Programování v Pythonu

Algoritmizace a programování | IV. ročník

Cíle hodiny

- Žák vysvětlí principy práce se soubory v Pythonu a na příkladech demonstruje čtení a zápis dat do textových souborů.
- Žák samostatně pracuje s CSV soubory, načítá z nich data a ukládá je zpět, s využitím vhodných knihoven Pythonu.

1. Evokace (15 minut)

- **Proč potřebujeme soubory?**

  Motivační problém - Ztráta dat (2 minuty)

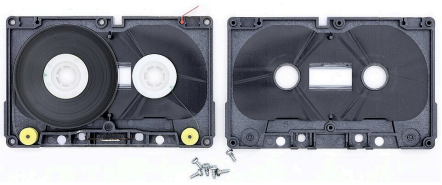
  Představte si situaci: Vytvoříte v Pythonu program pro evidenci známek ze všech předmětů za celý semestr. Program funguje perfektně, zadáte všechny známky, spočítáte průměry. Pak ale vypnete počítač a druhý den zjistíte, že všechna data jsou pryč. Proč se to stalo? Všechny proměnné a seznamy existují pouze v paměti počítače a po ukončení programu zmizí.

  Společné hledání řešení (3 minuty)

Problémová situace	Co potřebujeme vyřešit
Evidence známek z 15 předmětů	Uložit data tak, aby přežila vypnutí počítače
Seznam 200 kontaktů na spolužáky	Sdílet data mezi různými programy
Výsledky měření teploty za rok	Pracovat s velkým množstvím dat
Nastavení hry	Obnovit stav při příštím spuštění

  Diskuse: Kde všude ve vašich telefonech, počítačích a aplikacích se ukládají data? Studenti objeví, že fotky, kontakty, zprávy, nastavení - vše je uloženo v souborech. Bez souborů by každá aplikace začínala "od nuly".

  Praktický experiment: Spustíme jednoduchý Python program, který vytvoří seznam čísel, vypočítá jejich součet. Po ukončení programu a novém spuštění - data jsou pryč. Toto je důvod, proč potřebujeme soubory.



Datová kazeta Verbatim DC30
Zdroj: WikiMedia

```python
import os, time
x = "Welcome to The Mirror."
y = 0

while y <= len(x):
    os.system("clear")
    print(x[:y])
    time.sleep(0.2)
    y = y+1
time.sleep(2)
x = "You will stay here for as long as you can, you may forfeit when you would like to."
y = 0

while y <= len(x):
    os.system("clear")
    print(x[:y])
    time.sleep(0.2)
    y = y+1
time.sleep(2)
x = "Please stand here, and stare into the mirror."
y = 0

while y <= len(x):
    os.system("clear")
    print(x[:y])
    time.sleep(0.2)
    y = y+1
time.sleep(2)
x = "This is you in the mirror:"
y = 0

while y <= len(x):
    os.system("clear")
    print(x[:y])
```

Kód v Pythonu
Zdroj: WikiMedia

Výuka programování v Pythonu - Práce se soubory - https://theses.cz/id/s216m2/795427
Vstup a výstup v Pythonu - https://macek.sandbox.cz/texty/python-tutorial-cz/tut/node9.html
Základy práce s textovými soubory v Pythonu - https://semiclon.com/blog-post5/
Práce s textovými soubory v Pythonu (MUNI) - https://is.muni.cz/el/sci/podzim2017/C2184/um/lecture07.pdf
Python - Práce se soubory (sallyx.org) - https://www.sallyx.org/sally/python/python10.php
Zápis do textových souborů a ošetření chyb (ITnetwork) - https://www.itnetwork.cz/python/soubory/prace-s-textovymi-soubory-v-pythonu
Python: Čtení a zápis do souboru s automatickým zavíráním - http://home.zcu.cz/~pbruha/pages/cviceni5.html
Práce se soubory: Čtení a zápis - https://ksvi.mff.cuni.cz/~peskova/vyuka/21-22_p1m/cv11/11_soubory.html
Příklady práce se soubory v Pythonu - http://mamut.spseol.cz/nozka/python/priklad_soubory/
Cvičení: Zápis a čtení kontaktů ze souboru - https://ucimeseit.cz/python-priklady/

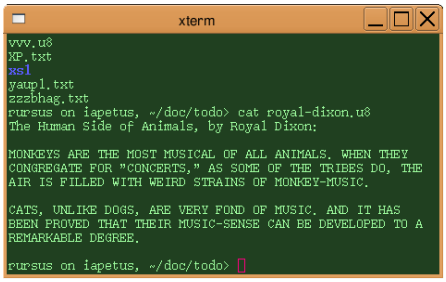
- Aktivita - Brainstorming (~10 minut)

2. Uvědomění (30 minut)

- **Základy práce s textovými soubory**

  Funkce open() a její parametry (2 minuty)

  Vysvětlím základní syntaxi `open(filename, mode)` s nejdůležitějšími módy: 'r' pro čtení (výchozí), 'w' pro zápis (přepíše obsah), 'a' pro připojování. Ukážu také parametr encoding='utf-8' pro správné zobrazení českých znaků. Zdůrazním, že open() vrací file objekt, se kterým můžeme pracovat.



Textový soubor
Zdroj: WikiMedia

```
# Základní použití
soubor = open('data.txt', 'r', encoding='utf-8')
```

  Metody čtení souborů (3 minuty)

  Demonstruji tři základní způsoby čtení:

  - `read()` - načte celý obsah jako jeden string
  - `readline()` - načte jeden řádek včetně znaku nového řádku
  - `readlines()` - načte všechny řádky do seznamu

```
# Čtení celého souboru
obsah = soubor.read()

# Čtení po řádcích
prvni_radek = soubor.readline()
vsechny_radky = soubor.readlines()
```

  Vysvětlím rozdíly mezi metodami a kdy kterou použít - read() pro malé soubory, readline() pro postupné zpracování, readlines() pro práci se seznamem řádků.

  Kontextový manažer with statement (3 minuty)

  Ukážu problém s nezavřenými soubory a představím elegantní řešení pomocí `with`:

```
# Bez kontextového manažera - problematické
soubor = open('data.txt', 'r')
obsah = soubor.read()
```

```
soubor.close()  # můžeme zapomenout

# S kontextovým manažerem - doporučené
with open('data.txt', 'r', encoding='utf-8') as soubor:
    obsah = soubor.read()
# soubor se automaticky zavře
```

Vysvětlím, že `with` automaticky zavře soubor i při chybě, což je bezpečnější přístup.

Praktická ukázka (2 minuty)

Společně vytvoříme jednoduchý příklad čtení textového souboru se jmény studentů:

```
Anna Nováková
Petr Svoboda
Jana Dvořáková
Tomáš Král
Eva Benešová
Martin Černý
Lucie Procházková
```

data.txt

První řádek obsahuje důležité informace o projektu.
Druhý řádek popisuje aktuální stav implementace.
Třetí řádek zmiňuje budoucí plány a vylepšení.
Čtvrtý řádek je jen pro ukázku, jak se pracuje s textem.

```
with open('studenti.txt', 'r', encoding='utf-8') as f:
    for radek in f:
        print(f"Student: {radek.strip()}")
```

Vysvětlím použití `strip()` pro odstranění bílých znaků a možnost iterace přímo přes file objekt.

- **Zápis do textových souborů a ošetření chyb**

Ukážu základní syntaxi zápisu do souboru pomocí metody write(). Metoda write() zapisuje řetězec do souboru a vrací počet zapsaných znaků. Důležité je otevřít soubor v režimu 'w' (write) nebo 'a' (append).

Function	Meaning
int bad ( )	Returns non – zero value if an invalid operation is attempted or any unrecoverable error has occurred. However, if it is zero (false value), it may be possible to recover from any other error reported and continue operation.
int eof ( )	Returns non – zero (true value) if end – of – file is encountered while reading ; otherwise returns zero (false value).
int fail ( )	Returns non – zero (true) when an input or output operation has failed.
int good ( )	Returns non – zero (true) if no error has occurred. This all the above functions are false. For instance, if **fin.good ( )** is true, everything is okay with the stream named as **fin** and we can proceed to perform I/O operations. When it returns zero, no further operations can be carried out.
clear ( )	Resets the error state so that further operations can be attempted.

Zpracování chyb
Zdroj: WikiMedia

```
# Základní zápis do souboru
with open('poznamky.txt', 'w') as soubor:
    soubor.write('První řádek\n')
    soubor.write('Druhý řádek\n')

# Zápis s proměnnými
jmeno = 'Pavel'
vek = 17
with open('student.txt', 'w') as soubor:
    soubor.write(f'Jméno: {jmeno}\n')
    soubor.write(f'Věk: {vek}\n')
```

Upozorním na rozdíl mezi režimy 'w' (přepíše celý soubor) a 'a' (připojí na konec). Metoda write() nepřidává automaticky znak nového řádku, musíme ho přidat ručně pomocí '\n'.

Zápis do textových souborů - metoda writelines() (2 minuty)

Vysvětlím metodu writelines(), která zapisuje seznam řetězců najednou. Na rozdíl od write() nepřidává automaticky oddělovače mezi prvky.

```
# Zápis seznamu řádků
radky = ['První úkol\n', 'Druhý úkol\n', 'Třetí úkol\n']
with open('ukoly.txt', 'w') as soubor:
    soubor.writelines(radky)

# Zápis bez '\n' - všechno na jeden řádek
slova = ['ahoj', 'světe', 'python']
with open('text.txt', 'w') as soubor:
    soubor.writelines(slova)  # výsledek: ahojsvětepython
```

Zdůrazním, že writelines() neformátuje data automaticky - pokud chceme každý prvek na novém řádku, musíme '\n' přidat sami.

Ošetření výjimek při práci se soubory (3 minuty)

Vysvětlím nejčastější chyby při práci se soubory a jak je ošetřit pomocí try-except bloku. Ukážu praktické příklady s konkrétními typy výjimek.

```
# Ošetření chyb při zápisu
try:
    with open('/neexistujici/cesta/soubor.txt', 'w') as soubor:
        soubor.write('Test')
except FileNotFoundError:
    print('Složka neexistuje!')
except PermissionError:
    print('Nemáte oprávnění k zápisu!')
except Exception as e:
    print(f'Neočekávaná chyba: {e}')

# Bezpečný zápis s kontrolou
def bezpecny_zapis(nazev_souboru, obsah):
    try:
        with open(nazev_souboru, 'w') as soubor:
            soubor.write(obsah)
        return True
    except (FileNotFoundError, PermissionError) as e:
        print(f'Chyba při zápisu: {e}')
        return False
```

Zmíním nejčastější výjimky: FileNotFoundError (neexistující cesta), PermissionError (nedostatečná oprávnění), OSError (obecné systémové chyby).

Praktické cvičení - zápis a ošetření chyb (2 minuty)

Studenti vytvoří program, který zapíše jejich jméno a tři oblíbené předměty do souboru s ošetřením chyb.

Úkol	Kód
Vytvořte soubor 'moje_data.txt'	Použijte with open() a režim 'w'
Zapište své jméno	Metoda write() s '\n'
Zapište tři předměty	Metoda writelines() se seznamem
Ošetřete chyby	try-except s FileNotFoundError

Studenti pracují samostatně, procházím mezi nimi a pomáhám s případnými problémy. Jeden student ukáže své řešení ostatním.

- **Práce s CSV soubory**

Úvod do CSV formátu (2 minuty) CSV (Comma-Separated Values) je jednoduchý formát pro ukládání tabulkových dat. Každý řádek představuje záznam, hodnoty jsou oddělené čárkami. Příklad: `jmeno,vek,trida` na prvním řádku, `Jan,17,4A` na druhém. CSV soubory lze otevřít v Excelu, LibreOffice Calc nebo textovém editoru. Výhody: jednoduchost, univerzálnost, malá velikost souborů.



Sloučení CSV souborů
Zdroj: WikiMedia

Import knihovny csv a základní čtení (3 minuty)

```
import csv

# Čtení CSV pomocí csv.reader()
with open('studenti.csv', 'r', encoding='utf-8') as soubor:
    reader = csv.reader(soubor)
    for radek in reader:
        print(radek)  # radek je seznam hodnot
```

Vysvětlím rozdíl oproti běžnému čtení textového souboru: csv.reader() automaticky rozdělí řádek podle oddělovače a vrátí seznam hodnot místo celého řetězce. Ukážu, jak přistupovat k jednotlivým sloupcům: `radek[0]` pro první sloupec, `radek[1]` pro druhý.

Zápis CSV pomocí csv.writer() (2 minuty)

```
import csv

# Zápis CSV pomocí csv.writer()
data = [
    ['Jan', '17', '4A'],
    ['Marie', '16', '4B'],
    ['Petr', '18', '4A']
]

with open('novy_soubor.csv', 'w', encoding='utf-8', newline='') as soubor:
    writer = csv.writer(soubor)
    writer.writerow(['jmeno', 'vek', 'trida'])  # hlavička
    writer.writerows(data)  # všechny řádky najednou
```

Zdůrazním parametr `newline=''` pro správné formátování na Windows systémech. Ukážu rozdíl mezi `writerow()` pro jeden řádek a `writerows()` pro více řádků.

Práce se slovníky - DictReader a DictWriter (2 minuty)

```
import csv

# Čtení jako slovníky
with open('studenti.csv', 'r', encoding='utf-8') as soubor:
    reader = csv.DictReader(soubor)
    for radek in reader:
        print(f"Student {radek['jmeno']} má {radek['vek']} let")

# Zápis slovníků
data = [
    {'jmeno': 'Anna', 'vek': '17', 'trida': '4C'},
    {'jmeno': 'Tomáš', 'vek': '16', 'trida': '4D'}
]

with open('slovniky.csv', 'w', encoding='utf-8', newline='') as soubor:
    writer = csv.DictWriter(soubor, fieldnames=['jmeno', 'vek', 'trida'])
    writer.writeheader()  # zapíše hlavičku
    writer.writerows(data)
```

Výhoda slovníků: přístup k datům podle názvu sloupce místo indexu, čitelnější kód.

Nastavení parametrů CSV (1 minuta) Ukážu nejčastější parametry: `delimiter=';'` pro středník místo čárky (český standard), `quotechar='"'` pro uvozovky kolem hodnot s oddělovačem. Příklad: `csv.reader(soubor, delimiter=';')`. Zmíním, že Excel v češtině často používá středník jako oddělovač.

3. Reflexe (5 minut)

- **Praktické cvičení: Správa jednoduchého seznamu úkolů**

  Rychlé procvičení CSV operací Studenti dostanou připravený CSV soubor `ukoly.csv` s jednoduchým seznamem úkolů (sloupce: název, priorita, hotovo). Úkol: načíst soubor, přidat jeden nový úkol a uložit zpět. Použití `csv.DictReader` pro čtení a `csv.DictWriter` pro zápis.

```
import csv

# Načtení existujících úkolů
ukoly = []
with open('ukoly.csv', 'r', encoding='utf-8') as f:
    reader = csv.DictReader(f)
    for radek in reader:
        ukoly.append(radek)

# Přidání nového úkolu
novy_ukol = {'název': 'Dokončit domácí úkol', 'priorita': 'vysoká', 'hotovo': 'ne'}
ukoly.append(novy_ukol)

# Uložení zpět
```

```
with open('ukoly.csv', 'w', encoding='utf-8', newline='') as f:
    writer = csv.DictWriter(f, fieldnames=['název', 'priorita', 'hotovo'])
    writer.writeheader()
    writer.writerows(ukoly)
```

Samostatná práce studentů (3 minuty) Každý student upraví kód tak, aby místo pevně daného úkolu přidal vlastní úkol podle svého výběru. Experimentování s různými hodnotami priorit a stavů dokončení.

Rychlá kontrola výsledků Jeden nebo dva studenti ukážou svůj výsledný CSV soubor - ověření, že se data správně uložila a načetla.