MEPL

The Most Elegant Programming Language

Motivations

The creation of MEPL was a personal challenge and an enriching learning experience, undertaken as part of the Computer Logic course at Insper. It involved a deep dive into the complexities of programming language design, encompassing everything from the intricacies of lexer and parser development to the nuanced realms of semantic analysis and the final stages of code generation.

The conception of MEPL is rooted in the desire to create an educational tool that enhances the teaching of fundamental programming concepts such as variables, loops, and conditionals, as well as programming logic. The aim is to make the learning process more intuitive and refined, thereby facilitating a deeper understanding of these essential elements in programming.

Features

MEPL is distinguished by its unique blend of features, designed to provide a seamless and intuitive coding experience. Key characteristics include:

- **Highly Descriptive:** MEPL is crafted with clarity in mind. Each element of the language is designed to be self-explanatory, ensuring that even complex concepts are easily understandable. This descriptiveness aids in making the language accessible to beginners and seasoned programmers alike.
- **Fluid Writing Style:** The syntax of MEPL flows like natural language, making the process of writing code more like narrating a story. This fluidity not only makes coding more enjoyable but also facilitates a smoother transition for those new to programming.
- **Intuitive Syntax:** MEPL's syntax is intuitively designed, emphasizing logic and instinctive understanding. From defining variables and establishing loops to crafting conditional statements, the syntax aligns closely with natural thought processes.

Curiosity

The stereotype of programmers as sloppy or careless is a narrative that has persisted in popular culture, often portraying them as individuals more focused on functionality than form, and less concerned with the aesthetics or elegance of their craft. This image, while perhaps rooted in the pragmatic and problem-solving nature of programming, overlooks the artistry and creativity inherent in coding.

MEPL stands as a philosophical statement to the idea that programming is not merely a technical endeavor but a form of expression that marries logic with elegance. By emphasizing a syntax that is both intuitive and fluid, yet sophisticated and refined, MEPL breaks away from the informal universe that programming is typically associated with. It invites programmers to take pride in their craft, to see themselves not just as technicians, but as artisans and creators, whose work is not only functional but also beautiful.

Example

Always write a salutation before starting to code.

```
Greetings.
```

Variable declaration:

```
I introduce x as an integer.
I introduce y as an integer carrying the value of 7.
```

Example

Information display on the terminal.

```
Kindly display the following expression: "Welcome!".
```

User input solicitation.

```
May I solicit a brand new value for x.
```

Example

Variable assignment.

```
Please, let x be given the value of 0.
```

Creation of loops.

```
Whilst x differs from y holds true:
Firstly,
    [Operations to be executed inside loop]
... as a conclusion to the matter.
```

Example

Construction of conditionals.

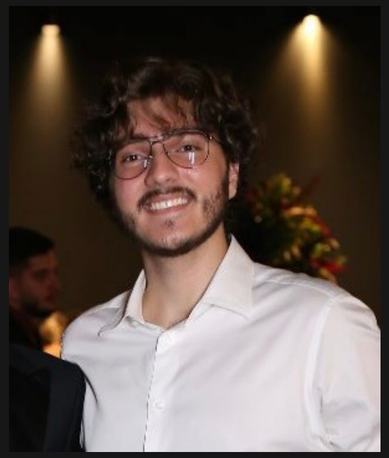
```
In the event that x exceeds y holds true:
Firstly,
    [Operations to be executed if condition is met]
... as a conclusion to the matter.
Otherwise:
Firstly,
    [Operations to be executed otherwise]
... as a conclusion to the matter.
```

Thank you

Thank you sincerely for your time and attention.

Your contributions and suggestions are cheerfully embraced.

Warm regards.



I am Renato Laffranchi Falcão, the creator of MEPL and a Student of Computer Engineering at Insper.

Contact me via email at:renatolf1@al.insper.edu.br