Серилизација објеката

Када је потребно сачувати велике количине комплексних података попут речника или сетова у виду фајлова (објеката), најједноставнији начин је извршити серилизацију објекта.
Када је објекат серилизован, он је конвертован у стрим (низ) бајтова који се лако смештају у један фајл.
У Пајтону, процес серилизације објекта се назива пикловање (pickling).
Стандардна Пајтон библиотека садржи модул pickle који има функције за серилизацију објеката.

Кораци при серилизацији објеката

После импортовања модула picke, изводе се следећи кораци за серилизацију објекта:

- Отвара се фајл за бинарно уписивање
- Позове се метода dump за пикловање објекта и његов упис у одређени фајл
- Када су пикловани сви жељени објекти и сачувани у фајл, фајл се затвара

Да би се отворио фајл за бинарно уписивање, користити се 'wb' за мод при позивања open функције.
Пример: отворити фајл под именом moji_podaci.dat за бинарно уписивање:
izlazni_fajl = open('moji_podaci.dat', 'wb')
Када је отворен фајл за бинарно уписивање, позива се функција dump.
Општи формат функције dump:
pickle.dump(objekat, fajl)
О овом формату, objekat је промењива која указује на објекат који треба пикловати, а fajl је промењива која указује на фајл објекат.
Када се функција изврши, објекат на који указује objekat ће бити серилизован и уписан у фајл (могу се пикловати листе, торке, речници, сетови, стрингови, цели бројеви и децимални бројеви).
У један фајл се пикловати неограничени број објеката.
Када се заврши са уписом, позива се метода close за затварање фајла.
Пример пикловања речника:

```
import pickle
telefonski_imenik = {'Mica' : '555-1111', 'Ana' : '555-2222', 'Deki' : '555-3333'}
izlazni_fajl = open('telefonski_imenik.dat', 'wb')
pickle.dump(telefonski_imenik, izlazni_fajl)
izlazni_fajl.close()
```

Рад са серилизованим објектом

Ако је објекат пиклован, у неком моменту је потребно депикловати тај објекат следећим корацима:

- Отвара се фајл за бинарно читање
- Позива се функција load за добијање објекта из фајла и његово депкловање
- После депикловања свих објеката из фајла, фајл се затвара

За отварање фајла са бинарним подацима, користи се мод 'rb' приликом употребе функције open.
Пример: отворити фајл под именом moji_podaci.dat за бинарно читање:
ulazni_fajl = open('moji_podaci.dat', 'rb')
Када се отвори фајл за бинарно читање, позива се функција load.
Општи формат функције load:
objekat = pickle.load(fajl)
Овде је objekat промењива а fajl је промењива која се односи на фајл објекат.
После извршења функције, промењива objekat ће указивати на објекат који је извучен из фајла и депиклован.
Из фајла се може депикловати неограничени број објеката.
После тога, позива се метода close за затварање фајла.
Пример депикловања речника:

```
ulazni_fajl = open('telefonski_imenik.dat', 'rb')
telefonski_imenik = pickle.load(ulazni_fajl)
print(telefonski_imenik)
ulazni_fajl.close()
```

Види се да се прво отвара фајл telefonski_imenik.dat за бинарно читање; са функцијом load се добија и депиклује објекат из telefonski_imenik.dat фајла; добијени објекат се додељује промењивој telefonski_imenik.

На екрану се добија: {'Mica' : '555-1111', 'Ana' : '555-2222', 'Deki' : '555-3333'}
Пример: унос података о особама у речник, при чему се речник пиклује у бинарном облику у посебан фајл

```
import pickle

def main():
    ponovo = 'd'
    izlazni_fajl = open('info.dat', 'wb')
    while ponovo.lower() == 'd':
        sacuvani_podaci(izlazni_fajl)
        ponovo = input('Unos jos podataka? (d/n): ')

    izlazni_fajl.close()

def sacuvani_podaci(fajl):
    osoba = {}
    osoba['ime'] = input('Ime: ')
    osoba['godine'] = int(input('Godine: '))
    osoba['tezina'] = float(input('Tezina: '))
    pickle.dump(osoba, fajl)

main()
```

Пример: приказ података о особама у виду речника, при чему се речник депиклује у бинарном облику из посебног фајла

```
import pickle

def main():
    kraj_fajla = False
    ulazni_fajl = open('info.dat', 'rb')
    while not kraj_fajla:
        try:
            osoba = pickle.load(ulazni_fajl)
            prikazi_podatke(osoba)
        except EOFError:
            kraj_fajla = True

    ulazni_fajl.close()

def prikazi_podatke(person):
    print('Ime:', person['ime'])
    print('Godine:', person['godine'])
    print('Tezina:', person['tezina'])
    print()

main()
```