

20CYS312 - Principles of Programming Languages

Exploring Programming Paradigms

Assignment-01

Presented by Seran Pandiyan I P

CB.EN.U4CYS21069

TIFAC-CORE in Cyber Security

Amrita Vishwa Vidyapeetham, Coimbatore Campus

Feb 2024



AMRITA
VISHWA VIDYAPEETHAM



- 1 Functional Programming Paradigm
- 2 Functional - Erlang
- 3 Event-Driven Programming Paradigm
- 4 Event Driven - RxJS
- 5 Comparison and Discussion
- 6 Bibliography



Functional Programming Paradigm

Functional programming is a programming paradigm that treats computation as the evaluation of mathematical functions and avoids changing-state and mutable data.

Key concepts and features commonly associated with functional programming:

- Immutable Data
- First-Class and Higher-Order Functions
- Pure Functions
- Recursion
- Lambda Calculus
- Pattern Matching

Languages that follow functional programming paradigm:
JavaScript, Haskell, Scala, F, etc

Real-World Application: WhatsApp (Erlang), VScode (F#)



Implementation in JavaScript

```
// Immutability using a spread operator
const originalArray = [1, 2, 3];
const newArray = [...originalArray, 4]; // Creates a new array with 1, 2, 3, 4

// Pure function example
const square = (x) => x * x; // Always returns the square of the input

// Function composition
const addThenSquare = (x) => square(x + 1); // Composes square with addition

// Higher-order function example
const map = (fn, array) => array.map(fn); // Applies a function to each element
const squaredValues = map(square, [1, 2, 3]); // Returns [1, 4, 9]
```



Functional - Erlang

Erlang is an open-source, functional programming language designed for building scalable and fault-tolerant distributed systems. It was originally developed by Ericsson for telecommunication applications, and it has since gained popularity in various domains, including distributed systems, messaging systems, and concurrent programming.

Some Features of Erlang:

- Concurrency and Distribution
- Fault Tolerance
- Pattern Matching
- Immutable Data

Paradigms Supported: Erlang primarily follows the functional programming paradigm but also incorporates aspects of concurrent and distributed programming.

File Extension: .erl

Current Version: 26.2.1



```
1 % hello world program
2 -module(helloworld).
3 -export([start/0]).
4
5 start() ->
6     io:fwrite("Hello, world!\n").
```



Event-Driven Programming Paradigm

Event-Driven Programming is a paradigm in which the flow of the program is determined by events, such as user actions, sensor outputs, or messages from other programs or threads. The core idea is to respond to events as they occur, rather than following a predefined sequence of actions.

Key concepts and features of logic programming:

- Events
- Event Handlers
- Callback Functions
- Asynchronous Execution
- Event Queue
- Listeners
- State Machines

Languages that follow logic programming paradigm:

Javascript, Python (with frameworks like Tkinter), C#, etc

Real-World Application: Graphical User Interfaces (GUIs) , Web Development



```
class Program
{
    static void Main(string[] args)
    {
        System.Console.WriteLine("Hello World!");
    }
}
```



RxJS is a library for reactive programming using Observables, making it a powerful tool for building asynchronous and event-driven applications in JavaScript.

Some Features of Event Driven - RxJS:

- Operators
- Event Streams
- Declarative Programming
- Asynchronous Programming

Paradigms Supported: reactive programming paradigm, Event-Driven

File Extension: .ts for typescript and .js for javascript

Current Version: 7.8.1



Implementation in RxJS

```
// Import the necessary module  
import { of } from 'rxjs';  
  
// Create an Observable that emits the string "Hello World!"  
const helloWorldObservable = of('Hello World!');  
  
// Subscribe to the Observable and handle its emissions  
helloWorldObservable.subscribe({  
  next: value => console.log(value), // Log the emitted value  
  error: error => console.error(error), // Handle any errors  
  complete: () => console.log('Completed!') // Notify when the Observable completes  
});
```



Comparison and Discussion

- Functional programming focuses on the evaluation of mathematical functions while event driven programming responds to events (user actions, messages, etc.) by triggering actions.
- Functional programming is Well-suited for parallel and concurrent programming, while event driven programming asynchronous nature supports concurrent execution.
- Functional Programming -Declarative style . Event Driven - Reactive Style
- Pure Functional Programming Language: Haskell (Mostly)
- Pure event driven Programming Language: Elm
- Languages which support both programming paradigm: Javascript, Scala, Erlang, Haskell etc.



<https://www.geeksforgeeks.org/functional-programming-paradigm/>

https://erlang.org/documentation/doc-5.6/pdf/programming_examples.pdf

<https://rxjs.dev/>

https://en.wikipedia.org/wiki/Event-driven_programming

