High-order Functions

Bài tập 1

Cho biết kết quả của biểu thức sau viết bằng Scala:

List(3,5,9,10).foldLeft(0)((x,y)=>x+y)

- 27
- List(27)
- List(3,5,9)
- 3

Chính xác

Chưa đúng, kiểu kết quả cùng kiểu với giá trị trong thông số thứ nhứt của foldLeft, tức là 0, nghĩa là kiểu Int trong trường hợp này.

Chưa đúng

Chưa đúng

Solution

1. Correct Option
2. Wrong
3. Wrong
4. Wrong

Bài tập 2

Cho filter là hàm bậc cao nhận vào một vị từ (predicate - hàm có kết quả kiểu luận lý) và trả về một danh sách con gồm các phần tử thoả vị từ đó và cho phép toán % là phép toán tính modulo. Hãy viết biểu thức để trả về một danh sách con chỉ gồm các giá trị chẵn của danh sách vào? List(2,3,4,5,6)._____ => List(2,4,6)

- List(2,3,4,5,6).filter(x => x % 2)
- List(2,3,4,5,6).filter(x % 2 == 0)
- List(2,3,4,5,6).filter(x => x % 2 == 0)

- List(2,3,4,5,6).filter((x,y)=> x % 2 == y)

Chưa chính xác, hàm filter cần có thông số là một predicate tức một hàm trả về trị luận lý.

Chưa chính xác, hàm filter cần có thông số là một hàm, x % 2 == 0 không phải là một hàm

Chính xác

Chưa chính xác, hàm filter cần thông số là một predicate có 1 thông số.

Solution

1. Wrong
2. Wrong
3. Correct Option
4. Wrong

Bài tập 3

Cho một danh sách List(4,2,6,9), hãy cho biết dùng hàm bậc cao nào để tạo được một danh sách là các giá trị bình phương của các phần tử trong danh sách được cho, tức List(16,4,36,81)

- foldLeft
- map
- filter
- forall

Chưa đúng,

Chính xác, khi danh sách kết quả có cùng số phần tử với danh sách vào và mỗi phần tử kết quả là từ mỗi phần tử của danh sách vào thì map là hàm bậc cao thích hợp nhất

Chưa đúng, hàm này sẽ trả về một danh sách con của danh sách vào

Chưa đúng, hàm này sẽ trả về một giá trị luận lý (true hoặc false)

Solution

1. Wrong
2. Correct Option
3. Wrong
4. Wrong

Bài tập 4

Cho reverse là hàm đảo ngược thứ tự các phần tử của một danh sách (ví dụ List(1,4,2).reverse => List(2,4,1)) và :: dùng để nối 1 phần tử vào đầu một danh sách (a :: List(b,c,d) => List(a,b,c,d)). Với lst đang chứa một danh sách các giá trị nguyên, cho biết biểu thức nào dưới đây có kết quả tương tự reverse?

- lst.map(x => x :: List())
- lst.foldLeft(List())((x,y)=>x::y)
- lst.forAll(x => x::List())
- lst.foldRight(List())((x,y) => x::y)

Chưa chính xác, biểu thức trên sẽ tạo ra một danh sách có các phần tử là các danh sách mà mỗi danh sách có duy nhất 1 phần tử của danh sách ban đầu. Ví dụ List(2,4,1) => List(List(2),List(4),List(1))

Chưa đúng, biểu thức này sẽ gây ra lỗi kiểu vì khi dùng foldLeft thì x sẽ cùng kiểu với thông số thứ nhứt của foldLeft, tức trong trường hợp này là List(), và y sẽ từ các phần tử của lst tức sẽ có kiểu nguyên, khi đó không thể viết x::y được vì thành phần sau :: phải có kiểu là một danh sách.

Chưa đúng, vì forAll cần thông số là một predicate trong khi x::List trả về một danh sách

Chính xác, hãy kiểm tra lại trên Scala.

Solution

1. Wrong
2. Wrong
3. Wrong
4. Correct Option

Bài tập 5

Hãy viết một hàm (forAllExist) nhận vào 3 thông số gồm 1 danh sách các số nguyên và 2 predicate, chỉ trả về kết quả true nếu danh sách có tất cả phần tử thoả predicate thứ nhứt và có ít nhất 1 phần tử thoả predicate thứ hai.

- def forAllExist(lst:List[Int],f1:Int=>Boolean,f2:Int=>Boolean) = lst.forall(f1) && lst.exists(f2)
- def forAllExist(lst:List[Int],f1:Boolean,f2:Boolean) = lst.forall(f1) && lst.exists(f2)
- def forAllExist(lst:List[Int],f1:Int,f2:Int) = lst.forall(f1) && lst.exists(f2)
- def forAllExist(lst:List[Int],f1:Boolean=>Boolean,f2:Boolean=>Boolean) = lst.forall(f1) && lst.exists(f2)

Chính xác

Chưa chính xác, predicate là một hàm nên nó phải có kiểu vào và kiểu ra và được viết dưới hình thức <kiểu vào> => <kiểu ra>. Ngoài ra, predicate phải có kiểu ra là Boolean nên nó phải có dạng <kiểu vào> => Boolean

Chưa chính xác, predicate là một hàm nên nó phải có kiểu vào và kiểu ra và được viết dưới hình thức <kiểu vào> => <kiểu ra>. Ngoài ra, predicate phải có kiểu ra là Boolean nên nó phải có dạng <kiểu vào> => Boolean

Chưa chính xác, theo yêu cầu của câu hỏi, các phần tử của danh sách số nguyên thoả predicate thứ nhứt, do đó, hàm predicate phải có kiểu vào là kiểu phần tử của danh sách (trong trường hợp này là kiểu nguyên).

Solution

1. Correct Option
2. Wrong
3. Wrong
4. Wrong

Bài tập 6

Hãy viết một hàm (doubleCheck) nhận vào 3 thông số gồm 1 danh sách các số nguyên và 2 predicate, chỉ trả về kết quả true nếu danh sách có ít nhất một phần tử thoả predicate thứ nhứt và có ít nhất một phần tử thoả predicate thứ hai.

- def doubleCheck(lst:List[Int],f1:Int=>Boolean,f2:Int=>Boolean) = lst.exists(x => f1(x) && f2(x))
- def doubleCheck(lst:List[Int],f1:Int=>Boolean,f2:Int=>Boolean) = lst.exists(f1) && lst.exists(f2))
- def doubleCheck(lst:List[Int],f1:Int=>Boolean,f2:Int=>Boolean) = lst.forall(x => f1(x) && f2(x))
- def doubleCheck(lst:List[Int],f1:Int=>Boolean,f2:Int=>Boolean) = lst.forall(f1) && lst.forall(f2))

Chưa chính xác, biểu thức này trả về true chỉ khi danh sách có 1 phần tử thoả cùng lúc 2 điều kiện

Chính xác

Chưa chính xác, biểu thức này chỉ trả về true khi tất cả các phần tử của danh sách thoả cả hai predicate

Chưa chính xác, biểu thức này chỉ trả về true khi tất cả các phần tử của danh sách thoả cả hai predicate

Solution

1. Wrong
2. Correct Option
3. Wrong

4. Wrong

Bài tập 7

CHo một hàm được định nghĩa như sau:

def increaseClosures(n:Int)(x:Float) = x + n

Một lênh khai báo biến inc3 được viết như sau:

val inc3 = increaseClosures(3) _

sẽ làm cho inc3 được cất giữ giá trị gì?

- Không cất giữ gì cả vì lệnh gọi hàm increaseClosures bị sai, không đủ thông số
- 3
- x + 3
- Một hàm có kiểu là Float => Float

Chưa chính xác, khi khai báo các thông số ở trong các dấu () độc lập thì hàm có thể được gọi mà không cần đủ số thông số

Chưa chính xác

Chưa chính xác, biến inc3 không cất giữ biểu thức x + 3

Chính xác, hàm này là (x:Float) => x + 3

Solution

1. Wrong
2. Wrong
3. Wrong
4. Correct Option

Bài tập 8

Cho định nghĩa hàm như sau:

def foo(x:Int)(y:Float) = x * y

Khi định nghĩa x như sau: val x = foo(3) _ thì x sẽ có kiểu là gì

- Int
- Float
- Không có kiểu gì cả vì việc gọi hàm foo bị sai do thiếu thông số
- Kiểu hàm: Float => Float

Chưa đúng

Chưa đúng

Chưa đúng

Chính xác, vì hàm foo có kiểu Int => Float => Float nên khi gọi hàm foo với chỉ 1 thông số foo thì sẽ nhận kiểu trả về là Float => Float

Solution

1. Wrong
2. Wrong
3. Wrong
4. Correct Option

Bài tập 9

Cho hàm foo được định nghĩa như sau:

def foo(x:Boolean,y:Float)(z:Float)= if (x) y else z

và m được dịnh nghĩa như sau: val m = foo(true) _ _

Cho biết kiểu của m?

- Float
- Không có kiểu gì cả vì lệnh gọi foo chưa đúng
- Kiểu hàm Float => Float
- Kiểu hàm Float => Float => Float

Chưa đúng

Chính xác

Chưa chính xác

Chưa chính xác

Solution

1. Wrong
2. Correct Option
3. Wrong
4. Wrong