Soubory a serializace - Ukládání a načítání dat, formáty souborů

Soubory obecně

Soubor je pojmenovaná kolekce dat, která je uložena na médiu (disk). Z technického hlediska jde o nějakou posloupnost dat (sekvence bajtů), které jsou nějakým způsobem organizovány a interpretován.

Charakteristika souboru

Název a přípona - identifikuje soubor a naznačuje jeho formát
Metadata - informace o souboru (vytvoření, velikost, práva….)
Obsah - samotná data v souboru

Vlastnosti souborů

Každý soubor má jedinečnou cestu v souborovém systému. Soubory jsou organizovány v hierarchické struktuře adresářů (složky). Přístup k souborům je řízen oprávněními

Soubory v Pythonu

V pythonu jsou soubory reprezentovaný jako objekty souborů. Používá se funkce open(. Máme několik režimů na otevírání souborů:
čtení (‘r’)
zapisování (‘w’)
přidávání (‘a’)
aktualizace (‘+’)
….

```
with open("data.txt", "r") as soubor:
   obsah = soubor.read()
   print(obsah)
```

Serializace
Proces převodu objektů nebo datových struktur do formátu, který bude jednoduše přenositelný. Typicky se používá pro ukládání stavu aplikace, načtení konfigurace a nebo přenos dat mezi procesy v PC. Během serializace jsou data přenášena jako série bytů, které se následně zase deserializují do původní podoby. Data, která jsou totiž po vypnutí programu na stacku nebo na heapu, se ztratí. Serializace nám umožňuje tyto objekty nebo datové struktury zachovat.

V pythonu nabízí několik věstavěných modulů pro serializaci
pickle - nativní protokol
json
xml
marshall
….

Serializace objektů v pythonu
Ukládání dat funguje v Pythonu poměrně jednoduše. Na rozdíl od Javy, kde musíme implementovat rozhraní Serializable, Python umožňuje serializovat objekty přímo pomocí modulu `pickle`. Většina Python objektů je serializovatelná automaticky bez nutnosti speciálních úprav.

```
import traceback
import pickle
class Zamestnanec:
   def __init__(self):
       self.jmeno = None
       self.adresa = None
       self.ssn = None
       self.cislo = None


# Vytvoření a inicializace instance
try:
   e = Zamestnanec()
   e.jmeno = "Ryan Ali"
   e.adresa = "Phokka Kuan, Ambehta Peer"
   e.ssn = 11122333
   e.cislo = 101

   # Serializace objektu
   try:
       with open("zamestnanec.pkl", "wb") as file_out:  # wb = write binary
           pickle.dump(e, file_out)
       print("Data jsou serializována v zamestnanec.pkl")
```

```
    except Exception as ex:
        print(f"Neočekávaná chyba při serializaci: {ex}")
        traceback.print_exc()
except Exception as init_err:
    print(f"Chyba při vytvoření nebo inicializaci objektu: {init_err}")
    traceback.print_exc()
```

Do souboru zamestnanec.pkl se uložila binární reprezentace objektu e (Zamestnanec). Při deserializaci vytvoříme novou instanci Zamestnanec a ta bude obsahovat všechny hodnoty. Je nutné ještě dodržet datové typy a pořadí prvků

Deserialiazce

V případě deserialiazce načítáme postupně binární informace o objektu pomocí pickle přes funkci load()

```
import pickle


# Definice třídy Zamestnanec (musí být stejná jako při serializaci)
class Zamestnanec:
    def __init__(self):
        self.jmeno = None
        self.adresa = None
        self.ssn = None
        self.cislo = None


# Začínáme s prázdnou referencí
e = None

try:
    # Otevření souboru pro čtení v binárním režimu
    fileIn = open("zamestnanec.pkl", "rb")

    # Deserializace objektu
    e = pickle.load(fileIn)

    # Zavření souboru
    fileIn.close()

    # Použití deserializovaného objektu
    print(f"Jméno: {e.jmeno}")
    print(f"Adresa: {e.adresa}")
    print(f"SSN: {e.ssn}")
    print(f"Číslo: {e.cislo}")

except FileNotFoundError:
    print("Soubor nebyl nalezen")
    import traceback

    traceback.print_exc()

except ModuleNotFoundError as c:
    print(f"Třída Zamestnanec nebyla nalezena")
    c.print_stack_trace()

except Exception as i:
    import traceback

    traceback.print_exc()
```

Je nutné mít vše ošetřené v try a catch blocích, které nám zachycují případné problémy se čtením ze souboru - neexistující soubor, neexistujicí instance atd…

Formáty souborů

Formáty představují standardizované způsoby ukládání a organizace dat. Definují nám strukturu v jaké jsou informace uloženy, aby mohly být správně interpretovány programy

Textové soubory

Obsahují čitelný text kódovaný podle znakových sad (ASCII, UTF-8).... Jsou pro nás pro lidi jednoduše čitelné a lze je lehce otevřít nebo upravit přes editory.
.txt, .csv, .xml, .json…..

Binární soubory

Obsahují data v nečitelné, binární podobě. Nejsou čitelné lidským okem a jsou skládány do nějakého formátu pro účely komplikace nebo interpretace. Vyznačují se také tím, že po přečtení souboru dostane přesně ty samá data, jako se nachází v souboru a nejsou žádné konverze, jako u textových, kde se následně bajty dekódují.

.pkl, .ser, .zip, .rar

Zvukové a video formáty

Zvukové formáty jsou speciální druh binárních souborů. Data můžou být v komprimované i nekomprimovaném formátu. Tyto soubory obsahují informace o tom, jaké hodnoty amplitudy se mění v čase.
Jsou uloženy jako binární čísla.

.mp3, .mp4, .jpg