First Steps With Lists

Effective Programming in Scala

Julien Richard-Foy

Standard Library

Domain-agnostic data structures and utility functions.

- Collections
- Error management
- Math functions
- Asynchronous execution

Standard Library

Domain-agnostic data structures and utility functions.

- Collections
- Error management
- Math functions
- Asynchronous execution

In this lesson, we will see an overview of the `List` data type: how it can be used to model collections of values, how to construct `List` instances, and some basic manipulation operations.

Address Book Example: Definition

Let us see an example of use of collections to model an address book.

> *An address book contains several contacts.*
> *A contact has a name, an email, and possibly several phone numbers.*

```
case class AddressBook(contacts: List[Contact])
case class Contact(
  name: String,
  email: String,
  phoneNumbers: List[String]
)
```

Collection Types

Collection types are **parameterized** by the type of their elements: for instance, the type `List[Contact]` is the type of a list with elements of type `Contact`, and the type `List[String]` is the type of a list with elements of type `String`.

Consequently, the elements of a list must all have the same type.

Collection Types

Collection types are **parameterized** by the type of their elements: for instance, the type `List[Contact]` is the type of a list with elements of type `Contact`, and the type `List[String]` is the type of a list with elements of type `String`.

Consequently, the elements of a list must all have the same type.

Question: What is the type of a list containing a list of numbers?

Collection Types

Collection types are **parameterized** by the type of their elements: for instance, the type `List[Contact]` is the type of a list with elements of type `Contact`, and the type `List[String]` is the type of a list with elements of type `String`.

Consequently, the elements of a list must all have the same type.

Question: What is the type of a list containing a list of numbers?

The type of a list of numbers is `List[Int]`.

Therefore, the type of a list of list of numbers is `List[List[Int]]`.

Address Book Example: Constructing Lists

With the domain model we have defined, we can construct an address book with two contacts, Alice and Bob:

```
val alice = Contact("Alice", "alice@sca.la", List())
val bob   = Contact("Bob", "bob@sca.la", List("+41787829420"))

val addressBook = AddressBook(List(alice, bob))
```

Constructing Lists

A list having $x_1, \ldots, x_n$ as elements is written `List(`$x_1, \ldots, x_n$`)`

Example

```
val fruits = List("apples", "oranges", "pears")
val nums   = List(1, 2, 3, 4)
val diag3  = List(List(1, 0, 0), List(0, 1, 0), List(0, 0, 1))
val empty  = List()
```

Address Book Example: Basic List Manipulation

```
val numberOfContacts: Int = addressBook.contacts.size
// numberOfContacts = 2

val isAliceInContacts = addressBook.contacts.contains(alice)
// isAliceInContacts = true

val contactNames: List[String] =
  addressBook.contacts.map(contact => contact.name)
// contactNames = List("Alice", "Bob")

val contactsWithPhone: List[Contact] =
  addressBook.contacts.filter(contact => contact.phoneNumbers.nonEmpty)
// contactsWithPhone = List(Contact("Bob", "bob@sca.la", List("+41787829420")))
```

Summary

Lists are one particular data structure modeling a collection of elements.

Lists can be manipulated with high-level operations for transforming or filtering their elements.