

Klaidų vengimas, valdymas

Esant tikėtina klaidos situacijai, naudojama `try` sakiny.

Toks sakiny veikia panašiai kaip `if` sakiny, tačiau čia nėra sąlygų tikrinimo.

Pilno `try` šablonas

```
try:
    pass #čia bus vykdomas pagrindinis kodas
except Exception as ex: #except šakų gali būti daugiau nei 1-na
    pass #čia bus vykdomas kodas, jei try šakoje KILS klaida
else: #nebūtinai
    pass #čia bus vykdomas kodas, jei NEKILS problemų
finally: #nebūtinai
    pass #kodas bus vykdomas nepaisant ar kilo problema, ar ne.
```

Galimi standartiniai klaidų tipai

Rekomendacijos:

- Nenaudoti bazinio `Exception` tipo - specifikuoti veiksmus pagal galimas problemas
- Jei galima užtikrinti su `if` konstrukcija, kad nesusidarytų sąlygos klaidai - naudokite `if`

Jei neturėtumėme klaidų valdymo

```
numbers = [-2,-1,0,1,2]
for n in numbers:
    print(5/n) # čia trečio žingsnio metu bus 5/0 - dalyba iš 0
```

Su try konstrukcija

```
numbers = [-2,-1,0,1,2]
for n in numbers:
    try:
        print(5/n) # čia trečio žingsnio metu bus 5/0 - dalyba iš
        ↪ 0
    except:
        print('dalyba iš 0! Bet programa veiks toliau')
```

Parašykite kodą, kuris paklaustų vartotojo skaičių A ir B , matematinio veiksmo (+, −, /, *), atliktų veiksmą, rezultatą išspausdintų. Pritaikykite `try` bloką, kad išvengtumėte tokių situacijų kaip:

- dalyba iš 0;
- vietoj skaičiaus tekstas;

Pritaikykite visas `try` bloko šakas.