



Programování 1 (NPRG030)

Cvičení 9: Výjimky, logování a ladění kódu

Shrnutí přednášky: Výjimky

```
1 # Ošetření výjimek a logování běhu programu
2 def prevod_retezce_na_cislo(retezec: str) -> float:
3     try:
4         cislo = float(retezec)
5     except ValueError as ex:
6         logging.warning("Převod řetězce '{retezec}' na číslo se nepovedl")
7         return None
8     except Exception as ex:
9         logging.error("Při převodu řetězce '{retezec}' nastala neočekávaná chyba: {ex}")
10        raise ex
11    else:
12        return cislo
13    finally:
14        logging.debug("Funkce byla vykonána")
```

ošetření výjimky

lze zachytit více výjimek

propagace výjimky

kód ve finally bloku se vykoná vždy, i po zavolání return

```
15 # Použití assertu
16 def deleni(delenec: float, delitel: float) -> float:
17     assert delitel != 0, "Dělení nulou není povoleno"
18     vysledek = delenec / delitel
19     return vysledek
```

Shrnutí přednášky: Přehled výjimek

1	# Přehled vybraných výjimek	
2	<code>BaseException</code>	# Základní třída pro všechny výjimky
3	<code>Exception</code>	# Obecná třída pro výjimky
4	<code>StopIteration</code>	# Signál pro ukončení iterace
5	<code>OverflowError</code>	# Přetečení při operacích s čísly
6	<code>ZeroDivisionError</code>	# Dělení nulou
7	<code>AssertionError</code>	# Selhání výroku "assert"
8	<code>AttributeError</code>	# Pokus o přístup k neexistujícímu atributu
9	<code>ImportError</code>	# Chyba při importování modulu nebo balíčku
10	<code>ModuleNotFoundError</code>	# Importovaný modul nebyl nalezen
11	<code>IndexError</code>	# Pokus o přístup k neexistujícímu indexu v seznamu nebo řetězci
12	<code>KeyError</code>	# Pokus o přístup k neexistujícímu klíči v slovníku
13	<code>MemoryError</code>	# Nedostatek (systémové) paměti
14	<code>NameError</code>	# Nepředefinovaná proměnná nebo jméno
15	<code>FileExistsError</code>	# Pokus o vytvoření již existujícího souboru
16	<code>FileNotFoundError</code>	# Soubor nebyl nalezen
17	<code>IsADirectoryError</code>	# Očekávání souboru, ale byl to adresář
18	<code>NotADirectoryError</code>	# Očekávání adresáře, ale byl zadáný soubor
19	<code>PermissionError</code>	# Chybějící oprávnění k operaci (např. přístup k souboru)
20	<code>NotImplementedError</code>	# Výjimka pro neimplementovanou funkcionalitu
21	<code>TypeError</code>	# Nesprávný datový typ nebo operace
22	<code>ValueError</code>	# Neplatná hodnota pro daný datový typ nebo funkci

Příklad 9.4: Převod řetězce na číslo

- ❖ Napište *funkci* v jazyce Python, která *převeďte* zadaný *řetězec na číslo*
- ❖ Vstupem funkce je řetězec
- ❖ Výstupem funkce je převedené číslo (float)
- ❖ Pokud se převod řetězce nepodaří, funkce vrátí None
 - ❖ Tj. *funkce výjimku ošetří* a *nepropaguje* ji do vyšší úrovně kódu

Cvičení 9.5: Vyhledání prvku v seznamu

- ❖ Napište *funkci* v jazyku Python, která bude *vyhledávat prvek v* uživatelem zadaném *seznamu*
- ❖ Pokud je prvek nalezen, funkce vrátí index prvního výskytu hledané hodnoty
- ❖ Naopak, pokud prvek v listu neexistuje, *vyhodte výjimku* `ValueError`

Cvičení 9.6: Vytvoření vlastní výjimky

- ❖ Vytvořte kód v jazyce Python, který provádí výpočet plochy obdélníka a který ověřuje, že délka a šířka obdélníka jsou kladná čísla
- ❖ Pokud je *některá ze stran záporná*, vyhodte vlastní výjimku (a vhodně ji pojmenujte)
 - ❖ Výjimka by měla být odvozena od třídy `Exception`

Cvičení 9.7: Logování

- ❖ Upravte kód z vyřešeného příkladu 9.4 (převod řetězce na číslo) tak, abyste namísto volání funkce `print` *použili logování* z knihovny `logging`
- ❖ Vhodně použijte *různé úrovně logování*

Úroveň	Kdy použít
DEBUG	Podrobné informace, které mají význam při určování příčiny problému
INFO	Potvrzení, že kód funguje podle očekávání
WARNING	Nastala neočekávaná událost nebo upozornění na blížící se problém
ERROR	Program není schopen z důvodu závažnější chyby vykonat požadovanou funkci
CRITICAL	Závažná chyba, která může celý program uvést do nekonzistentního stavu

Cvičení 9.8: Debugging

- ❖ Úkolem programátora bylo napsání funkce v jazyku Python, která načte seznam studentů s jejich jmény a hodnocením a vypočítá průměrné hodnocení zaokrouhlené na dvě desetinná místa
- ❖ Nicméně programátor udělal v kódu několik chyb
 - ❖ Vaším úkolem je *chyby nalézt a opravit*
 - ❖ Pro nalezení chyb *použijte záznamy z logování*

Jak psát kód: Elegantní zpracování výjimek

- ❖ Vhodně zpracovávejte výjimky, abyste předešli (neočekávaným) pádům a zlepšili uživatelský komfort

```
1 try:  
2     result = perform_operation()  
3 except ValueError as e:  
4     print("Error:", e)  
5     result = None
```

Jak psát kód: Lokální zpracování výjimek

- ❖ Výjimky zachycujte a zpracovávejte jen tam, kde to má největší smysl, a nenechávejte je (zbytečně) šířit do vyšších úrovní kódu

```
1 # Propagace výjimky do vyšší úrovně
2 def divide(left_op, right_op):
3     if left_op == 0:
4         raise ZeroDivisionError("Cannot divide by zero")
5     return left_op / right_op
```

```
1 # Zachycení a zpracování výjimky
2 def divide(left_op, right_op):
3     try:
4         result = left_op / right_op
5     except ZeroDivisionError:
6         print("Cannot divide by zero")
7         result = None
8     return result
```


Reference

Výjimky

- ❖ https://www.w3schools.com/python/python_try_except.asp

Logování

- ❖ <https://docs.python.org/3/howto/logging.html>

Debugging