Nội dung thực hành

[I. Khai báo biến và control-flow cơ bản 1](#_Toc14353988)

[II. Class và case class 2](#_Toc14353989)

[III. Function 3](#_Toc14353990)

[IV. Collection 5](#_Toc14353991)

Chúng ta thực hành online trên trang <https://scastie.scala-lang.org/>

# **Khai báo biến và control-flow cơ bản**

1. Khai báo giá trị *a* là *75*, giá trị của *b* là *“Hello world”* rồi in ra màn hình. Bạn hãy thử thay giá trị khác cho *a, b* và sẽ thấy báo lỗi
2. Khai báo 2 biến *x, y* có giá trị là *100* và *“Xin chào”*, rồi in ra màn hình. Bạn hãy thử thay giá trị khác cho *x, y* rồi in lại ra màn hình
3. Dựa vào giá trị *a* đã khai báo ở câu 1, hãy thực hiện yêu cầu:

* Khai báo 1 biến có tên là *result* có giá trị là chuỗi rỗng (sử dụng lệnh *var*)
* Trong trường hợp *a>=85*, thì gán *result = “A”*
* Trường hợp *a>=70*, gán *result = “B”*
* Trường hợp *a>=60*, gán *result = “C”*
* Trường hợp *a>=50*, gán *result = “D”*
* Trường hợp *a<50*, gán *result = “F”*

1. Dựa vào kết quả của biến result ở câu 3, hãy dụng lệnh match thực hiện:

* Trường hợp *result* là “A” hoặc “B” in ra màn hình *“Congratulations”*
* Trường hợp *result* là “C” in ra màn hình: *“You will be better next time”*
* Trường hợp khác, in ra: *“You are failed this time”.*

1. Thay vì dùng *result* là 1 biến, hãy khai báo *result* dạng giá trị (val) và sửa code để vẫn thỏa mãn được yêu câu của bài 3 và 4.

**Đáp án:**

1. val a = 75 hoặc val a:Int = 75

val b = “Hello world” hoặc val b:String = “Hello world”

println(a)

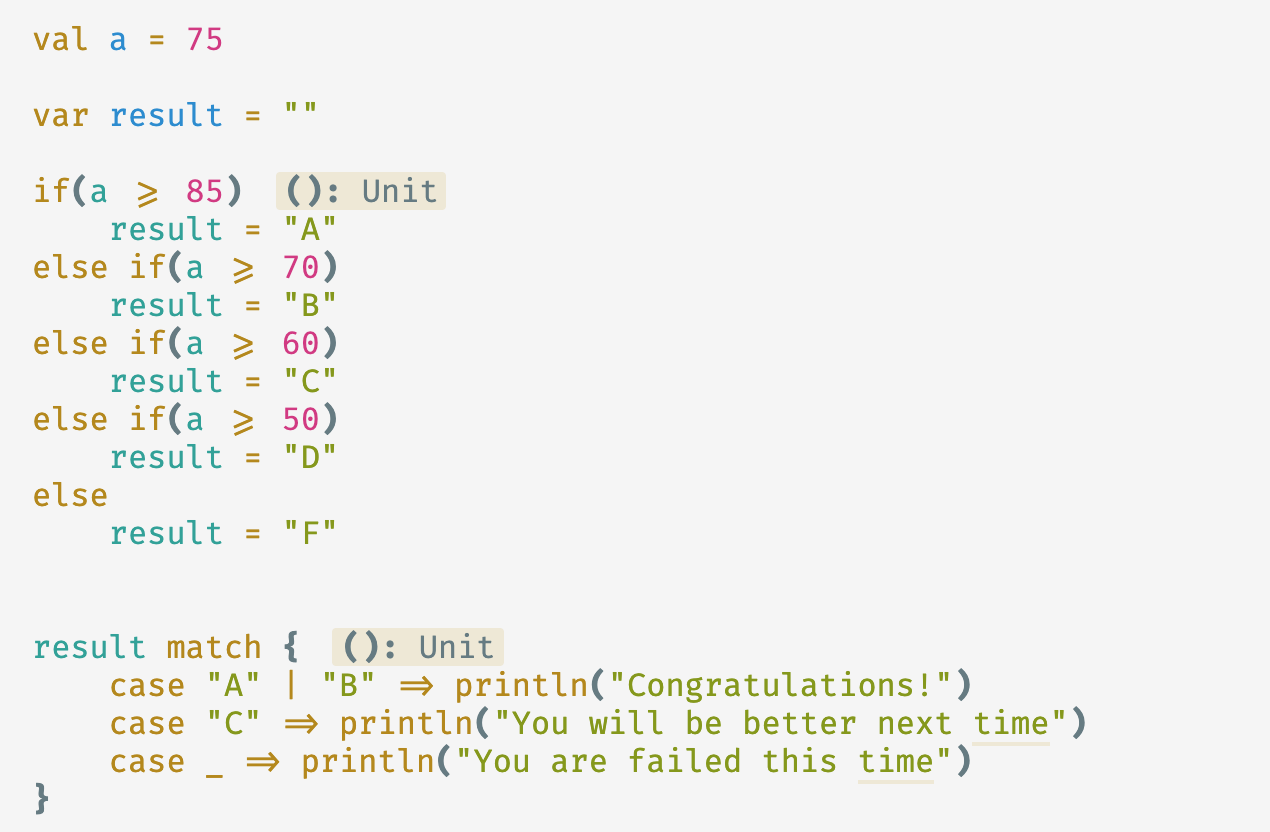
println(b)

1. var x = 100

var y = “Xin chào”

println(x)

println(y)

1. và 4
2. Xem câu trên
3. 

# **Class và case class**

1. Tạo class có tên Person gồm:

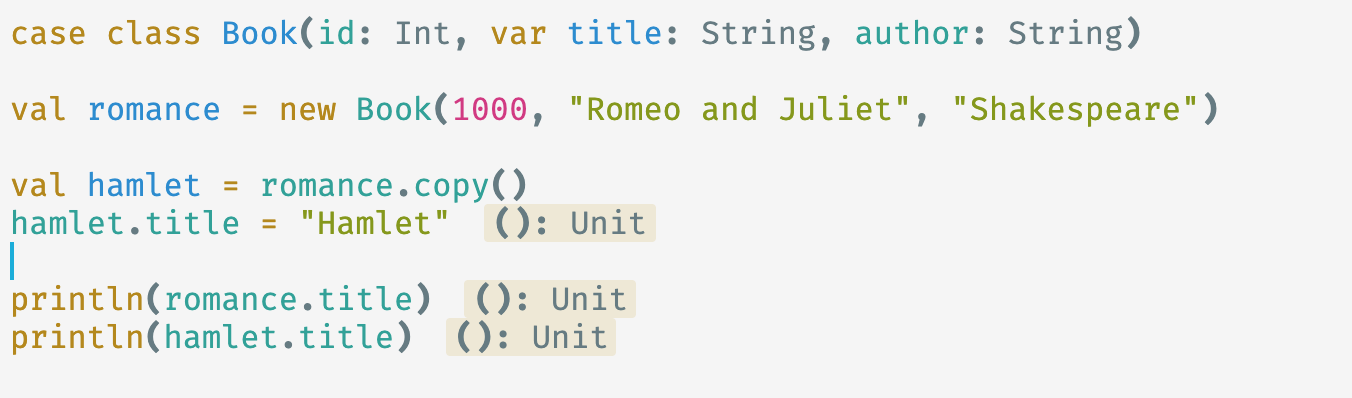
* 2 trường kiểu String là firstName, lastName. Trong đó firstName có thể set lại được.
* Phương thức có tên là hello, in ra màn hình dòng chữ “Xin chào “ + firstName + lastName

Sau đó tạo ra một vài instance của class này và gọi phương thức hello()

1. Tạo case class có tên Book

* 3 trường *id* kiểu *Int*. *title, author: String*, trong đó *title* có thể thay đổi được
* Tạo instance tên *romance* có *id = 1000, title = “Romeo and Juliet”, author = “Shakespeare”*
* Tạo biến *hamlet* bằng cách copy *romance*, sau đó set lại *title* của *hamlet* thành *“Hamlet”*
* In ra màn hình title của romance và hamlet, bạn sẽ thấy 2 title khác nhau

**Đáp án**

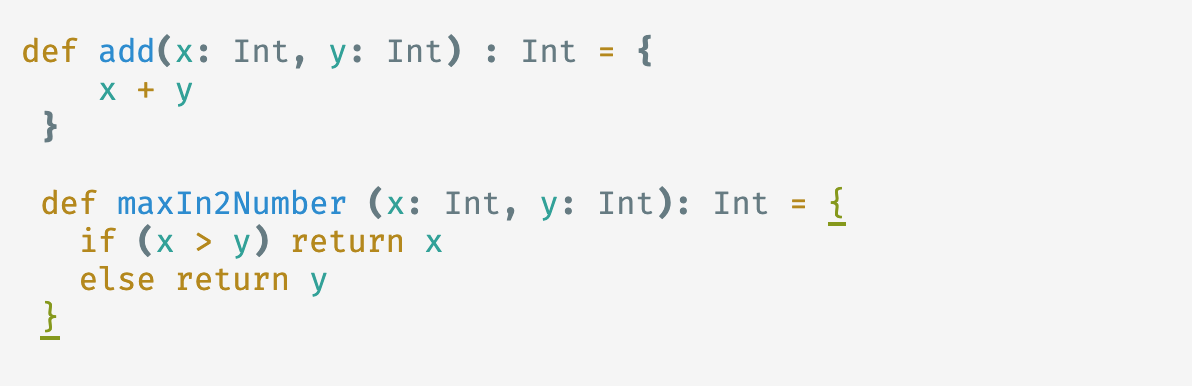
1. 
2. 

# **Function**

1. Viết hàm add để tính tổng 2 số
2. Viết hàm tìm số lớn hơn trong 2 số
3. Tìm cách chuyển 2 hàm trên thành anonymous.
4. Viết hàm getBook, nhận input là kiểu int và trả ra là 1 Tuple gồm 3 trường kiểu Int, String và String. Hàm này kiểm tra nếu input bằng 1 thì trả về id, “Romeo and Juliet”, “Shakespeare”, trường hợp khác thì trả về id, “Not found”, “Not found”. Sau đó gọi hàm trên và truyền giá trị 1 vào hàm và in ra màn hình các giá trị trả về.

**Đáp án:**

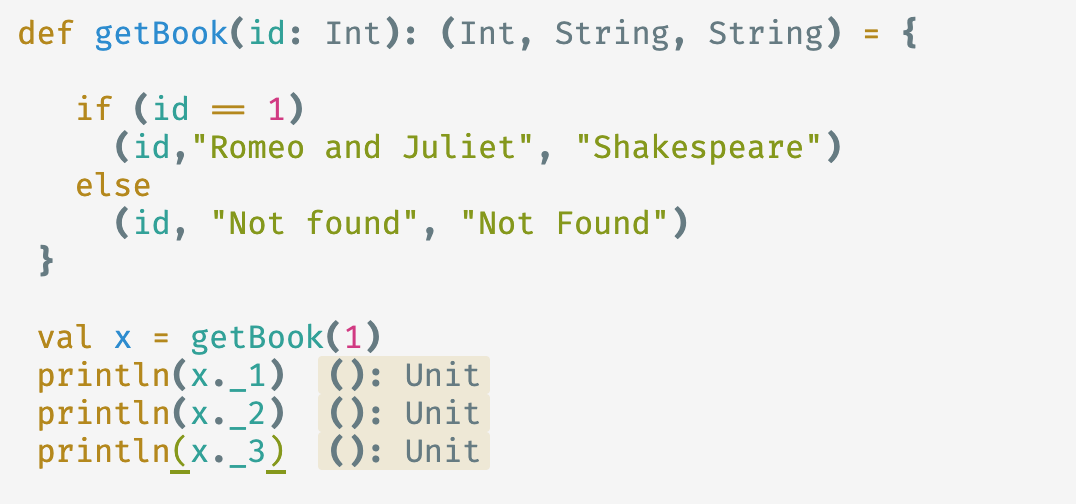
1 &2 :



3:



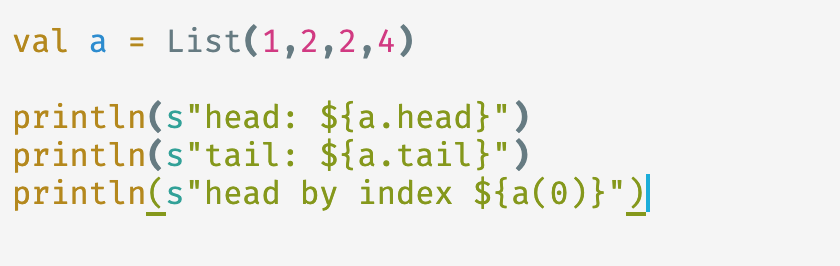
4:

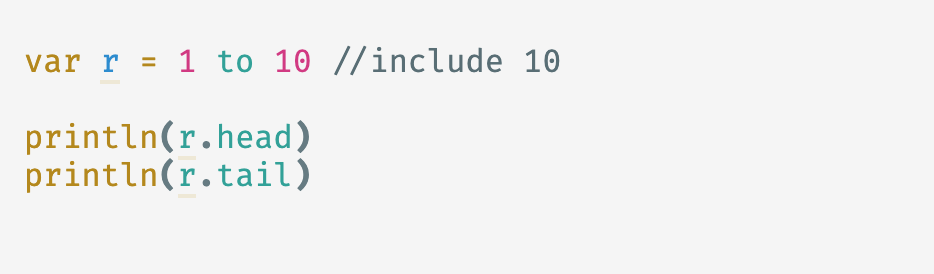


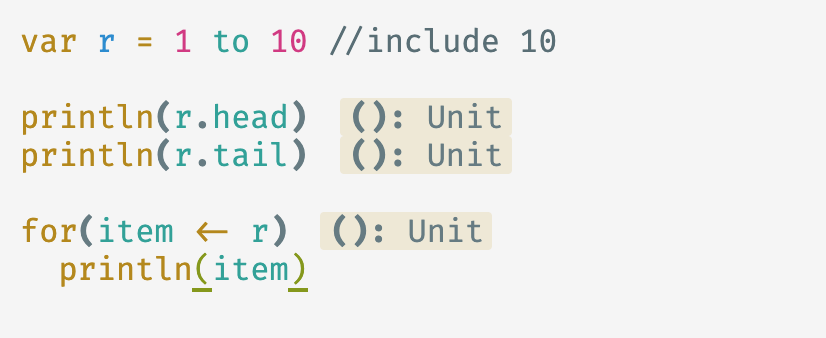
# **Collection**

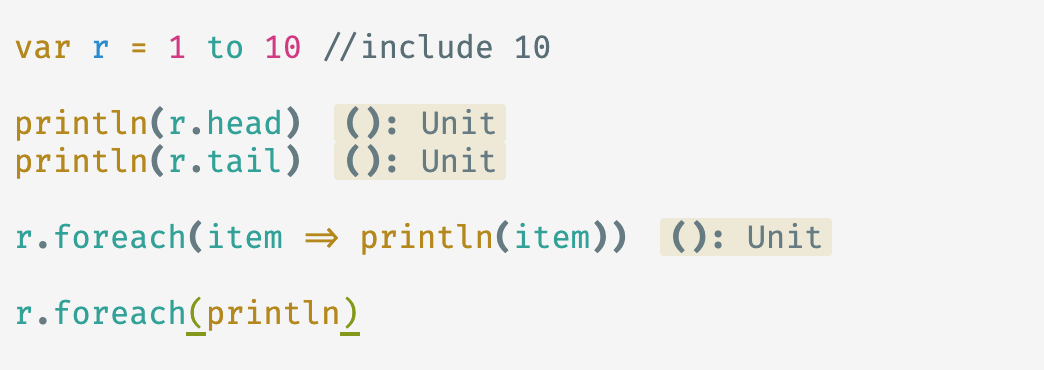
1. Tạo ra list gồm bốn phần từ 1,2,2,4. Sau đó in ra màn hình *head, tail* và phần tử thứ 3 dùng index
2. Dùng Range để tạo tập hợp các số từ 1 đến 10. Sau đó in ra màn hình *head, tail* của danh sách đó.
3. Dùng vòng lặp for để in từng phần tử của list trong câu 2 ra màn hình.
4. Dùng foreach để in từng phần tử của list trong câu 2 ra màn hình.
5. Dùng hàm map để tăng mỗi phần tử trong list ở câu 1 lên 1 đơn vị
6. Dùng hàm reduce để tính tổng các phần tử trong list ở câu 1.
7. Tạo list gồm các phần tử: “the”, “cat”, “sat”, “on”, “the”, “mat”, “the aavark”, “sat”, “on”, “the”, “sofa”. Từ list trên hãy tạo ra 1 list mới mà mỗi phần tử của list cũ sẽ có chuyển thành dạng (item,1)

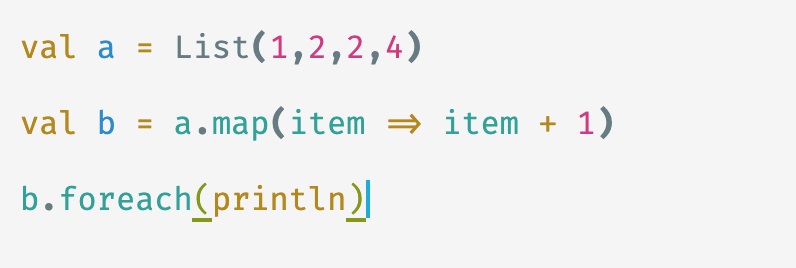
Đáp án:

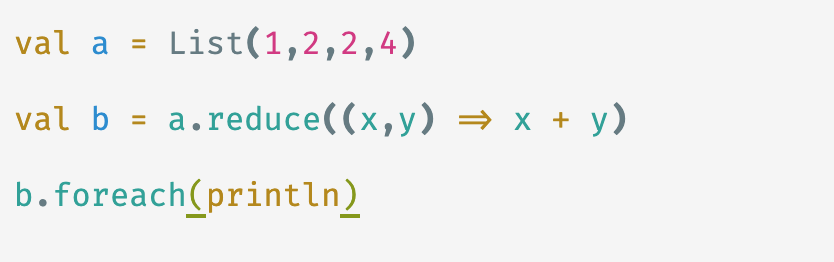


1. 









1. 