήσεις `map` και `filter`.

Ερώτημα 7

Να οριστεί η συνάρτηση `evenSquares` που να δέχεται έναν ακέραιο αριθμό `n` και να επιστρέφει μια λίστα με τα τετράγωνα των άρτιων ακεραίων τιμών από το 1 μέχρι και το `n`. Να χρησιμοποιηθεί `comprehension`.

Ερώτημα 8

Δίνεται η `curried` συνάρτηση `greaterThan` που δέχεται ένα ακέραιο και μια λίστα και να επιστρέφει τα στοιχεία της λίστας που είναι μεγαλύτερα από τον ακέραιο. Να οριστούν οι συναρτήσεις `gt5` και `gt10` ως μερικές εφαρμογές της `greaterThan`, όπου τη θέση του πρώτου ορίσματος της `greaterThan` θα λαμβάνουν οι τιμές 5 και 10 αντίστοιχα.

Ερωτήματα στην Python 30%

Υλοποιήστε τα ερωτήματα που δόθηκαν για την Haskell και στην Python, χρησιμοποιώντας τη βιβλιοθήκη `functools` όπου χρειάζεται. Προσθέστε `unit tests` έτσι ώστε να υπάρχουν οι ίδιοι έλεγχοι με τον αντίστοιχο κώδικα σε Haskell. Ονομάστε τα αρχεία `assignment.py` και `assignment_test.py`. Η εκτέλεση των `unit tests` θα πρέπει να γίνεται και να δίνει αποτέλεσμα όπως στη συνέχεια:

```
$ python assignment_test.py
```

```
.....
```

```
-----  
Ran 10 tests in 0.000 s
```

OK

Υποβολή και εξέταση της εργαστηριακής άσκησης

Η προθεσμία υποβολής της εργαστηριακής άσκησης είναι στις 12/5/2025 (Δευτέρα). Στις 13/5/2025 (Τρίτη) οι φοιτητές που θα έχουν υποβάλει την εργαστηριακή άσκηση θα εξεταστούν **γραφτώς** σε θέματα κώδικα παρόμοια με αυτά που χρειάστηκε να αναπτύξουν για την υλοποίηση της εργαστηριακής άσκησης. Η μη συμμετοχή στην εξέταση της εργαστηριακής άσκησης ακυρώνει την υποβολή της. Επιπλέον, ισχύουν τα ακόλουθα:

- Η εργασία μπορεί να υποβληθεί μόνο στο <https://ecourse.uoi.gr/course/view.php?id=1945>.
- Η εργασία είναι ατομική.
- Τα παραδοτέα (σε ένα zip αρχείο με όνομα AM_EPVNYMO_ONOMA_AGP_ERGASIA3.zip, όπου AM είναι ο αριθμός μητρώου του φοιτητή που υποβάλει την άσκηση, ενώ το επώνυμο και το όνομα θα πρέπει να γραφούν με λατινικούς χαρακτήρες) είναι τα ακόλουθα:
 - Κώδικας (το αρχείο Assignment.hs, το αρχείο AssignmentTest.hs, το αρχείο assignment.py και το αρχείο assignment_test.py).
 - Αρχείο README.txt με σύντομες οδηγίες εκτέλεσης.