SML

Alumno: Antonio Ramos Gonzalez

Matricula: 372576

Maestro: Carlos Gallegos

Materia: Paradigmas de la programacion

Fecha de entrega: 30 de mayo de 2024

Introduccion: PARADIGMA FUNCIONAL

La programacion funcional es una manera de programacion que trata la computacion como la evaluacion de funciones matematicas y evita cambiar el estado y los datos mutables. Esto contrasta con paradigmas vistos anteriormente. Por eso para esta actividad se nos pidio realizar un recorrido por el repositorio de SML

Desarrollo:

```
Standard ML of New Jersey (64-bit)
    v110.99.3 [built: Wed Feb 22 13
    :21:40 2023]
- val u = () : unit
val b = true : bool
val i = 1 : int
val iNegative = ~1 : int
val r = 2.0 : real
val rNegative = ~2.0 : real
val s = "s" : string
val c = #"c" : char
```

```
Standard ML of New Jersey (64-bit)
    v110.99.3 [built: Wed Feb 22 13
    :21:40 2023]
- Hello!
Another line!
```

```
Standard ML of New Jersey (64-bit)
    v110.99.3 [built: Wed Feb 22 13
    :21:40 2023]
- val trueCond = 1 : int
val elseCond = ~1 : int
```

```
Standard ML of New Jersey (64-bit)
    v110.99.3 [built: Wed Feb 22 13
    :21:40 2023]
- val t = (1,2) : int * int
val l = [1,2,3] : int list
val r = {name="Zaphod Beeblebrox"
    ,occupation="President of the
    Galaxy"}
  : {name:string, occupation:string}
val tupleField1 = 1 : int
val zaphodsOccuption = "President of
    the Galaxy" : string
val tuplesAreRecords = true : bool
```

```
Standard ML of New Jersey (64-bit)
    v110.99.3 [built: Wed Feb 22 13
    :21:40 2023]
- stdIn:10.6-10.32 Warning: binding
    not exhaustive
          _ :: second :: _ :: nil =
              ...
datatype dog = dog of {name:string}
val n = (1,2,3) : int * int * int
val two = 2 : int
val x = 1 : int
val y = 2 : int
val z = 3 : int
val charlie = dog {name="Charlie"} :
    dog
val lucky = dog {name="Lucky"} : dog
val pup1 = "Lucky" : string
val pup2 = "Charlie" : string
val second = 2 : int
```

```
Standard ML of New Jersey (64-bit)
    v110.99.3 [built: Wed Feb 22 13
    :21:40 2023]
- val add = fn : int -> int -> int
val sub = fn : int -> int -> int
val mul = fn : int * int -> int
val divide = fn : int * int -> int
val divmod = fn : int * int -> int *
    int
val printExample = fn : unit -> unit
val add' = fn : int * int -> int
```

```
Standard ML of New Jersey (64-bit)
    v110.99.3 [built: Wed Feb 22 13
    :21:40 2023]
- val inc = fn : int -> int
val add = fn : int -> int -> int
val inc' = fn : int -> int
val t = true : bool
```

```
Standard ML of New Jersey (64-bit)
    v110.99.3 [built: Wed Feb 22 13
    :21:40 2023]
- Hello.
Valar morghulis.
signature GREETING = sig
  val greeting : string
end
functor Greeter(G: sig
  val greeting : string
end) :
sig
  val greet : unit -> unit
end
structure EnglishGreeting : GREETING
structure ValyrianGreeting :
    GREETING
structure englishGreeter : sig
  val greet : unit -> unit
end
structure essosGreeter : sig
  val greet : unit -> unit
end
val u = () : unit
val u' = () : unit
```

```
Standard ML of New Jersey (64-bit)
    v110.99.3 [built: Wed Feb 22 13
    :21:40 2023]
- val lexpr = 3 : int
```

```
Standard ML of New Jersey (64-bit)
    v110.99.3 [built: Wed Feb 22 13
    :21:40 2023]
- structure Math : sig
  val e : real
end
```

```
Standard ML of New Jersey (64-bit)
    v110.99.3 [built: Wed Feb 22 13
    :21:40 2023]
- val map = fn : ('a -> 'b) -> 'a
    list -> 'b list
val map' = fn : ('a -> 'b) -> 'a
    list -> 'b list
```

```
Standard ML of New Jersey (64-bit)
    v110.99.3 [built: Wed Feb 22 13
    :21:40 2023]
- val sum = fn : int list -> int
val sum_iter = fn : int list -> int
val s = 6 : int
val s' = 6 : int
```

```
Standard ML of New Jersey (64-bit)
    v110.99.3 [built: Wed Feb 22 13
    :21:40 2023]
- infixr 4 +:
datatype 'a list = +: of 'a * 'a
    list | eol
datatype 'a tree = leaf | node of
    {left:'a tree, right:'a tree,
    value:'a}
val ints = 1 +: 2 +: 3 +: eol : int
    list
val inttree =
  node
    {left=node {left=leaf,right=leaf
        ,value=2},
     right=node {left=leaf,right
         =leaf,value=3},value=1} :
         int tree
```

```
Standard ML of New Jersey (64-bit)
    v110.99.3 [built: Wed Feb 22 13
    :21:40 2023]
- Hello, world!
val u = () : unit
```

```
Standard ML of New Jersey (64-bit)
    v110.99.3 [built: Wed Feb 22 13
    :21:40 2023]
- Hello!
val i = <hidden> : int
val j = 10.0 : real
val k = 10 : int
val i' = 11 : int
val i = 10 : int
val iEqK = true : bool
```

Conclusion:

El paradigma funcional es una herramienta poderosa para la programacion. Esta practica me permitio probar la programacion funcionl de primera mano, permitiendome ver su potencial y sus aplicaciones en distintas areas.