

/\*Diego Andres Alonzo Medinilla 20172 Seccion 10\*/

/\*TypeAlias\*/

typealias Duration = Long

typealias songList = List<Song>

typealias discList = List<Disc>

/\*Extension Functions\*/

fun songList.getFirstTwoSongs():Pair<Song, Song>{

if (this.count()==2){

return Pair(this[0], this[1])

}

else if (this.count()==1){

return Pair(this[0], Song("Null", 0))

}

else {

var songs = mutableListOf<Int>()

for (k in this){

val num = k.trackNumber

if (num!= null){

songs.add(num)

}

}

songs.sort()

val list1 = listOf(songs[0], songs[1])

val list2 = mutableListOf<Song>()

for (k in list1){

for (j in this){

val num1 = j.trackNumber

if (num1!=null){

list2.add(j)

}

}

}

return Pair(list2[0], list2[1])

}

}

fun Duration.getStringDuration():String{

val S = toInt()

val Secs = S.div(1000)

val min = Secs.div(60)

val Segs = Secs.mod(60)

val Min = min.toInt()

return Min.toString()+":"+Segs.toString()

}

/\*Data Classes

data class ClaseDeAyuda(

val param1: Int // Atributos que forman parte del constructor principal

) {

var param2: Int = 0 // Atributos que NO forman parte del constructor principal

// Sobre-escribir la función toString de la clase

override fun toString(): String {

return "$param1 / $param2"

}

}

\*/

data class Song (

var name : String,

var duration : Duration,

){

var trackNumber : Int? = null

constructor( name: String, duration : Duration, trackNumber : Int ): this( name, duration ){

this.trackNumber = trackNumber

}

override fun toString():String{

if (trackNumber!=null){

return this.trackNumber.toString()+". "+this.name+"-"+this.duration.getStringDuration()

}

return this.name+"-"+this.duration.getStringDuration()

}

}

data class Disc(

var name : String,

var songs : songList

)

{

override fun toString():String{

return this.name+", "+songs.count().toString()+" songs on it"

}

}

data class Artist(

var name : String,

var country : String

)

{

var singles : songList = listOf()

var discography : discList = listOf()

var type : String = ""

constructor(name: String, country: String, discography: List<Disc>, type: String):this( name, country ){

this.discography = discography

this.type = type

}

override fun toString(): String{

if (this.singles.isEmpty()){

return this.name+" is a "+this.type+" from "+this.country+". "+this.type+" has "+this.discography.count()+" discs and no singles"

}

else{

return this.name+" is a "+this.type+" from "+this.country+". "+this.type+" has "+this.discography.count()+" discs and "+this.singles.count()+" singles"

}

}

}

/\*Main\*/

fun main() {

/\*

val myClass = ClaseDeAyuda(10) // Así la instancian

myClass.param2 = 20 // Así acceden a sus atributos que no agregaron en el constructor

println(myClass)

\*/

val song1 = Song("Set Fire To The Rain", 32198)

val song2 = Song("Love me again", 734829, 23)

val song3 = Song("The Rythm of The Weekend", 734829, 1)

val song4 = Song("Dandelions", 873923, 0)

val d : Duration = 1000000

println(d.getStringDuration())

println(song1.toString())

println(song2.toString())

val listSong1 = listOf(song1, song2, song3, song4)

val listSong2 = listOf(song1)

println(listSong2.getFirstTwoSongs())

val disk = Disc("Mylo Xyloto", listSong1)

println(disk)

println(listSong1.getFirstTwoSongs())

val listSong3 = listOf(song1, song4)

println(listSong3.getFirstTwoSongs())

val songs = listOf(

Song(

name = "Whiplash",

duration = 255000L,

trackNumber = 6

),

Song(

name = "Hit the lights",

duration = 215000L,

)

)

val disc = Disc(

name = "Kill'em All",

songs = songs

)

val artist = Artist(

name = "Bad Bunny",

country = "Puerto Rico",

discography = listOf(disc),

type = "Singer"

)

println(songs)

println(artist)

println(disc)

/\*

var songs = mutableListOf<Int>()

\*Asi se crea una lista vacia tipo <T>

\*

\*

for (k in listSong1){

val num = k.trackNumber

if (num!=null){

songs.add(num)

}

}

songs.sort()

if (songs.count()>=2){

val list1 = listOf(songs[0], songs[1])

for (k in list1){

for (j in listSong1){

val num = j.trackNumber

if (num != null && num==k){

println(j)

}

}

}

}

else {

println(songs[0])

}

\*/

}

https://pl.kotl.in/pX93X2s5t?theme=darcula