

# MEPEC

Mostra de Ensino, Pesquisa, Extensão e Cidadania



## Uso de reflexão para verificação de tipos dinâmicos em objetos Python

Gabriel Eduardo Lima, Hylson Vescovi Netto

[limaedugabriel@gmail.com](mailto:limaedugabriel@gmail.com)

Atualmente, cada vez mais os sistemas computacionais demonstram-se importantes para a sociedade, observada a quantidade de atividades realizadas com o auxílio dessas ferramentas. Um motivo para esse sucesso refere-se a possibilidade de implementação de diferentes algoritmos usando linguagens de programação, abstraindo a complexidade da máquina que é programada. Essas linguagens podem ser analisadas em diferentes aspectos, como por exemplo, o método de implementação, os paradigmas de programação, e seu sistema de tipos. Esse último define o comportamento dos dados e seus tipos em uma linguagem. Os sistemas de tipos são classificados, tipicamente, sob dois aspectos: tipagem estática/dinâmica e tipagem forte/fraca. Considerados os aspectos anteriores, outro detalhe interessante presente em algumas linguagens, refere-se à habilidade de realizar reflexão, ou metaprogramação. No contexto da computação, reflexão refere-se à capacidade de um sistema em conhecer a si próprio e o ambiente no qual está inserido, sendo capaz de manipulá-lo. É uma espécie de “programação sobre a programação”, onde utiliza-se informações referente ao próprio código e ambiente da linguagem para programar. Usualmente, o recurso é implantado junto ao modelo de orientação a objetos da linguagem. Sendo assim, o presente trabalho tem como objetivo discutir a possibilidade de usar recursos reflexivos na linguagem Python, visando realizar a checagem de tipos em objetos em uma linguagem de tipagem dinâmica. Como justificativa, cita-se que devido ao fato da tipagem dinâmica associar os tipos de dados no instante da execução do programa, erros como atribuição do tipo incorreto podem ocorrer e comprometer o sistema em tempo de execução. Além disso, Python foi escolhido por atender aos critérios de tipagem dinâmica, reflexão e orientação a objetos, bem como por ser uma linguagem de fácil aprendizado. A metodologia aplicada consiste na realização de uma pesquisa bibliográfica e uma pesquisa exploratória, objetivando desenvolver um mecanismo reflexivo para checagem de tipos em objetos. Dessa forma, como resultado foi possível desenvolver um modelo reflexivo simples utilizando o recurso de anotações do Python, onde um método genérico é capaz de verificar a coerência de tipos simples entre as definições e associações de dados. Com isso, por exemplo, pode-se validar os tipos dos dados associados aos objetos na hora de sua instanciação, retornando um aviso de erro caso exista inconsistência. Dessa forma, conclui-se ser possível utilizar recursos de metaprogramação para verificação de tipos em uma linguagem de tipagem dinâmica como Python. Além disso, destaca-se a possibilidade de criação de um modelo genérico de abstração para realizar essa funcionalidade. Por fim, como trabalhos futuros cita-se: Aprimoramento do modelo para tipos complexos – como listas; aplicação do mesmo

conceito em outras linguagens similares; implantação de um sistema real com o recurso em questão, analisando os impactos positivos e negativos do modelo.

ANDERSEN A. A note on reflection in Python 1.5. Lancaster: Distributed Multimedia Group, Lancaster University, 1998 (tech. report MPG-98- 05).

BAILEY C. Annotations. Real Python. Disponível em: <<https://realpython.com/lessons/annotations/#description>>. Acesso em: 11 set. 2022.

IBRAHIM M. H. Reflection in object-oriented programming. International Journal on Artificial Intelligence Tools, v. 1, n. 1, p. 117-136, 1992.

MAES P. Concepts and experiments in computational reflection. ACM SIGPLAN Notices, v.22, n.12, p. 147–155, 1987.

MALIK F. Advanced Python: Metaprogramming. Medium, 30/6/2020. Disponível em: <<https://medium.com/fintechexplained/advanced-python-metaprogramming-980da1be0c7d>>. 11/9/22

MENEZES, Nilo Ney Coutinho. Introdução à programação com Python: algoritmos e lógica de programação para iniciantes. 2. ed. São Paulo: Novatec, 2014. 328 p.

PYTHON SOFTWARE FOUNDATION. Python 3.8.13 documentation, 2022. Página de documentação. Disponível em: <<https://docs.python.org/3.8/>>. Acesso em: 11 set. 2022.

SEBESTA, Robert W. Conceitos de linguagens de programação. 9. ed. Porto Alegre: Bookman, 2011. 792 p. ISBN 9788577807918.