README.md

Ringkasan :

- Bab 8 Kesalahan dan Pengecualian

Hingga saat ini pesan kesalahan belum banyak disebutkan, tetapi jika Anda telah mencoba contoh-contohnya, Anda mungkin telah melihat beberapa.

Syntax Errors

Parser mengulangi baris yang bermasalah dan menampilkan 'panah' kecil yang menunjuk ke titik paling awal di baris tempat kesalahan terdeteksi.

Pengecualian

Bahkan jika sebuah pernyataan atau ekspresi benar secara sintaksis, hal itu dapat menyebabkan kesalahan saat dilakukan upaya untuk mengeksekusinya. Baris terakhir dari pesan kesalahan menunjukkan apa yang terjadi. Bagian sebelumnya dari pesan kesalahan menunjukkan konteks di mana pengecualian terjadi, dalam bentuk stack traceback. Pengecualian Bawaan mencantumkan pengecualian bawaan dan artinya.

Menangani Pengecualian

- Pertama, klausa try pernyataan dieksekusi. - Jika tidak ada pengecualian yang terjadi, klausa kecuali akan dilewati dan eksekusi pernyataan try selesai. - Jika pengecualian terjadi selama eksekusi klausa try, klausa lainnya akan dilewati. Kemudian, jika tipenya cocok dengan pengecualian yang dinamai setelah kata kunci kecuali, klausa kecuali dieksekusi, dan kemudian eksekusi dilanjutkan setelah blok coba/kecuali.

Pernyataan try mungkin memiliki lebih dari satu kecuali klausa, untuk menentukan penangan untuk pengecualian yang berbeda. Kelas dalam klausa yang kecuali kompatibel dengan pengecualian jika itu adalah kelas yang sama atau kelas dasar daripadanya . Ketika pengecualian terjadi, itu mungkin memiliki nilai terkait, juga dikenal sebagai argumen pengecualian. Kehadiran dan jenis argumen bergantung pada jenis pengecualian.

Klausa kecuali dapat menentukan variabel setelah nama pengecualian. Variabel terikat pada instance pengecualian yang biasanya memiliki atribut args yang menyimpan argumen. Untuk kenyamanan, tipe pengecualian bawaan mendefinisikan **str** untuk mencetak semua argumen tanpa mengakses secara eksplisit . Output **str** pengecualian dicetak sebagai bagian terakhir dari pesan untuk pengecualian yang tidak tertangani.

BaseException adalah kelas dasar umum dari semua pengecualian. Salah satu subkelasnya, Pengecualian, adalah kelas dasar dari semua pengecualian non-fatal. Pengecualian yang bukan merupakan subkelas Pengecualian biasanya tidak ditangani, karena digunakan untuk menunjukkan bahwa program harus dihentikan.

Meningkatkan Pengecualian

Pernyataan kenaikan memungkinkan programmer untuk memaksa pengecualian tertentu terjadi. Satu-satunya argumen yang akan dinaikkan menunjukkan pengecualian yang akan dimunculkan.

Rangkaian Pengecualian

Untuk informasi selengkapnya tentang mekanisme rangkaian, lihat Pengecualian Bawaan.

Pengecualian yang Ditetapkan Pengguna

Program dapat menamai pengecualiannya sendiri dengan membuat kelas pengecualian baru. Kelas pengecualian dapat didefinisikan yang melakukan apa pun yang dapat dilakukan kelas lain, tetapi biasanya tetap sederhana, seringkali hanya menawarkan sejumlah atribut yang memungkinkan informasi tentang kesalahan diekstraksi oleh penangan untuk pengecualian. Banyak modul standar menentukan pengecualian mereka sendiri untuk melaporkan kesalahan yang mungkin terjadi pada fungsi yang mereka tentukan.

Mendefinisikan Tindakan Pembersihan

Pernyataan try memiliki klausa opsional lain yang dimaksudkan untuk menentukan tindakan pembersihan yang harus dijalankan dalam semua keadaan. Jika klausa akhirnya ada, klausa akhirnya akan dieksekusi sebagai tugas terakhir sebelum pernyataan try selesai. Jika pengecualian tidak ditangani oleh pengecualian klausa, pengecualian akan dimunculkan kembali setelah klausa akhirnya dieksekusi.

- Pengecualian dapat terjadi selama pelaksanaan klausa kecuali atau yang lain. Sekali lagi, pengecualian dimunculkan kembali setelah klausa akhirnya dieksekusi.
- Jika klausa akhirnya menyertakan pernyataan pengembalian, nilai yang dikembalikan akan menjadi salah satu dari pernyataan pengembalian klausa akhirnya, bukan nilai dari pernyataan pengembalian klausa try. Seperti yang Anda lihat, klausa akhirnya dijalankan dalam acara apa pun.

Tindakan Pembersihan yang Telah Ditentukan

Masalah dengan kode ini adalah bahwa kode ini membiarkan file terbuka untuk waktu yang tidak ditentukan setelah bagian kode ini selesai dieksekusi. Ini bukan masalah dalam skrip sederhana, tetapi bisa menjadi masalah untuk aplikasi yang lebih besar. Setelah pernyataan dieksekusi, file f selalu ditutup, bahkan jika terjadi masalah saat memproses baris. Objek yang, seperti file, memberikan tindakan pembersihan yang telah ditentukan sebelumnya akan menunjukkan hal ini dalam dokumentasinya.

Menaikkan dan Menangani Banyak Pengecualian yang Tidak Terkait

Ada situasi di mana perlu melaporkan beberapa pengecualian yang telah terjadi. Dengan menggunakan exception* sebagai ganti dari exception, kita hanya dapat menangani pengecualian dalam grup yang cocok dengan tipe tertentu secara selektif. Dalam contoh berikut, yang memperlihatkan grup pengecualian bersarang, setiap klausa exception* mengekstrak dari pengecualian grup dari jenis tertentu sambil membiarkan semua pengecualian lainnya menyebar ke klausa lain dan akhirnya dinaikkan kembali. Perhatikan bahwa pengecualian yang bersarang di grup pengecualian harus berupa instance, bukan tipe.

Memperkaya Pengecualian dengan Catatan

Saat pengecualian dibuat untuk dimunculkan, biasanya diinisialisasi dengan informasi yang menjelaskan kesalahan yang terjadi. Ada kasus di mana berguna untuk menambahkan informasi setelah pengecualian tertangkap. Untuk tujuan ini, pengecualian memiliki metode add_note yang menerima string dan menambahkannya ke daftar catatan pengecualian.