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

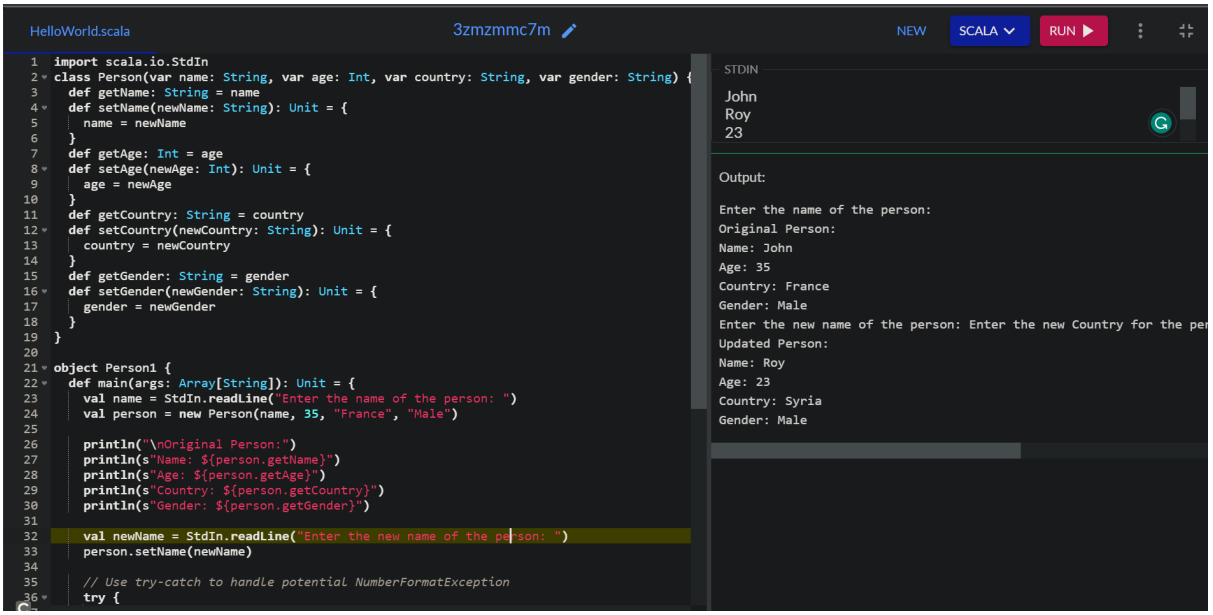
Updated Person:
Name: Lior Daniela
Age: 30
Country: Canada

The 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}")
  }
}
```




```scala
    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}")
  }
}
```