Basics of Scala

Vicente De Leon

UID: 2001014594

Assignment 5

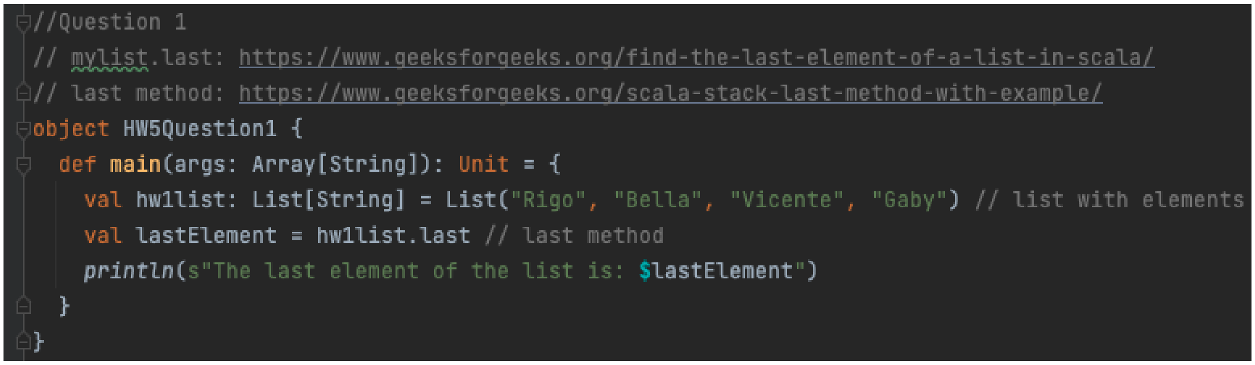
1. Question 1

   Resources:

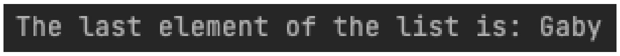
   - Mylist.last: https://www.geeksforgeeks.org/find-the-last-element-of-a-list-in-scala/
   - Last method: https://www.geeksforgeeks.org/scala-stack-last-method-with-example/

   Observations:

   - We create a basic Scala list with elements to run the code.
   - This code is straightforward it uses last method to retrieve the last element of the list.



```scala
//Question 1
// mylist.last: https://www.geeksforgeeks.org/find-the-last-element-of-a-list-in-scala/
// last method: https://www.geeksforgeeks.org/scala-stack-last-method-with-example/
object HW5Question1 {
  def main(args: Array[String]): Unit = {
    val hw1list: List[String] = List("Rigo", "Bella", "Vicente", "Gaby") // list with elements
    val lastElement = hw1list.last // last method
    println(s"The last element of the list is: $lastElement")
  }
}
```



```
The last element of the list is: Gaby
```

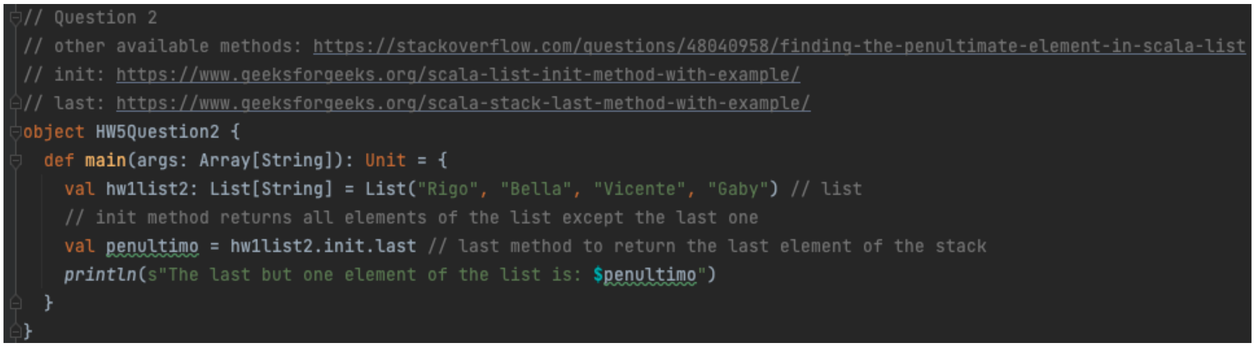
2. Question 2

   Resources:

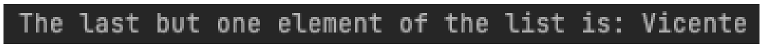
   - Init method(): https://www.geeksforgeeks.org/scala-list-init-method-with-example/
   - Last method(): https://www.geeksforgeeks.org/scala-stack-last-method-with-example/

   Observations:

   - This code is very basic, it uses the init() method to return all the elements of the created list except for the last one. Then we proceed to use the last() method to retrieve the last element (we apply this to the init() method result).



```
// Question 2
// other available methods: https://stackoverflow.com/questions/48040958/finding-the-penultimate-element-in-scala-list
// init: https://www.geeksforgeeks.org/scala-list-init-method-with-example/
// last: https://www.geeksforgeeks.org/scala-stack-last-method-with-example/
object HW5Question2 {
  def main(args: Array[String]): Unit = {
    val hw1list2: List[String] = List("Rigo", "Bella", "Vicente", "Gaby") // list
    // init method returns all elements of the list except the last one
    val penultimo = hw1list2.init.last // last method to return the last element of the stack
    println(s"The last but one element of the list is: $penultimo")
  }
}
```



```
The last but one element of the list is: Vicente
```

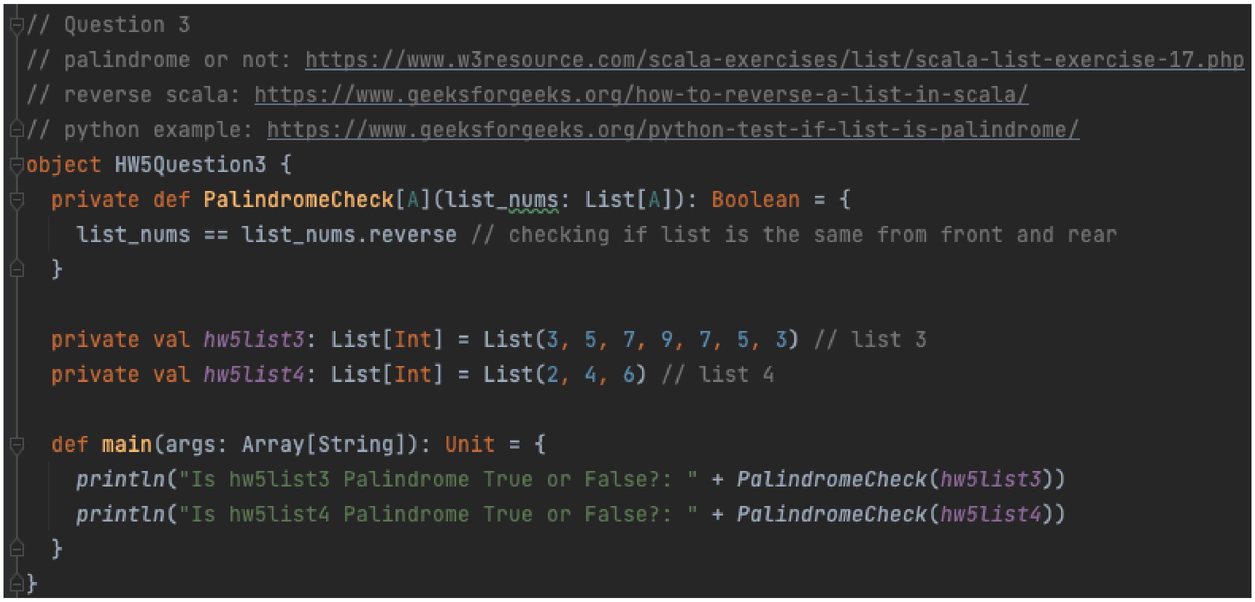
3. Question 3

Resources:

- Palindrome: https://www.w3resource.com/scala-exercises/list/scala-list-exercise-17.php
- Reverse: https://www.geeksforgeeks.org/how-to-reverse-a-list-in-scala/
- Python palindrome: https://www.geeksforgeeks.org/python-test-if-list-is-palindrome/

Observations:

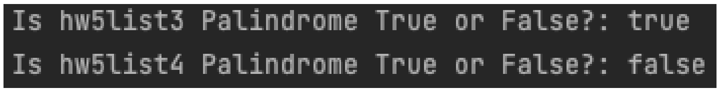
- This code comes from the first resource ("Palindrome"). It also very basic, it takes the created lists and returns Boolean values True for palindrome and False for not palindrome. We apply the reverse function to reverse the created lists.



```scala
// Question 3
// palindrome or not: https://www.w3resource.com/scala-exercises/list/scala-list-exercise-17.php
// reverse scala: https://www.geeksforgeeks.org/how-to-reverse-a-list-in-scala/
// python example: https://www.geeksforgeeks.org/python-test-if-list-is-palindrome/
object HW5Question3 {
  private def PalindromeCheck[A](list_nums: List[A]): Boolean = {
    list_nums == list_nums.reverse // checking if list is the same from front and rear
  }

  private val hw5list3: List[Int] = List(3, 5, 7, 9, 7, 5, 3) // list 3
  private val hw5list4: List[Int] = List(2, 4, 6) // list 4

  def main(args: Array[String]): Unit = {
    println("Is hw5list3 Palindrome True or False?: " + PalindromeCheck(hw5list3))
    println("Is hw5list4 Palindrome True or False?: " + PalindromeCheck(hw5list4))
  }
}
```



```
Is hw5list3 Palindrome True or False?: true
Is hw5list4 Palindrome True or False?: false
```

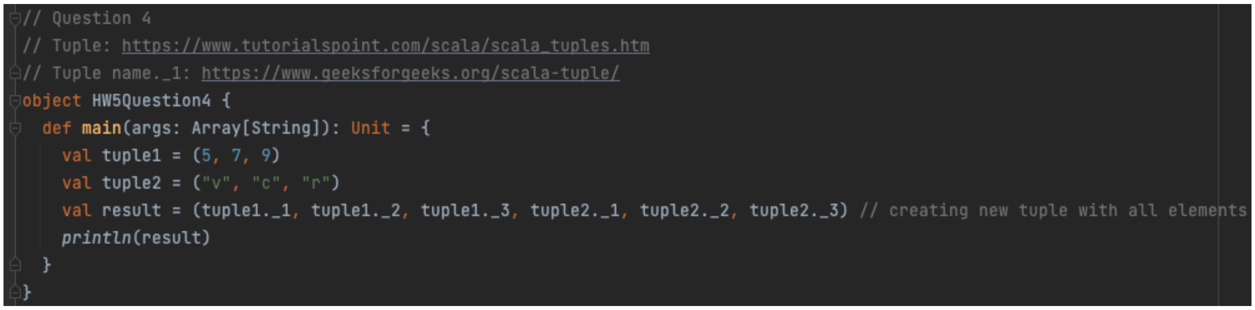
4. Question 4

Resources:

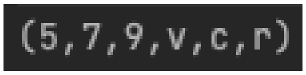
- Tuple: https://www.tutorialspoint.com/scala/scala_tuples.htm
- Name._1: https://www.geeksforgeeks.org/scala-tuple/

Observations:

- We created two tuples to run the code.
- The third tuple, result tuple, is created by using the following syntax: tuple._1, tuple._2 etc This idea came from the second online resource seen above.



```
// Question 4
// Tuple: https://www.tutorialspoint.com/scala/scala_tuples.htm
// Tuple name._1: https://www.geeksforgeeks.org/scala-tuple/
object HW5Question4 {
  def main(args: Array[String]): Unit = {
    val tuple1 = (5, 7, 9)
    val tuple2 = ("v", "c", "r")
    val result = (tuple1._1, tuple1._2, tuple1._3, tuple2._1, tuple2._2, tuple2._3) // creating new tuple with all elements
    println(result)
  }
}
```



```
(5,7,9,v,c,r)
```

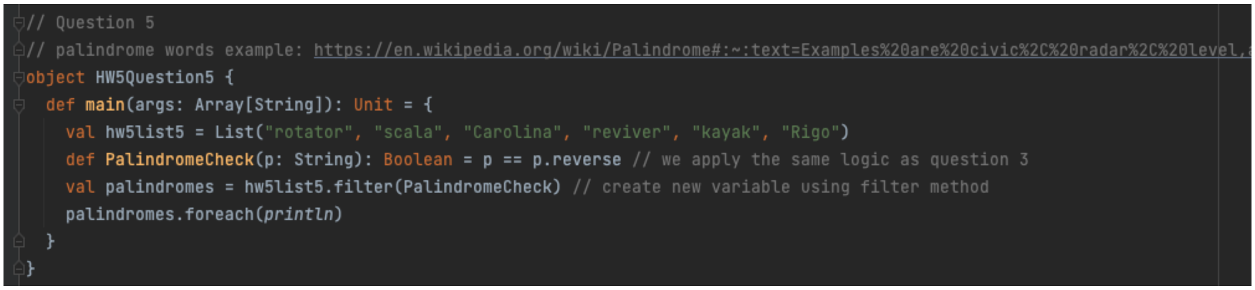
5. Question 5

Resources:

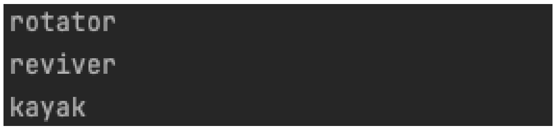
- Palindromes: https://en.wikipedia.org/wiki/Palindrome#:~:text=Examples%20are%20civic%2C%20radar%2C%20level,are%20orders%20of%20magnitude%20rarer.

Observations:

- Few palindrome words were chosen to be part of the list. These words come from Wikipedia resource. This is a simple code that applies the same logic as question 3 by applying reverse to the created list. As a result, we get palindrome words of list5.



```scala
// Question 5
// palindrome words example: https://en.wikipedia.org/wiki/Palindrome#:~:text=Examples%20are%20civic%2C%20radar%2C%20level,
object HW5Question5 {
  def main(args: Array[String]): Unit = {
    val hw5list5 = List("rotator", "scala", "Carolina", "reviver", "kayak", "Rigo")
    def PalindromeCheck(p: String): Boolean = p == p.reverse // we apply the same logic as question 3
    val palindromes = hw5list5.filter(PalindromeCheck) // create new variable using filter method
    palindromes.foreach(println)
  }
}
```



```
rotator
reviver
kayak
```

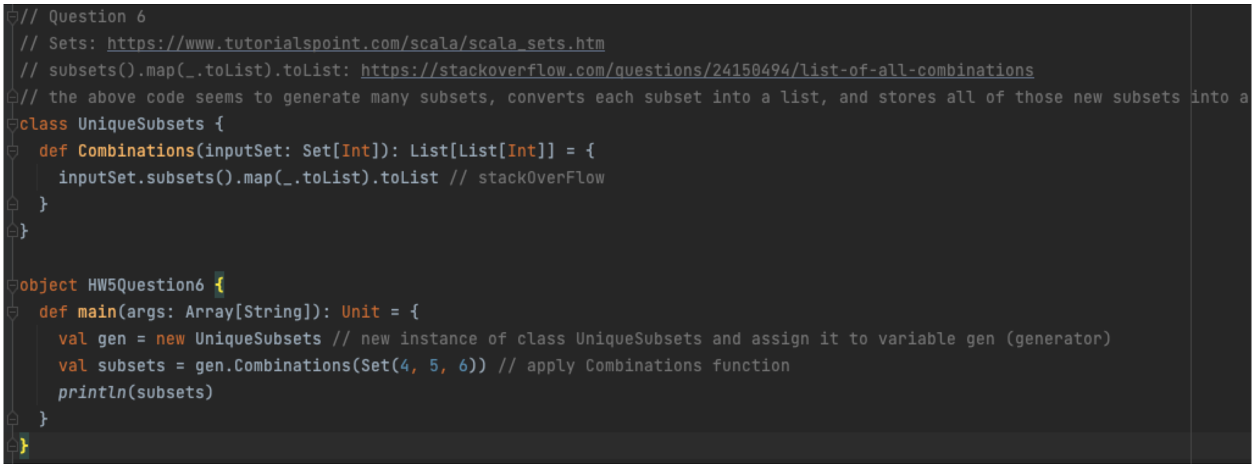
6. Question 6

   Resources:

   - Sets: https://www.tutorialspoint.com/scala/scala_sets.htm
   - Subsets().map(_.toList).toList: https://stackoverflow.com/questions/24150494/list-of-all-combinations.

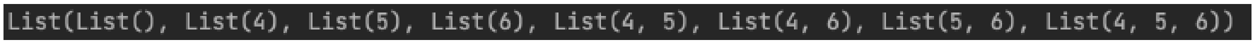
   Observations:

   - Subsets().map(_.toList).toList generates all possible subsets, converts each subset into a list, and stores all of those new subsets into a single list. The idea of using this code came from StackOverFlow. We proceed to create a new instance of the class UniqueSubsets and assign it to variable gen, which is our generator. We apply the Combinations function and store values in new variable named subsets.



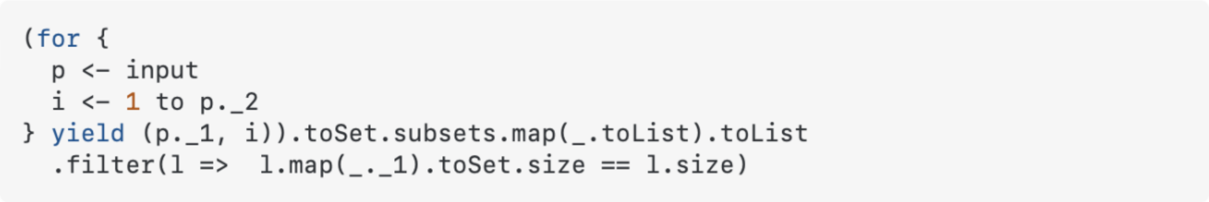
```
// Question 6
// Sets: https://www.tutorialspoint.com/scala/scala_sets.htm
// subsets().map(_.toList).toList: https://stackoverflow.com/questions/24150494/list-of-all-combinations
// the above code seems to generate many subsets, converts each subset into a list, and stores all of those new subsets into a
class UniqueSubsets {
  def Combinations(inputSet: Set[Int]): List[List[Int]] = {
    inputSet.subsets().map(_.toList).toList // stackOverFlow
  }
}

object HW5Question6 {
  def main(args: Array[String]): Unit = {
    val gen = new UniqueSubsets // new instance of class UniqueSubsets and assign it to variable gen (generator)
    val subsets = gen.Combinations(Set(4, 5, 6)) // apply Combinations function
    println(subsets)
  }
}
```



```
List(List(), List(4), List(5), List(6), List(4, 5), List(4, 6), List(5, 6), List(4, 5, 6))
```

Example from resource:



```
(for {
  p <- input
  i <- 1 to p._2
} yield (p._1, i)).toSet.subsets.map(_.toList).toList
  .filter(l =>  l.map(_._1).toSet.size == l.size)
```

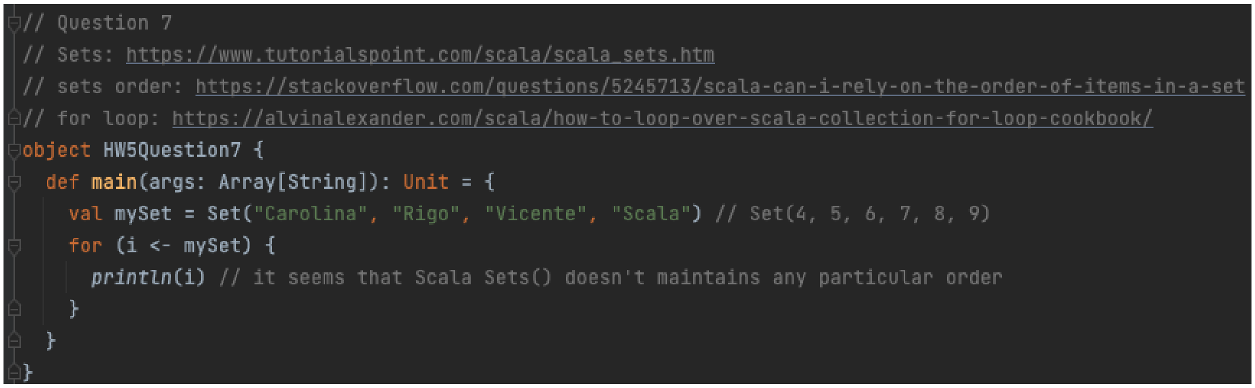
7. Question 7

Resources:

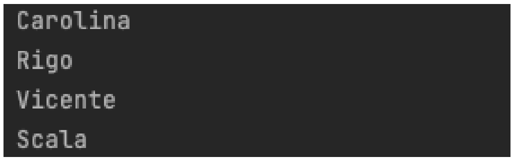
- Sets: https://www.tutorialspoint.com/scala/scala_sets.htm
- Sets order: https://stackoverflow.com/questions/5245713/scala-can-i-rely-on-the-order-of-items-in-a-set
- For loop: https://alvinalexander.com/scala/how-to-loop-over-scala-collection-for-loop-cookbook/

Observations:

- It seems that Scala Set() doesn’t follow any particular order, if we try and run the code using the following example: Set(4 ,5, 6, 7, 8, 9) we will end up getting results with different order. This simple code just uses a for loop over mySet (Set created) to iterate over each element to get results.



```scala
// Question 7
// Sets: https://www.tutorialspoint.com/scala/scala_sets.htm
// sets order: https://stackoverflow.com/questions/5245713/scala-can-i-rely-on-the-order-of-items-in-a-set
// for loop: https://alvinalexander.com/scala/how-to-loop-over-scala-collection-for-loop-cookbook/
object HW5Question7 {
  def main(args: Array[String]): Unit = {
    val mySet = Set("Carolina", "Rigo", "Vicente", "Scala") // Set(4, 5, 6, 7, 8, 9)
    for (i <- mySet) {
      println(i) // it seems that Scala Sets() doesn't maintains any particular order
    }
  }
}
```



```
Carolina
Rigo
Vicente
Scala
```

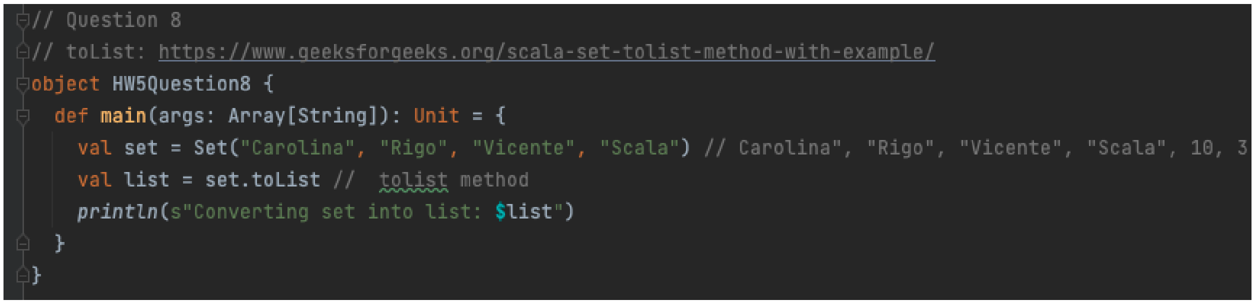
8. Question 8

   Resources:

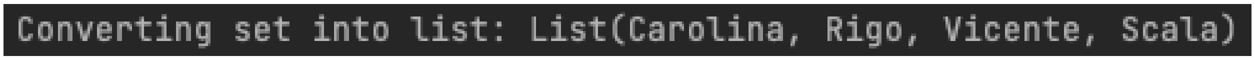
   - Tolist method: https://www.geeksforgeeks.org/scala-set-tolist-method-with-example/

   Observation:

   - In this code we use the tolist method on the created set to convert it into a list. We store the values in new variable named list.



```
// Question 8
// toList: https://www.geeksforgeeks.org/scala-set-tolist-method-with-example/
object HW5Question8 {
  def main(args: Array[String]): Unit = {
    val set = Set("Carolina", "Rigo", "Vicente", "Scala") // Carolina", "Rigo", "Vicente", "Scala", 10, 3
    val list = set.toList //  tolist method
    println(s"Converting set into list: $list")
  }
}
```



```
Converting set into list: List(Carolina, Rigo, Vicente, Scala)
```

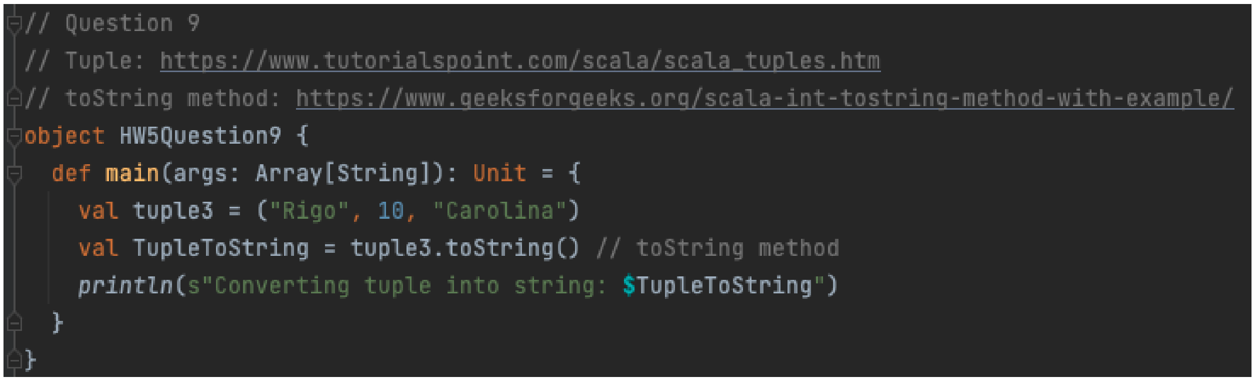
9. Question 9

   Resources:

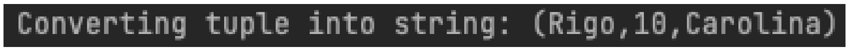
   - Tuple: https://www.tutorialspoint.com/scala/scala_tuples.htm
   - toString method: https://www.geeksforgeeks.org/scala-int-tostring-method-with-example/

   Observations:

   - We apply toString method to tuple3 in order to return the string representation of it. We store these values in the new variable named TupleToString.



```
// Question 9
// Tuple: https://www.tutorialspoint.com/scala/scala_tuples.htm
// toString method: https://www.geeksforgeeks.org/scala-int-tostring-method-with-example/
object HW5Question9 {
  def main(args: Array[String]): Unit = {
    val tuple3 = ("Rigo", 10, "Carolina")
    val TupleToString = tuple3.toString() // toString method
    println(s"Converting tuple into string: $TupleToString")
  }
}
```



```
Converting tuple into string: (Rigo,10,Carolina)
```

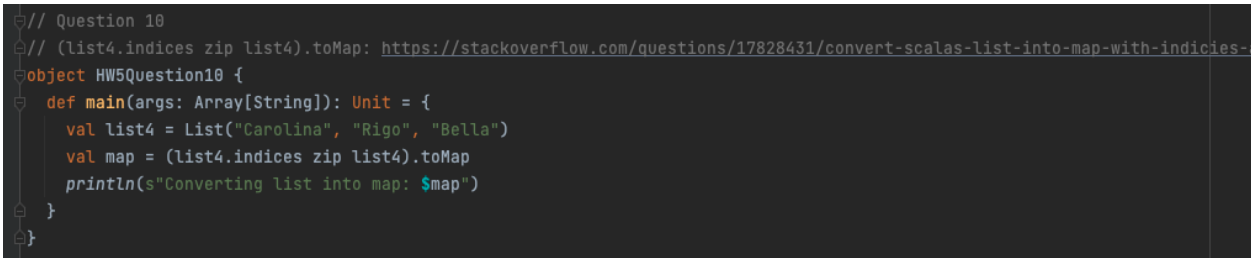
10. Question 10

Resources:

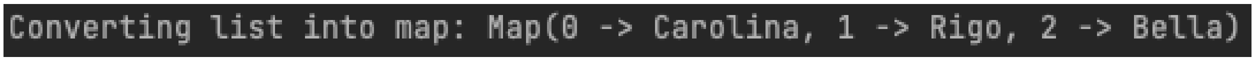
- (xs.indices zip xs).toMap: https://stackoverflow.com/questions/17828431/convert-scalas-list-into-map-with-indicies-as-keys

Observations:

- We are using the StacOverFlow “(xs.indices zip xs).toMap” to convert Scala list into map. We apply the above code to the created list “list4” in order to convert into map as shown in the results below.



```
// Question 10
// (list4.indices zip list4).toMap: https://stackoverflow.com/questions/17828431/convert-scalas-list-into-map-with-indicies-
object HW5Question10 {
  def main(args: Array[String]): Unit = {
    val list4 = List("Carolina", "Rigo", "Bella")
    val map = (list4.indices zip list4).toMap
    println(s"Converting list into map: $map")
  }
}
```



```
Converting list into map: Map(0 -> Carolina, 1 -> Rigo, 2 -> Bella)
```

Codes:

```
import scala.collection.immutable._


//Question 1

object HW5Question1 {

  def main(args: Array[String]): Unit = {

    val hw1list: List[String] = List("Rigo", "Bella", "Vicente", "Gaby") // list with elements

    val lastElement = hw1list.last // last method

    println(s"The last element of the list is: $lastElement")

  }

}


// Question 2

object HW5Question2 {

  def main(args: Array[String]): Unit = {

    val hw1list2: List[String] = List("Rigo", "Bella", "Vicente", "Gaby") // list

    // init method returns all elements of the list except the last one

    val penultimo = hw1list2.init.last // last method to return the last element of the stack

    println(s"The last but one element of the list is: $penultimo")

  }

}
```

```
// Question 3

object HW5Question3 {

  private def PalindromeCheck[A](list_nums: List[A]): Boolean = {

    list_nums == list_nums.reverse // checking if list is the same from front and rear

  }


  private val hw5list3: List[Int] = List(3, 5, 7, 9, 7, 5, 3) // list 3

  private val hw5list4: List[Int] = List(2, 4, 6) // list 4


  def main(args: Array[String]): Unit = {

    println("Is hw5list3 Palindrome True or False?: " + PalindromeCheck(hw5list3))

    println("Is hw5list4 Palindrome True or False?: " + PalindromeCheck(hw5list4))

  }

}


// Question 4

object HW5Question4 {

  def main(args: Array[String]): Unit = {

    val tuple1 = (5, 7, 9)

    val tuple2 = ("v", "c", "r")

    val result = (tuple1._1, tuple1._2, tuple1._3, tuple2._1, tuple2._2, tuple2._3) // creating new tuple with
all elements

    println(result)

  }

}
```

```
// Question 5
object HW5Question5 {
  def main(args: Array[String]): Unit = {
    val hw5list5 = List("rotator", "scala", "Carolina", "reviver", "kayak", "Rigo")
    def PalindromeCheck(p: String): Boolean = p == p.reverse // we apply the same logic as question 3
    val palindromes = hw5list5.filter(PalindromeCheck) // create new variable using filter method
    palindromes.foreach(println)
  }
}


// Question 6
class UniqueSubsets {
  def Combinations(inputSet: Set[Int]): List[List[Int]] = {
    inputSet.subsets().map(_.toList).toList // stackOverFlow
  }
}


object HW5Question6 {
  def main(args: Array[String]): Unit = {
    val gen = new UniqueSubsets // new instance of class UniqueSubsets and assign it to variable gen (generator)
    val subsets = gen.Combinations(Set(4, 5, 6)) // apply Combinations function
    println(subsets)
  }
}
```

```
// Question 7
object HW5Question7 {
  def main(args: Array[String]): Unit = {
    val mySet = Set("Carolina", "Rigo", "Vicente", "Scala") // Set(4, 5, 6, 7, 8, 9)
    for (i <- mySet) {
      println(i) // it seems that Scala Sets() doesn't maintains any particular order
    }
  }
}

// Question 8
object HW5Question8 {
  def main(args: Array[String]): Unit = {
    val set = Set("Carolina", "Rigo", "Vicente", "Scala") // Carolina", "Rigo", "Vicente", "Scala", 10, 3
    val list = set.toList //  tolist method
    println(s"Converting set into list: $list")
  }
}

// Question 9
object HW5Question9 {
  def main(args: Array[String]): Unit = {
    val tuple3 = ("Rigo", 10, "Carolina")
    val TupleToString = tuple3.toString() // toString method
    println(s"Converting tuple into string: $TupleToString")
  }
}
```

```scala
// Question 10
object HW5Question10 {
  def main(args: Array[String]): Unit = {
    val list4 = List("Carolina", "Rigo", "Bella")
    val map = (list4.indices zip list4).toMap
    println(s"Converting list into map: $map")
  }
}
```