

Python MRO

تحقیق پژوهشی

برنامه‌نویسی وب

عبارت python MRO کوتاه شده‌ی عبارت python method resolution order است که به کمک آن مسیر جستجوی کلاس‌ها مشخص می‌شود. این روش برای یافتن method درست در کلاس‌هایی که از multiple-inheritance استفاده می‌کنند به کار می‌آید. در پایتون هنگامی که ارث‌بری چندگانه بوجود می‌آید، برای جستجو در کلاس‌ها یک لیست ارث‌بری درست می‌شود که مسیر جستجو را تعیین می‌کند. برای این منظور یک روش قدیمی و یک روش جدید وجود دارد که بعد از پایتون 2.2 روش جدید ایجاد شد. در ادامه روش قدیمی و روش جدید توضیح داده خواهد شد.

روش قدیمی:

روش قدیمی در پایتون 2.2 به بعد تا قبل از پایتون 3 کار می‌کند به شرطی که هیچ کدام از پدران کلاسی که از آن نمونه گرفته شده از کلاس object پایتون ارث‌بری نکرده باشند. حال این روش را به کمک یک مثال نمایش می‌دهیم.

```
Class X():
    #def printer()
Class Y():
    def printer()
Class A(X, Y):
    #def printer()
Class B(Y, X):
    #def printer()
Class F(A, B):
    #def printer()

inst = F()
inst.printer()
```

در این حالت چون پدر کلاس A و B کلاس object نیست، از همان روش قدیمی استفاده می‌شود. این روش به این شکل است که عمقی طی می‌کند (الگوریتمی شبیه dfs):

۱. در خود کلاسی که نمونه از آن گرفته شده به دنبال متد printer می‌گردد.
 ۲. اگر متد printer را نیافت، به اولین پدرش می‌رود. حال اگر در پدرش نیز متد printer را نیابد، به پدر پدرش می‌رود و به همین صورت ادامه می‌یابد تا پدری وجود نداشته باشد.
 ۳. اما اگر پدر کلاسی در روند بالا وجود نداشت، می‌رود خود کلاس را می‌بیند و اگر از کلاس دیگری ارث‌بری کرده بود، آن کلاس را انتخاب کرده و در آن کلاس به دنبال متد printer می‌گردد و روند ۲ را دوباره ادامه می‌دهد.
- در مثال فوق این مسیر برابر است با:

F, A, X

حال چون متد print در X موجود نبود، می‌رود و کلاس دیگر یعنی Y را می‌گردد.

F, A, X, Y

حال دوباره چون متد print در Y موجود نبود، یک کلاس باز می‌گردد و به دنبال کلاس دیگری می‌گردد که دوباره پیدا نمی‌کند و دوباره یک مرحله باز می‌گردد که به خود کلاسی که از آن نمونه گرفته شده بود می‌رسیم، حال اگر کلاس دیگری موجود بود، مراحل را برای آن طی می‌کنیم، وگرنه به جستجو پایان می‌دهیم. حال بعد از انجام مراحل برای B نتیجه به شکل زیر می‌شود:

F, A, X, Y, B, X, Y

سپس، چون عنصر تکراری مجاز نیست، عناصر تکراری را حذف می‌کنیم و نتیجه نهایی برابر است با:

F, A, X, Y, B

روش جدید:

روش جدید در حالتی که پدران کلاسی که از آن نمونه گرفته شده، از کلاس python object ارث‌بری کرده باشند، به کار می‌رود و چون از پایتون 3 به بعد پدر تمامی کلاس‌ها به صورت پیش‌فرض object است، همواره از روش جدید استفاده می‌شود. حال این روش را بر اساس مثالی که در بالا گفته شد توضیح می‌دهیم. در این روش مسیر به کمک فرمول زیر تعیین می‌شود.

$$L[C(B1 \dots BN)] = C + \text{merge}(L[B1] \dots L[BN], B1 \dots BN)$$

هم چنین الگوریتم merge نیز به صورت زیر انجام می‌شود:

۱. اولین عنصر لیست L را بر می‌دارد.
۲. حال اگر این عنصر در دنباله‌ی (tail) لیست‌های دیگر نبود، این عنصر را به لیست خطی C اضافه می‌کنیم و آن را از لیست در merge حذف می‌کنیم.
۳. اگر شرط بالا برقرار نبود، به لیست بعدی می‌رویم و دوباره مراحل ۱، ۲ را انجام می‌دهیم.
۴. این الگوریتم را تا جایی انجام می‌دهیم که همه کلاس‌ها از لیست حذف شده باشند یا آنکه نتوانیم هیچ اولین عنصری (head) با شرایط ۲ بیابیم.
۵. اگر هیچ عنصر قابل قبولی نیافتیم، اکسپشن رخ می‌دهد و در پایتون 2.3 از ایجاد کلاس جلوگیری می‌شود.

حال این الگوریتم را برای مثالی که در بالا ذکر شد انجام می‌دهیم:

$$L[A(X, Y)] = A + \text{merge}(L[X], L[Y], X, Y) = A, X, Y$$

$$L[B(Y, X)] = B + \text{merge}(L[Y], L[X], Y, X) = B, Y, X$$

$$\begin{aligned} L[F(A, B)] &= F + \text{merge}(L[A], L[B], A, B) \\ &= F + \text{merge}((A, X, Y) + (B, Y, X), A, B) \end{aligned}$$

طبق روشی که در بالا گفته شد، اولین عنصر لیست اول A است، و چون دنباله‌ی هیچ لیست دیگری نیست، A را از لیست اولیه حذف می‌کنیم و به لیست خطی F اضافه می‌کنیم.

$$L[F(A, B)] = F, A + \text{merge}((X, Y) + (B, Y, X), B)$$

اکنون اولین عنصر لیست اول X است که دنباله‌ی لیست دیگری است، پس به سراغ لیست بعدی می‌رویم. اولین عنصر لیست بعدی B است که دنباله‌ی لیست دیگری نیست، پس B را از لیست اولیه حذف می‌کنیم و به لیست خطی F اضافه می‌کنیم.

$$L[F(A, B)] = F, A, B + \text{merge}((X, Y) + (Y, X))$$

در نهایت دوباره اولین عنصر لیست اول X است که دنباله‌ی لیست دیگری است. پس سراغ لیست دوم می‌رویم که اولین عنصر آن Y است. Y نیز دنباله‌ی لیست دیگری است، پس در این حالت هیچ گزینه‌ی دیگری وجود ندارد. هم‌چنین چون امکان خطی‌سازی وجود ندارد، اکسپشن رخ می‌دهد.

دو روشی که در بالا ذکر شد، برای یافتن مسیر جستجوی کلاس‌ها در ارث‌بری چندگانه استفاده می‌شوند.

این محتوا از این منبع گرفته شده و از این سایت قابل دسترسی است.