**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Математической Кибернетики и Информационных Технологий

****

**Отчет по лабораторной работе**

по предмету «Функциональное программирование»

Выполнил: студент группы

БВТ1802

Ткачев Александр Владимирович

Руководитель:

Мосева Марина Сергеевна

Москва 2020

Задания к работе указаны в файлах с кодом.

Выполнение:

package exercise4  
  
  
 object TypeClasses {  
  
 // a) Определите тайп-класс Reversable, который представляет в обратном порядке значения.  
  
  
 trait Reversable[T] {  
 def reverse(x: T): T  
 }  
  
 // b) Реализуйте функцию Reverse для String.  
  
 object Reversable {  
 implicit object ReversableString extends Reversable[String] {  
 def reverse(str: String) : String = str.reverse  
 }  
 }  
  
  
 def reverse[T](str: T)(implicit rev: Reversable[T]): T = rev.reverse(str)  
 // примените тайп-класс-решение из пункта (a) здесь  
 def testReversableString(str: String): String = reverse(str)  
  
  
  
 // c) Определите тайп-класс Smash таким образом чтобы в нем была функция smash, которая выполняет операцию со значениями одного типа.  
  
 trait Smash[T] {  
 def smash(a: T, b: T): T  
 }  
  
 object Smash {  
 implicit object SmashInt extends Smash[Int] {  
 def smash(a: Int, b: Int): Int = a + b  
 }  
  
 implicit object SmashDouble extends Smash[Double] {  
 def smash(a: Double, b: Double): Double = a \* b  
 }  
  
 implicit object SmashString extends Smash[String] {  
 def smash(a: String, b:String): String = a.concat(b)  
 }  
 }  
  
 // d) Реализуйте функции Smash для типа Int и Double.  
 // Используйте сложение для типа Int у умножение для типа Double.  
  
 def smash[T](a: T, b: T)(implicit sm : Smash[T]) : T = sm.smash(a, b)  
  
  
 // примените тайп-класс-решение из пункта (d) здесь  
 def testSmashInt(a: Int, b: Int): Int = smash(a, b)  
  
 // примените тайп-класс-решение из пункта (d) здесь  
 def testSmashDouble(a: Double, b: Double): Double = smash(a, b)  
  
  
 // e) Реализуйте функцию Smash для типа String. Необходимо выполнить конкатенацию строк, которые будут получены в качестве параметра.  
  
  
  
 // примените тайп-класс-решение из пункта (d) здесь  
 def testSmashString(a: String, b: String): String = smash(a, b)  
 }

**Вывод**

Тайпклассы являются очень удобным средством для работы одного и того же кода с несколькими типами данных. Данный инструмент позволяет сэкономить на написании очень большого количества кода в тех ситуациях, когда имеется много типов данных, с которыми придется работать.