



UNIVERSIDAD DE LAS FUERZAS ARMADAS "ESPE"

PROG. ORIENTADA A OBJETOS

NRC: 1940

FECHA: 28/08/2024

XXX



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Programming Paradigms

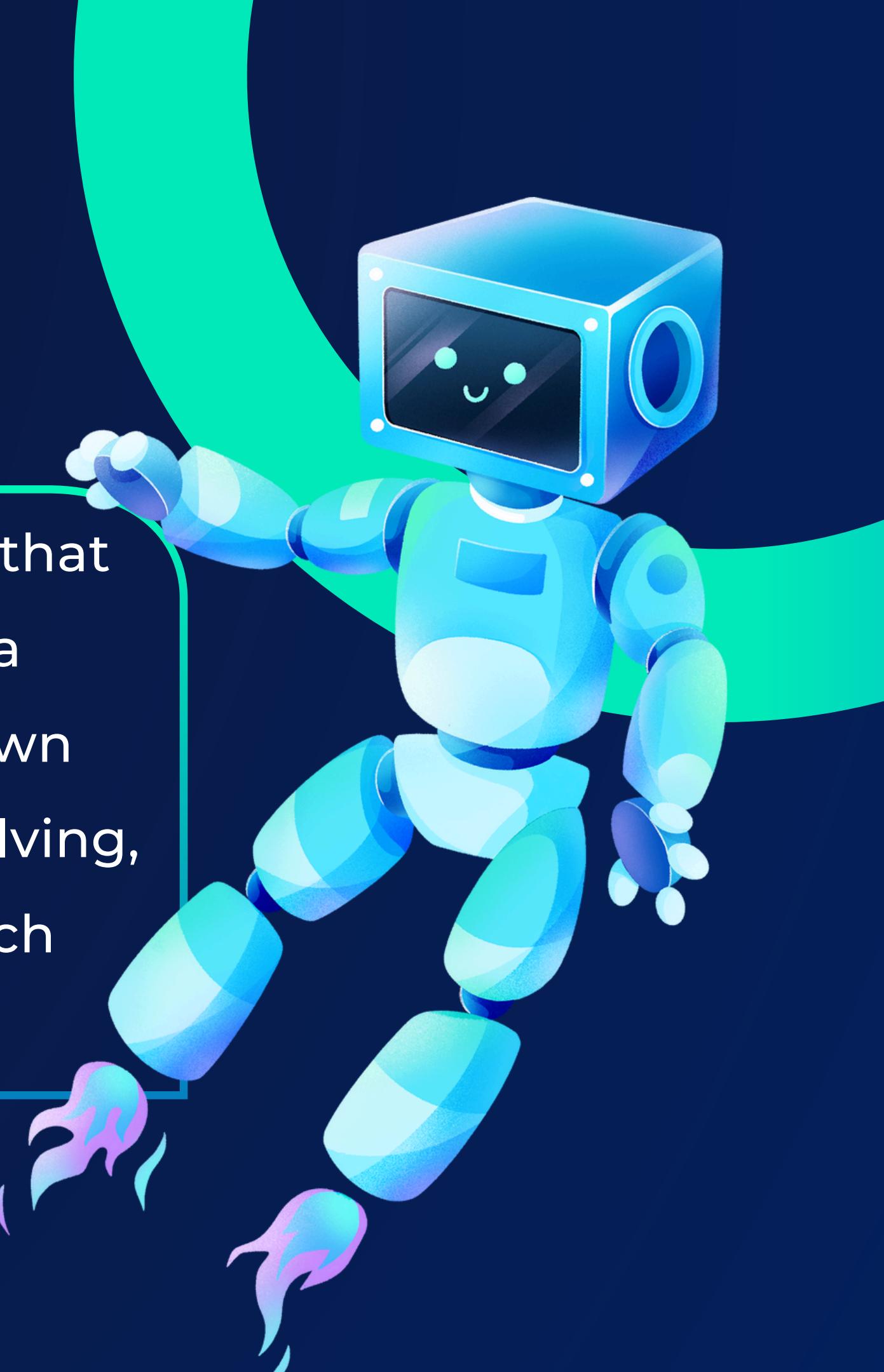
INTEGRANTES:

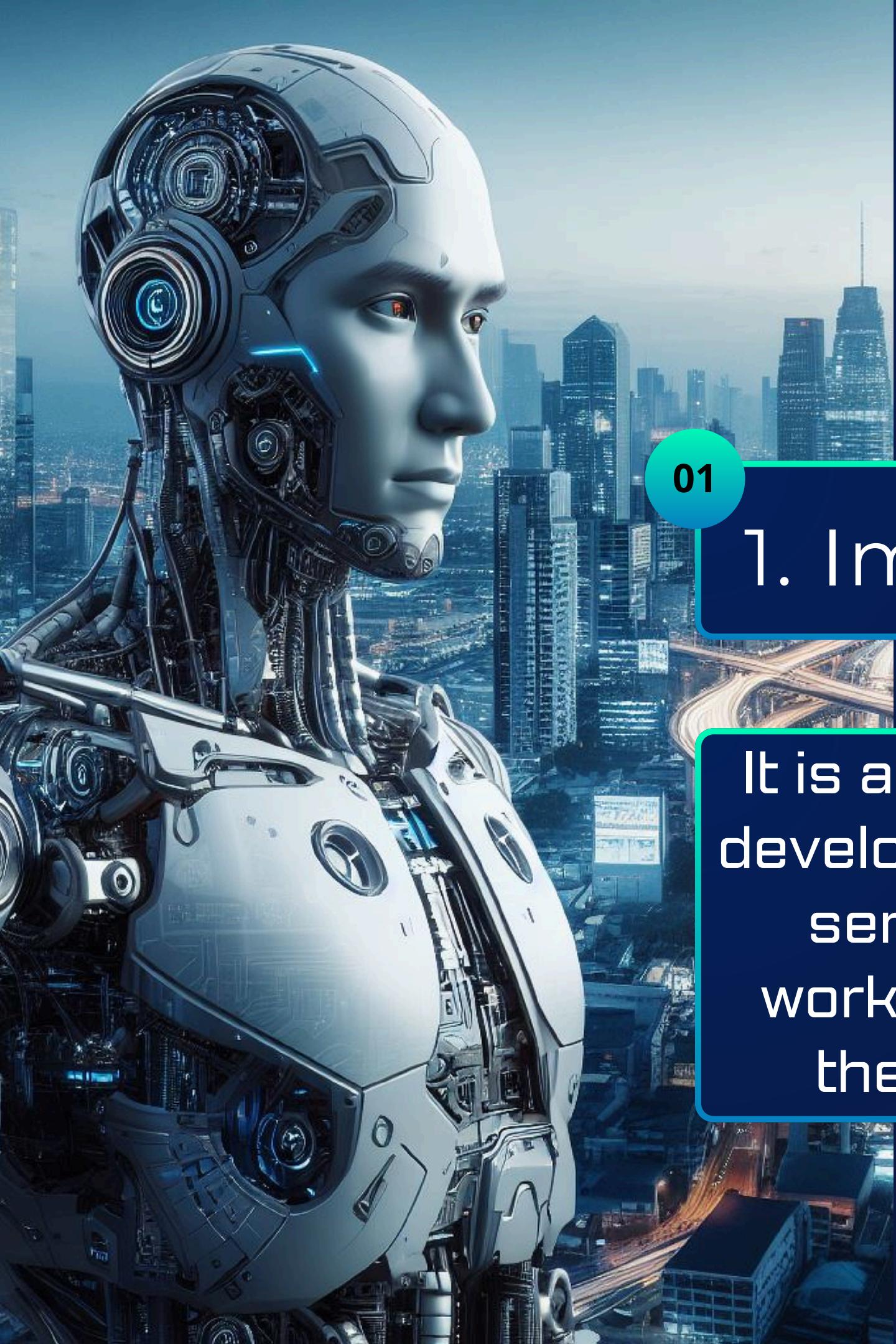
- CHASIPANTA IAN
- CALVOPIÑA DAVID
- CHASIPANTA SAUL
- ROJAS JONATHAN



Programming paradigms

Programming paradigms are approaches or styles that guide how code is structured and organized in a programming language. Each paradigm has its own principles, techniques, and methods for problem-solving, influencing how programmers think and approach software development.





The main paradigms in programming

01

1. Imperative Paradigm

It is a method that allows programs to be developed through procedures. Through a series of instructions, how the code works is explained step by step so that the process is as clear as possible.



Example:

C++ Code:

```
1  /*Implement an algorithm to determine whether  
2   an integer is positive or negative*/  
3  
4  #include <iostream>  
5  
6  using namespace std;  
7  
8  int main() {  
9      int n1;  
10     cout << "Enter an integer number:" << endl;  
11     cin >> n1;  
12  
13    if (n1 >= 0) {  
14        cout << "The number is positive." << endl;  
15    } else {  
16        cout << "The number is negative" << endl;  
17    }  
18  
19    return 0;  
20 }  
21
```



The main paradigms in programming

02

2. Declarative Paradigm

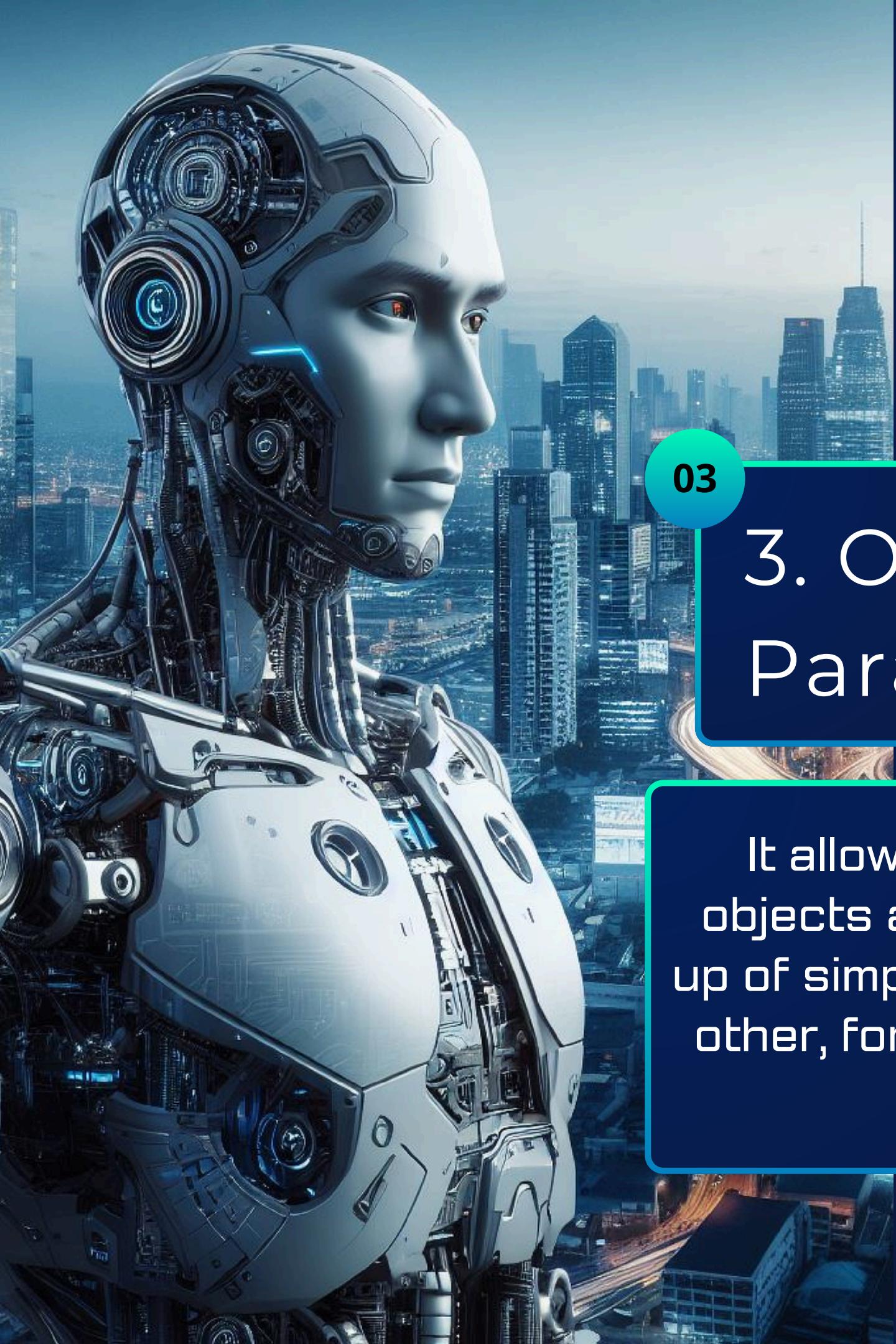
It is concerned with the final result
from the beginning.
The thing that you need / Final result.



Example:

C++ Code:

```
1  /*Implement an algorithm to display the  
2   odd numbers from 1 to 100.*/  
3  
4  #include <iostream>  
5  using namespace std;  
6  
7  int main() {  
8      int count=0;  
9      for (int i=1; i<=100; i+=2) {  
10          count += 1;  
11          cout << "Odd # " << count << " |is: " << i << endl;  
12      }  
13      return 0;  
14  }  
15
```

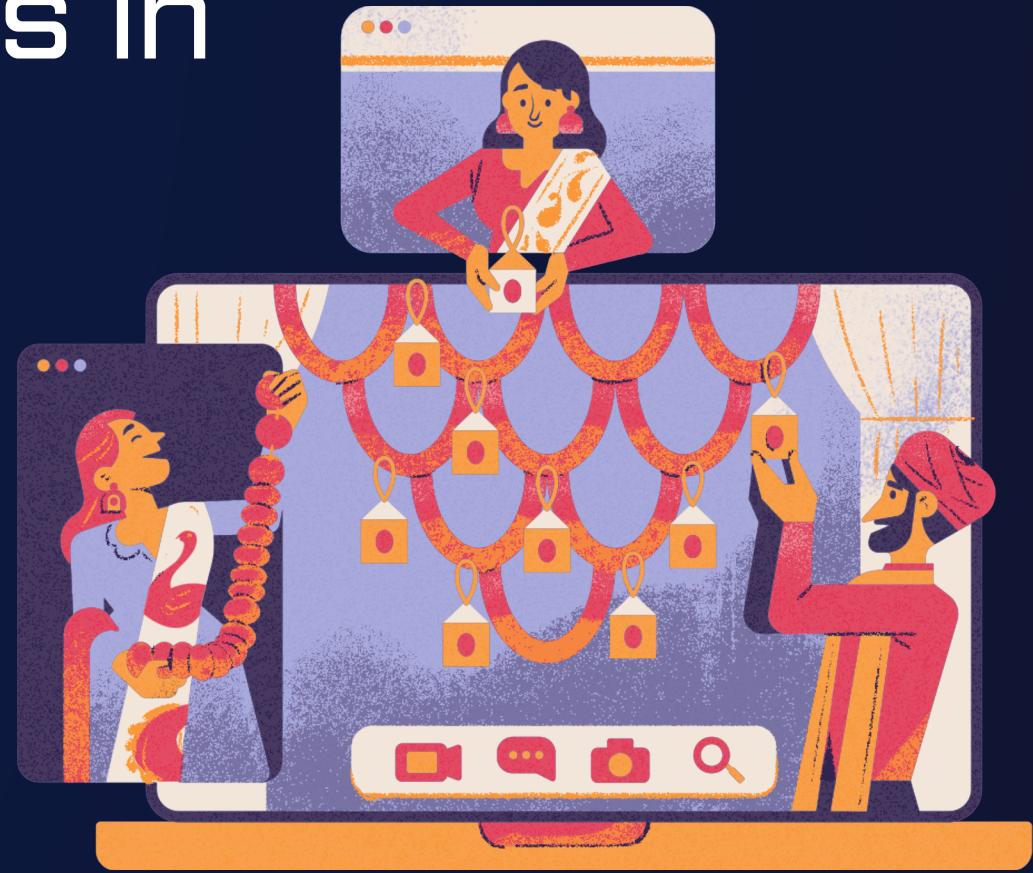


The main paradigms in programming

03

3. Object Oriented Paradigm (OOP)

It allows you to identify how to work with it through objects and code plans. This type of paradigm is made up of simple pieces or objects that, when related to each other, form different components of the system we are working on.

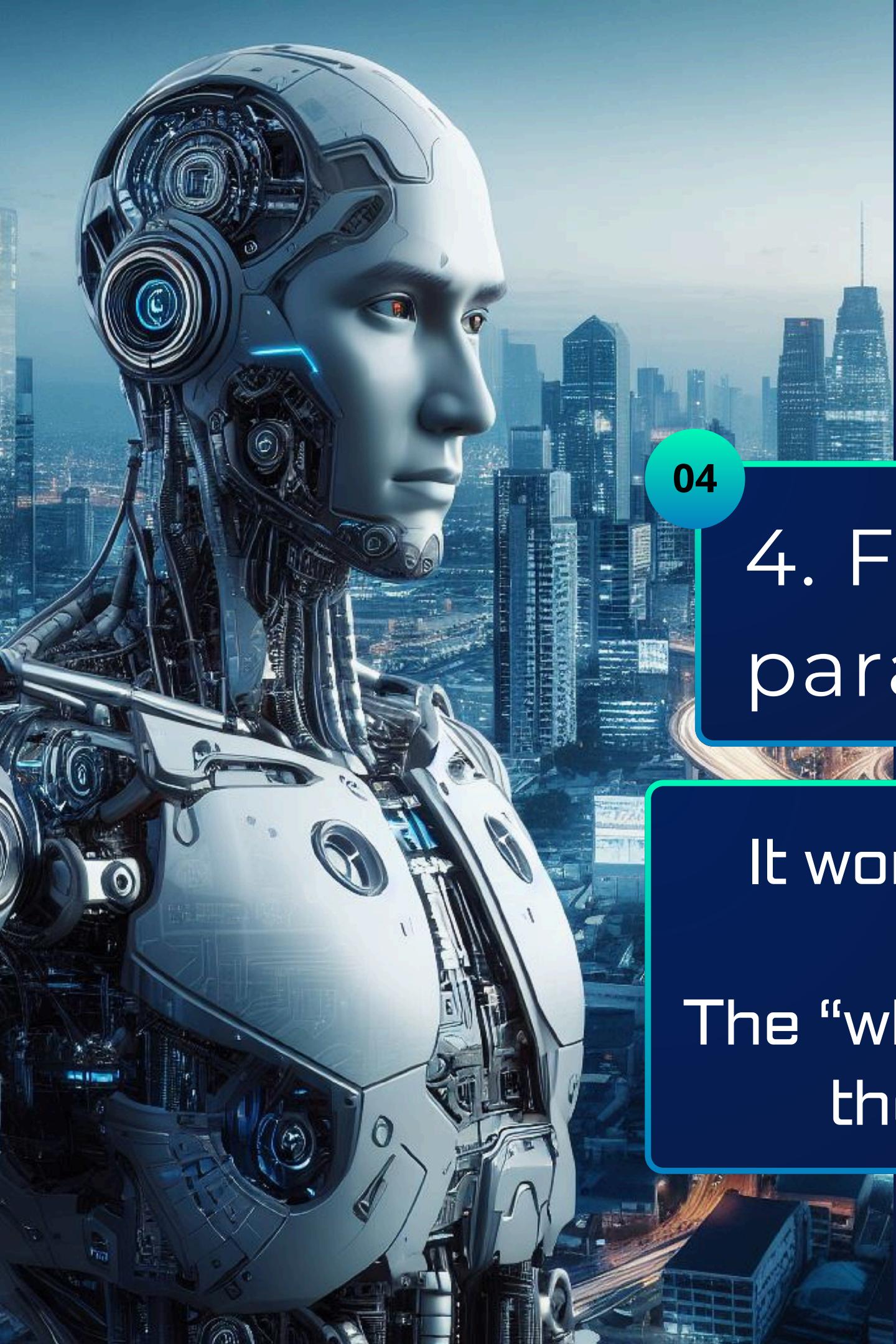


Example: Java

```
public class GFG {  
  
    static String Employee_name;  
    static float Employee_salary;  
  
    static void set(String n, float p) {  
        Employee_name = n;  
        Employee_salary = p;  
    }  
  
    static void get() {  
        System.out.println("Employee name is: " +Employee_name );  
        System.out.println("Employee CTC is: " + Employee_salary);  
    }  
  
    public static void main(String args[]) {  
        GFG.set("Rathod Avinash", 10000.0f);  
        GFG.get();  
    }  
}
```

Output

```
Employee name is: Rathod Avinash  
Employee CTC is: 10000.0
```



The main paradigms in programming

04

4. Functional paradigm

It works through certain mathematical functions.

The “what” matters more and not so much the “how” a project is developed.



Example: Python

```
1 # Implement an algorithm to display the odd numbers from 1 to 100.  
2 numbers = [1, 2, 3, 4, 5]  
3 squares = list(map(lambda x: x ** 2, numbers))  
4 print(squares)
```

```
[1, 4, 9, 16, 25]  
  
...Program finished with exit code 0  
Press ENTER to exit console.█
```

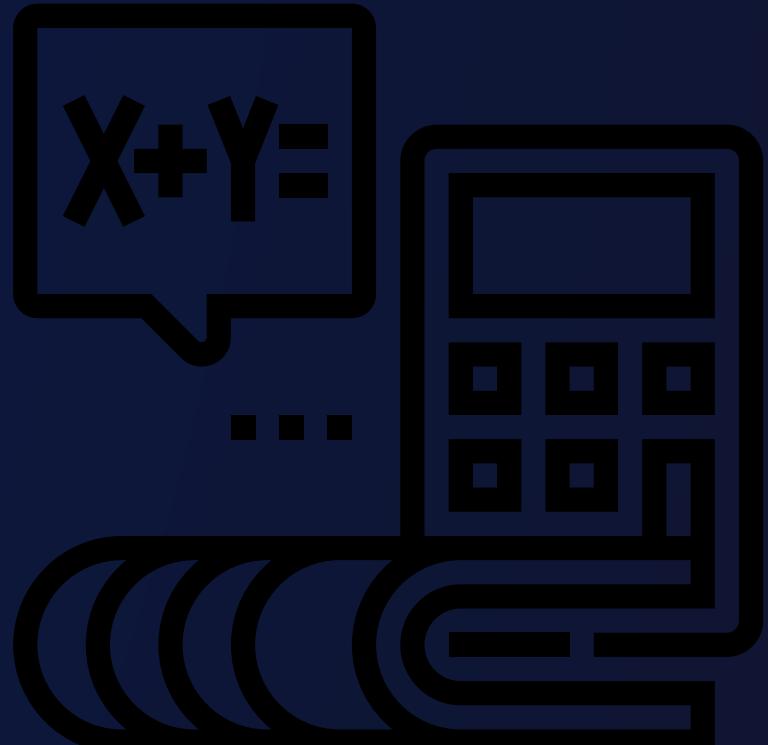


The main paradigms in programming

05

5. Reactive paradigm

The reactive paradigm is focused on analyzing the flow of data, whether finite or infinite, in order to respond to the needs that arise during the development of projects in order to the change of values, that are produced by data flows.



Example:

Java

```
1 /* Create an Observable that emits workdays from Monday to Friday. */
2
3 Observable<String> workdays = Observable.fromArray( "Monday", "Tuesday", "Wednesday", "Thursday", "Friday")
4 workdays.subscribe(
5     day -> System.out.println(day),
6     error -> System.out.println("Error: " + error),
7     () -> System.out.println("Stream completed.")
8 );
```

Paradigm transition

In the world of programming there are different techniques and approaches with which efficient software can be developed in solving problems. These approaches are known as programming paradigms and each of these paradigms has its own style and restrictions, which is why it is They have created different types of paradigms. Once the usefulness of each paradigm is known, a transition can be made, taking into account the following aspects.

- 1) Have a solid understanding of the paradigm in which you are going to work
- 2) Become familiar with the tools used in each paradigm
- 3) Make the transition gradually



JAVA



PYTHON



C++



SQL



SCALA



C

THANK YOU <3

NO QUESTIONS
pls



Página de Recursos

