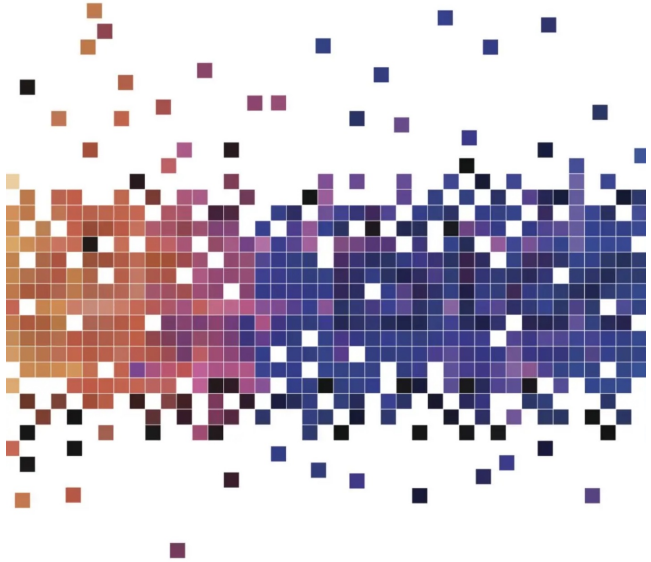
Functional Programming in C#

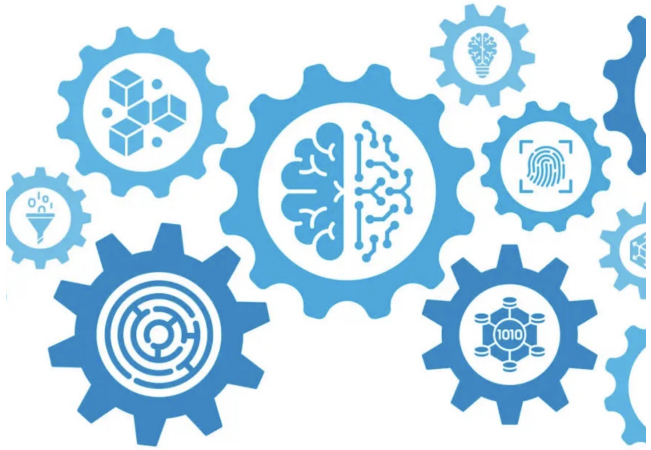
About Functional Programming

- Use of pure functions
- No side effects
- Immutable data structures
- C# is multiparadigm and supports functional programming



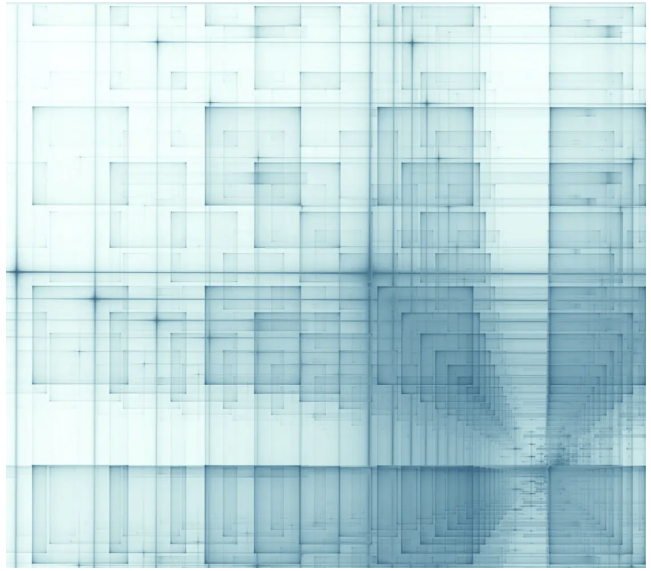
Benefits of Functional C#

- Code reliability and testability
- Efficient, scalable parallelism
- Expressiveness of code



Functional Features in C#

- LINQ - expressive and concise queries and transformations
- Higher-order functions and composition
- Lambda expressions and anonymous functions



Example 1

Computing sum of the squares with LINQ

```
    var sumOfSquares = Enumerable.Range(1,5).
        Select(n => n * n).
        Sum();
```

Example 2

Higher-order function and its usage

```
public static void RunTwice(Action action) {
    action();
    action()
});

RunTwice(() =>
    Console.Out.WriteLine("Hello World"));
```