

SCALA PROGRAMMING

NAME - KAPAROTU VENKATA SURYA THARANI

USN - 22BTRAD018

BRANCH - AIDE

Ques - Write a Scala program that creates a class called Person with properties like name, age and country. Implement methods to get and set properties.

CODE :

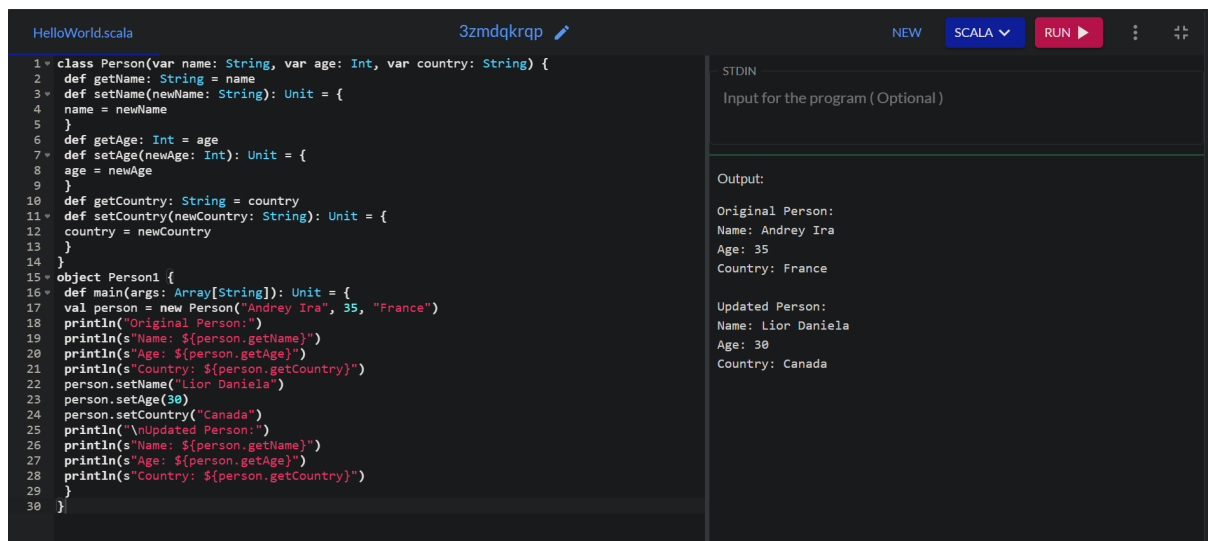
```
class Person(var name: String, var age: Int, var country: String) {
  def getName: String = name
  def setName(newName: String): Unit = {
    name = newName
  }
  def getAge: Int = age
  def setAge(newAge: Int): Unit = {
    age = newAge
  }
  def getCountry: String = country
  def setCountry(newCountry: String): Unit = {
    country = newCountry
  }
}

object Person1 {
  def main(args: Array[String]): Unit = {
    val person = new Person("Andrey Ira", 35, "France")
    println("Original Person:")
    println(s"Name: ${person.getName}")
    println(s"Age: ${person.getAge}")
    println(s"Country: ${person.getCountry}")
    person.setName("Lior Daniela")
    person.setAge(30)
    person.setCountry("Canada")
    println("\nUpdated Person:")
    println(s"Name: ${person.getName}")
    println(s"Age: ${person.getAge}")
    println(s"Country: ${person.getCountry}")
  }
}
```

OUTPUT :

Original Person:
Name: Andrey Ira
Age: 35
Country: France

Updated Person:
Name: Lior Daniela
Age: 30
Country: Canada



The screenshot shows a Scala IDE with a file named 'HelloWorld.scala'. The code defines a 'Person' class with attributes 'name', 'age', and 'country', and methods to get and set these attributes. It also includes a 'main' method that creates a 'Person' object, prints its initial state, updates its name, age, and country, and prints the updated state. The output on the right shows the execution results, matching the text above.

```
1 class Person(var name: String, var age: Int, var country: String) {  
2   def getName: String = name  
3   def setName(newName: String): Unit = {  
4     name = newName  
5   }  
6   def getAge: Int = age  
7   def setAge(newAge: Int): Unit = {  
8     age = newAge  
9   }  
10  def getCountry: String = country  
11  def setCountry(newCountry: String): Unit = {  
12    country = newCountry  
13  }  
14 }  
15 object Person1 {  
16   def main(args: Array[String]): Unit = {  
17     val person = new Person("Andrey Ira", 35, "France")  
18     println("Original Person:")  
19     println(s"Name: ${person.getName}")  
20     println(s"Age: ${person.getAge}")  
21     println(s"Country: ${person.getCountry}")  
22     person.setName("Lior Daniela")  
23     person.setAge(30)  
24     person.setCountry("Canada")  
25     println("\nUpdated Person:")  
26     println(s"Name: ${person.getName}")  
27     println(s"Age: ${person.getAge}")  
28     println(s"Country: ${person.getCountry}")  
29   }  
30 }
```

Output:

Original Person:
Name: Andrey Ira
Age: 35
Country: France

Updated Person:
Name: Lior Daniela
Age: 30
Country: Canada

The following code will take the input from the user regarding the new updated details of the person. And also new property is added i.e Gender.

If the user enters invalid age then it will execute the try catch block to avoid getting errors in the code.

CODE :

```
import scala.io.StdIn
```

```
class Person(var name: String, var age: Int, var country: String, var gender: String) {  
  def getName: String = name  
  def setName(newName: String): Unit = {  
    name = newName  
  }  
  def getAge: Int = age  
  def setAge(newAge: Int): Unit = {
```

```

    age = newAge
  }
  def getCountry: String = country
  def setCountry(newCountry: String): Unit = {
    country = newCountry
  }
  def getGender: String = gender
  def setGender(newGender: String): Unit = {
    gender = newGender
  }
}

object Person1 {
  def main(args: Array[String]): Unit = {
    val name = StdIn.readLine("Enter the name of the person: ")
    val person = new Person(name, 35, "France", "Male")

    println("\nOriginal Person:")
    println(s"Name: ${person.getName}")
    println(s"Age: ${person.getAge}")
    println(s"Country: ${person.getCountry}")
    println(s"Gender: ${person.getGender}")

    val newName = StdIn.readLine("Enter the new name of the person: ")
    person.setName(newName)

    // Use try-catch to handle potential NumberFormatException
    try {
      val newAge = StdIn.readInt()
      person.setAge(newAge)
    } catch {
      case _: NumberFormatException => println("Invalid age input. Age not updated.")
    }

    val newCountry = StdIn.readLine("Enter the new Country for the person: ")
    person.setCountry(newCountry)

    val newGender = StdIn.readLine("Enter the new gender of the person: ")
    person.setGender(newGender)

    println("\nUpdated Person:")
    println(s"Name: ${person.getName}")
    println(s"Age: ${person.getAge}")
    println(s"Country: ${person.getCountry}")
    println(s"Gender: ${person.getGender}")
  }
}

```

```
HelloWorld.scala 3zmzmmc7m NEW SCALA RUN

1 import scala.io.StdIn
2 class Person(var name: String, var age: Int, var country: String, var gender: String) {
3   def getName: String = name
4   def setName(newName: String): Unit = {
5     name = newName
6   }
7   def getAge: Int = age
8   def setAge(newAge: Int): Unit = {
9     age = newAge
10  }
11  def getCountry: String = country
12  def setCountry(newCountry: String): Unit = {
13    country = newCountry
14  }
15  def getGender: String = gender
16  def setGender(newGender: String): Unit = {
17    gender = newGender
18  }
19 }
20
21 object Person1 {
22   def main(args: Array[String]): Unit = {
23     val name = StdIn.readLine("Enter the name of the person: ")
24     val person = new Person(name, 35, "France", "Male")
25
26     println("\nOriginal Person:")
27     println(s"Name: ${person.getName}")
28     println(s"Age: ${person.getAge}")
29     println(s"Country: ${person.getCountry}")
30     println(s"Gender: ${person.getGender}")
31
32     val newName = StdIn.readLine("Enter the new name of the person: ")
33     person.setName(newName)
34
35     // Use try-catch to handle potential NumberFormatException
36     try {
```

STDIN

```
John
Roy
23
```

Output:

```
Enter the name of the person:
Original Person:
Name: John
Age: 35
Country: France
Gender: Male
Enter the new name of the person: Enter the new Country for the person:
Updated Person:
Name: Roy
Age: 23
Country: Syria
Gender: Male
```

```
    try {
      val newAge = StdIn.readInt()
      person.setAge(newAge)
    } catch {
      case _: NumberFormatException => println("Invalid age input. Age not updated.")
    }

    val newCountry = StdIn.readLine("Enter the new Country for the person: ")
    person.setCountry(newCountry)

    val newGender = StdIn.readLine("Enter the new gender of the person: ")
    person.setGender(newGender)

    println("\nUpdated Person:")
    println(s"Name: ${person.getName}")
    println(s"Age: ${person.getAge}")
    println(s"Country: ${person.getCountry}")
    println(s"Gender: ${person.getGender}")
  }
}
```