CYFROWA SZKOŁA
Wielkopolsk@ 2020

Obsługa wyjątków i liczby zmiennoprzecinkowe o większej precyzji

Wprowadzenie do języka Python (VII)

Nikt nie chce, aby ich programy się zawieszały. Powinniśmy napisać kod, który przewiduje złe rzeczy, a użytkownik może spowodować awarię programu poprzez dostarczenie złych danych i tą sytuację naprawić.

Jednak czasami użytkownik może próbować uzyskują dostęp do bazy danych, która nie istnieje, lub po prostu baza danych odmawia wykonywania instrukcji. To są niektóre przypadku gdzie pojawia się obsługa wyjątków.

Dzięki obsłudze wyjątków możemy złapać błąd, który normalnie spowodowałby awarię naszego programu i dać użytkownikowi możliwość właściwego postępowania.

W tym pierwszym przykładzie poprosimy użytkownika o wprowadzenie numeru, ale jeśli odmówi, poprosimy ponownie o

wyłapywanie i rozwiązywanie błędu.

Nasz problematyczny kod otoczymy blokiem try. Postaramy się wtedy złapać oczekiwany błąd w bloku z wyjątkiem. Jeśli użytkownik zrobi coś zupełnie nieoczekiwanego, użyjemy bloku bez definiowania konkretnego wyjątku, aby przechwycić wszystkie inne wyjątki.

CYFROWA SZKOŁA
Wielkopolsk@ 2020

KOD

```
# Nadając pętli while wartość True, będzie się powtarzać aż
do osiągnięcia insrukcji break
while True:


    # Jeśli spodziewamy się wystąpienia błędu, otocz
potencjalny błąd poleceniem try
    try:
        number = int(input("Proszę wprowadzić liczbę: "))
        break
```

CYFROWA SZKOŁA
Wielkopolsk@ 2020

```
 # Kod w bloku z wyjątkiem zawiera komunikat o błędzie, aby
naprawić wszystko
    # Możemy celować w konkretny błąd, taki jak ValueError
    except ValueError:
        print("Nie wprowadziłeś numeru")

    # Możemy kierować wszystkie inne błędy z wartością domyślną
    except:
        print("Wystąpił nieznany błąd")
        print("Dziękujemy za wprowadzenie numeru")
```

CYFROWA SZKOŁA
Wielkopolsk@ 2020

Problem programistyczny w Pythonie

Zasady pętli Do While są takie, że zawsze wykonują one cały kod przynajmniej raz. Po pierwszym obiegu pętli, jeśli warunek jest spełniony, pętla uruchamia kod ponownie.

W innych językach pętla Do While ma format

do {

... instrukcje do wykonania wewnątrz pętli ...

} while (warunek)

Należy stworzyć grę w zgadywanie, w której użytkownik musi wybrać liczbę od 1 do 10 w następującym formacie.

Odgadnij liczbę od 1 do 10 : 1

Odgadnij liczbę od 1 do 10 : 3

Odgadnij liczbę od 1 do 10 : 5

Odgadnij liczbę od 1 do 10 : 7

Zgadłeś

Podpowiedź: Potrzebujesz instrukcji while i break;

CYFROWA SZKOŁA
Wielkopolsk@ 2020

KOD

```
secret_number = 7
while True:
    zgadnij = int(input("Odgadnij liczbę od 1 do 10 : "))
    if zgadnij == secret_number:
        print("Zgadłeś")
        break
```

CYFROWA SZKOŁA
Wielkopolsk@ 2020

Liczby zmiennoprzecinkowe o większe precyzji

Jeśli nie jesteś zadowolony z precyzji 15 miejsc po przecinku w obliczeniach związanych z liczbami zmiennoprzecinkowymi to można zakres dokładności zwiększyć.

Moduł dziesiętny zapewnia dokładniejsze obliczenia zmiennoprzecinkowe. Domyślnie moduł posiada 28 pozycyjne rozwinięcie wartości po przecinku danej liczby.

Możemy również utworzyć tutaj alias oznaczony wartością Decimal, aby uniknąć konfliktów z metodami o tej samej nazwie z importu modułu dziesiętnego. Oto przykładowy kod, którego możesz użyć do pracy z dokładnymi liczbami zmiennoprzecinkowymi.

CYFROWA SZKOŁA
Wielkopolsk@ 2020

KOD

```
from decimal import *


suma = Decimal(0)
suma += Decimal("0.01")
suma += Decimal("0.01")
suma += Decimal("0.01")
suma -= Decimal("0.03")
print("Suma = ", suma)
```