

Výjimky

PVA2 Programování a vývoj aplikací

Obsah

1. Obsah
2. Výjimky
3. Jak vypadá výjimka?
4. Jaké chyby existují?
5. Jaké chyby existují?
6. Jak zachytit výjimku?
7. Jak zachytit všechny výjimky?
8. Zachytit více výjimek
9. Blok else a finally
10. Vlastní vyvolání výjimky
11. Využijte podmínky ve svůj prospěch

Výjimky

- Výjimkou `Exception` označujeme situaci, kdy program narazí na chybu.
- Mohou být vyvolány programem nebo interpretrem.
- Takovou chybu se ve vývoji snažíme ošetřit zachycením výjimky `catch`
- Pokud výjimku nezachytíme, program skončí s chybou.

Jak vypadá výjimka?

- Zkusíme spustit následující kód:

```
1 a = 10
2 b = 0
3 print( a/b ) # Dělení nulou
```

- Co se stane? Interpret Pythonu vyvolá výjimku `ZeroDivisionError`

```
1 Traceback (most recent call last):
2   File "test.py", line 3, in <module>
3     print(a/b)
4   ZeroDivisionError: division by zero
```

Jaké chyby existují?

<https://docs.python.org/3/library/exceptions.html#exception-hierarchy>

- `ArithmeticError` - Obecná aritmetická chyba
 - `ZeroDivisionError` - Dělení nulou
 - `OverflowError` - Přetečení
 - `FloatingPointError` - Chyba s pohyblivou řádovou čárkou
- `SyntaxError` - Chyba syntaxe
 - `IndentationError` - Chyba odsazení
 - `TabError` - Chyba tabulátoru (kombinování mezer a tabulátorů v odsazení)

Jaké chyby existují?

- `NameError` - Neexistující proměnná
- `TypeError` - Nesprávný typ proměnné
- `ValueError` - Nesprávná hodnota proměnné
- `IndexError` - Index mimo rozsah
- `KeyError` - Klíč neexistuje
- `FileNotFoundError` - Soubor nenalezen
- `ImportError` - Chyba při importu modulu

Jak zachytit výjimku?

- Výjimku zachytíme pomocí bloku `try` a `except`
- V bloku `try` se nachází kód, který může vyvolat výjimku
- V bloku `except` zachytíme výjimku a můžeme s ní pracovat
- Pokud výjimka nenastane, blok `except` se neprovede

```
1  try:
2      a = 10
3      b = 0
4      print( a/b ) # Dělení nulou
5  except ZeroDivisionError:
6      print("Nelze dělit nulou")
```

Jak zachytit všechny výjimky?

- Pokud chceme zachytit všechny výjimky, můžeme použít blok `except` bez specifikace typu výjimky
- Použitím `as e` uložíme výjimku do proměnné `e` a tu následně zobrazíme uživateli

```
1  try:
2      a = 10
3      b = 0
4      print( a/b ) # Dělení nulou
5
6  # Zachytíme všechny výjimky a uložíme je do proměnné e
7  except Exception as e:
8      print("Chyba:", e)
```


Zachytit více výjimek

- Můžeme zachytit více výjimek pomocí více bloků `except`
- Každý blok `except` zachytí jiný typ výjimky
- Výjimky se zpracovávají shora dolů, takže pokud chceme zachytit konkrétní výjimku, musíme ji mít výše

```
1  try:
2      a = 10
3      b = 0
4      print( a/b ) # Dělení nulou
5
6  # Zachycení konkrétní výjimky
7  except ZeroDivisionError:
8      print("Nelze dělit nulou")
9
10 # Zachycení nadřazené výjimky skupiny výpočetních chyb
11 except ArithmeticError:
12     print("Výpočetní chyba")
13
14 # Zachycení všech výjimek
15 except Exception as e:
16     print("Chyba:", e)
```

Blok `else` a `finally`

- Blok `else` se provede, pokud v bloku `try` nenastala výjimka
- Blok `finally` se provede vždy, bez ohledu na to, zda v bloku `try` nastala výjimka nebo ne

```
1  try:
2      a = 10
3      b = 2
4      print( a/b ) # Dělení nulou
5  except ZeroDivisionError:
6      print("Nelze dělit nulou")
7  except Exception as e:
8      print("Chyba:", e)
9  else:
10     print("Vše proběhlo v pořádku")
11 finally:
12     print("Konec programu")
```

Vlastní vyvolání výjimky

- Můžeme vyvolat vlastní výjimku pomocí klíčového slova `raise`
- Výjimku můžeme vyvolat s vlastním textem

```
1  try:
2      raise Exception("Vlastní chyba")
3  except Exception as e:
4      print("Chyba:", e)
```

```
1  def sumSquare(strana):
2      if strana > 0:
3          return strana ** 2
4      else:
5          # raise zpracování a vyvolá exception
6          # za raise se udává typ výjimky. nejčastěji ValueError
7          raise ValueError(f'Délka strany musí být kladná. Zadáno: {strana}')
```

```
1  # Zavoláme funkci `sumSquare` s parametrem -5 a očekáváme výjimku `ValueError`
2  # Vrátí chybu `ValueError: Délka strany musí být kladná. Zadáno: -5`
3  try:
4      print(sumSquare(-5))
5  except ValueError as e:
6      print("Chyba:", e)
```

Využijte podmínky ve svůj prospěch

- Cyklus se bude opakovat neustále dokola, dokud nebude zadáno, co jsme od uživatele požadovali

```
1  def numInput():
2      while True:
3          answer = input('Zadej celé číslo: ')
4          try:
5              # Zkusí se provést konverze na číslo
6              # Pokud není úspěšná, je vrácena chyba
7              return int(answer)
8          except ValueError:
9              print('Nebylo zadáno celé číslo!')
10
11  numInput()
```

Děkuji za pozornost

Otázky?

Repository / Prezentace