

Relatório

A transição de Java para Python é marcada, em primeiro lugar, por uma sintaxe visivelmente mais simples. O código Python dispensa a declaração obrigatória de tipos de variáveis, o uso de ponto e vírgula no final das linhas e as chaves ({}) para definir blocos, resultando em uma alternativa mais limpa em comparação à sintaxe do Java. O principal desafio de adaptação reside na diferença de "tipagem": o Java usa tipagem estática, verificando os tipos antes do programa rodar, enquanto o Python usa tipagem dinâmica, fazendo essa verificação apenas quando o programa já está em execução.

Ao migrar de C para Python, a mudança mais profunda é a abstração do gerenciamento de memória: o controle manual de alocação e liberação, essencial em C, é totalmente substituído pelo coletor de lixo automático do Python. Isso, por si só, elimina classes inteiras de erros complexos, como vazamentos de memória e falhas de segmentação. Na prática, as estruturas de C foram trocadas pelos dicionários do Python, que são mais flexíveis, e a complexa e arriscada manipulação de textos (strings) de C deu lugar aos métodos nativos e seguros do Python. Essas mudanças representam uma melhoria clara na robustez do código e na produtividade do desenvolvimento.

Por fim, a migração de COBOL para Python provou ser o desafio mais complexo, exigindo uma mudança total na forma de pensar a programação. A arquitetura do COBOL, que é rigidamente dividida em seções, teve que ser completamente reestruturada para se adaptar a um script Python moderno. A maneira como o COBOL declara dados, é o oposto direto das variáveis dinâmicas do Python. Além disso, os comandos de controle de fluxo, foram traduzidos para os loops for que são padrão hoje. A maior oportunidade de modernização foi converter os "arrays" de tamanho fixo do COBOL em listas dinâmicas do Python. Isso permitiu que o novo sistema, pela primeira vez, ultrapassasse as limitações de design do programa original.