

Programowanie proceduralne

INP001210WL

rok akademicki 2020/21

semestr letni

Wykład 2

Karol Tarnowski

karol.tarnowski@pwr.edu.pl

L-1 p. 220



Plan prezentacji (1)

- Odczyt i zapis plików - wprowadzenie
- Otwarcie pliku
- Zapis danych
- Zamknięcie pliku
- Odczyt danych



Plan prezentacji (2)

- Obsługa znaku nowego wiersza przy zapisie i odczycie
- Obsługa danych liczbowych - konwersja typów
- Obsługa zapisu i odczytu z wykorzystaniem pętli



Odczyt i zapis plików

- Programy komputerowe nie muszą bazować tylko na danych podanych bezpośrednio przez użytkownika, mogą korzystać z danych zapisanych w plikach
- Przykłady programów zapisujących dane w plikach:
 - edytory tekstu,
 - edytory graficzne,
 - arkusze kalkulacyjne,
 - gry,
 - przeglądarki WWW.



Odczyt i zapis plików

Aby użyć w programie pliku, należy:

1. Otworzyć plik - po otwarciu pliku następuje utworzenie połączenia między programem a plikiem (plik można otworzyć do zapisu lub do odczytu)
2. Przetworzyć plik - program zapisuje lub odczytuje dane
3. Zamknąć plik - gdy program zakończy przetwarzanie pliku, należy zamknąć połączenie między plikiem a programem



Odczyt i zapis plików

Typy plików:

- pliki tekstowe - dane tekstowe zapisane w kodowaniu ASCII lub Unicode
- pliki binarne - dane, które nie zostały skonwertowane na postać tekstu



Odczyt i zapis plików

Metody dostępu do plików:

- dostęp sekwencyjny - dane są przetwarzane po kolei - od początku pliku do końca
- dostęp swobodny - można przeskoczyć do konkretnego miejsca w pliku bez konieczności przetwarzania wcześniejszych danych



Otworzenie pliku

- Funkcja `open()` tworzy obiekt reprezentujący plik i wiąże go z konkretnym plikiem na dysku
- Ogólna postać wywołania

```
zmienna_pliku = open(nazwa_pliku, tryb)
```

- *zmienna_pliku* - to zmienna reprezentująca plik
- *nazwa_pliku* - ciąg tekstowy określający nazwę pliku
- *tryb* - ciąg tekstowy określający tryb



Otworzenie pliku

- Wybrane tryby otwarcia plików

tryb	opis
'r'	plik otwarty tylko do odczytu - zawartość nie może być modyfikowana
'w'	plik otwarty do zapisu - zawartość istniejącego pliku zostanie usunięta, jeśli plik nie istnieje zostanie utworzony
'a'	plik otwarty do zapisu nowych danych - wszystkie nowe dane zostaną umieszczone na końcu pliku, jeśli plik nie istnieje będzie utworzony



Zapis danych w pliku

- Funkcja `write()` jest metodą obiektu reprezentującego plik
- Ogólna postać wywołania metody `write()`

```
zmienna_pliku.write(ciąg_tekstowy)
```

- Przykładowo

```
file.write('Jan Kowalski')
```

```
name = 'Jan Kowalski'
```

```
file.write(name)
```



Zamknięcie pliku

- Funkcja `close ()` jest metodą obiektu reprezentującego plik
- Ogólna postać wywołania metody `close ()`

```
zmienna_pliku.close ()
```

Zapis danych w pliku

Przykład

```
01_file_write.py
File Edit Format Run Options Window Help
# Przykładowy program wykonujący
# zapis do pliku.

def main():
    #otwarcie pliku "nobliści.txt" do zapisu
    outfile = open('nobliści.txt', 'w')

    #zapisanie w pliku imion i nazwisk
    #polskich laureatów Nagrody Nobla
    #w dziedzinie literatury
    outfile.write('Maria Skłodowska-Curie\n')
    outfile.write('Henryk Sienkiewicz\n')
    outfile.write('Władysław Reymont\n')
    outfile.write('Czesław Miłosz\n')
    outfile.write('Wisława Szymborska\n')
    outfile.write('Olga Tokarczuk\n')

    outfile.close()

main()
```

Zapis danych w pliku

Przykład

```
01_file_write.py
File Edit Format Run Options Window Help
# Przykładowy program wykonujący
# zapis do pliku.
```

```
noblisci.txt — Notatnik
Plik Edycja Format Widok Pomoc
Maria Skłodowska-Curie
Henryk Sienkiewicz
Władysław Reymont
Czesław Miłosz
Wisława Szymborska
Olga Tokarczuk
```

```
outfile.close()
```

```
main()
```



Odczyt danych z pliku

- Funkcja `read()` jest metodą obiektu reprezentującego plik, która pozwala wczytać całą zawartość pliku do pamięci
- Ogólna postać wywołania metody `read()`

```
contents = zmienna_pliku.read()
```



Odczyt danych z pliku

- Funkcja `readline()` jest metodą obiektu reprezentującego plik, która pozwala odczytać jedną linię z pliku
- Ogólna postać wywołania metody `readline()`

```
contents = zmienna_pliku.readline()
```



Odczyt danych z pliku

```
02_file_read.py
File Edit Format Run Options Window Help
# Przykładowy program wykonujący
# odczyt z pliku.

def main():
    #otwarcie pliku "nobliści.txt" do odczytu
    infile = open('nobliści.txt','r')

    #wczytanie zawartości pliku metodą read()
    file_contents = infile.read()

    #zamknięcie pliku
    infile.close()

    #wyświetlenie danych wczytanych do pamięci
    print(file_contents)

main()
```




Odczyt danych z pliku

```
02_file_read.py
File Edit Format Run Options Window Help
# Przykładowy program wykonujący
# odczyt z pliku.

def main():
    #otwarcie pliku "nobliści.txt" do odczytu
    infile = open('nobliści.txt', 'r')

    #wczytanie zawartości pliku metodą read()
    file_contents = infile.read()

    #zamknięcie pliku
    infile.close()

    #wyświetlenie danych wczytanych do pamięci
    print(file_contents)

main()
```



Odczyt danych z pliku

```
02_file_read.py
File Edit Format Run Options Window Help
# Przykładowy program wykonujący
# odczyt z pliku.
def
    Maria Skłodowska-Curie
    Henryk Sienkiewicz
    Władysław Reymont
    Czesław Miłosz
    Wisława Szymborska
    Olga Tokarczuk

    >>>

    print(file_contents)

main()
```



Odczyt danych z pliku

```
03_file_read.py
File Edit Format Run Options Window Help
# Przykładowy program wykonujący
# odczyt z pliku.

def main():
    #otwarcie pliku "nobliści.txt" do odczytu
    infile = open('nobliści.txt','r')

    #wczytanie zawartości pliku metodą readline()
    line1 = infile.readline()
    line2 = infile.readline()
    line3 = infile.readline()
    line4 = infile.readline()
    line5 = infile.readline()
    line6 = infile.readline()

    #zamknięcie pliku
    infile.close()

    #wyświetlenie danych wczytanych do pamięci
    print(line1)
    print(line2)
    print(line3)
    print(line4)
```



Odczyt danych z pliku

```
03_file_read.py
File Edit Format Run Options Window Help
# Przyj
# odczy
def ma
#ot
int
    Władysław Reymont
#wc
lin
lin
lin
lin
lin
lin
    Wisława Szymborska
lin
#za
int
#wy
    >>>
pri
print(line2)
print(line3)
print(line4)
```



Znak nowego wiersza

- Znak nowego wiersza służy do rozdzielania linii tekstu w pliku
- Ciągi tekstowe mają metodę `rstrip()`, którą można wykorzystać do usunięcia określonych znaków tekstowych (np. znaku nowego wiersza) z końca ciągu



Znak nowego wiersza

```
04_file_write.py
File Edit Format Run Options Window Help
# Przykładowy program wykonujący
# zapis danych tekstowych w pliku
# dodając znak nowego wiersza.

def main():
    print('Podaj imiona trójki przyjaciół.')
    name1 = input('Przyjaciel #1: ')
    name2 = input('Przyjaciel #2: ')
    name3 = input('Przyjaciel #3: ')

    #otwarcie pliku "imiona.txt" do zapisu
    outfile = open('imiona.txt','w')
    #zapisanie 3 linii tekstu
    #(dołączony znak nowej linii)
    outfile.write(name1 + '\n')
    outfile.write(name2 + '\n')
    outfile.write(name3 + '\n')
    #zamknięcie pliku
    outfile.close()

main()
```



Znak nowego wiersza

```
04_file_write.py
File Edit Format Run Options Window Help
# Przykładowy program wykonujący
# zapis danych tekstowych w pliku
# dodając znak nowego wiersza.

def main():
    print('Podaj imiona trójki przyjaciół.')
    name1 = input('Przyjaciel #1: ')
    name2 = input('Przyjaciel #2: ')
    name3 = input('Przyjaciel #3: ')

    #otwarcie pliku "imiona.txt" do zapisu
    outfile = open('imiona.txt','w')
    #zapisanie 3 linii tekstu
    #(dołączony znak nowej linii)
    outfile.write(name1 + '\n')
    outfile.write(name2 + '\n')
    outfile.write(name3 + '\n')
    #zamknięcie pliku
    outfile.close()

main()
```



Znak nowego wiersza

```
05_file_rstrip.py
File Edit Format Run Options Window Help
# Przykładowy program wykonuje
# odczyt danych tekstowych z pliku (readline),
# i usuwa znak nowej linii (rstrip).

def main():
    print('Program wczytuje imiona trójki przyjaciół.')

    #otwarcie pliku "imiona.txt" do odczytu
    infile = open('imiona.txt','r')
    #odczytanie 3 linii tekstu
    line1 = infile.readline()
    line2 = infile.readline()
    line3 = infile.readline()
    #usunięcie znaków nowej linii
    line1 = line1.rstrip('\n')
    line2 = line2.rstrip('\n')
    line3 = line3.rstrip('\n')
    #zamknięcie pliku
    infile.close()

    print('Przyjaciel #1:',line1)
    print('Przyjaciel #2:',line2)
    print('Przyjaciel #3:',line3)

main()
```




Znak nowego wiersza

```
05_file_rstrip.py
File Edit Format Run Options Window Help
# Przykładowy program wykonuje
# odczyt danych tekstowych z pliku (readline),
# i usuwa znak nowej linii (rstrip).

def main():
    print('Program wczytuje imiona trójki przyjaciół.')

    #otwarcie pliku "imiona.txt" do odczytu
    infile = open('imiona.txt','r')
    #odczytanie 3 linii tekstu
    line1 = infile.readline()
    line2 = infile.readline()
    line3 = infile.readline()
    #usunięcie znaków nowej linii
    line1 = line1.rstrip('\n')
    line2 = line2.rstrip('\n')
    line3 = line3.rstrip('\n')
    #zamknięcie pliku
    infile.close()

    print('Przyjaciel #1:',line1)
    print('Przyjaciel #2:',line2)
    print('Przyjaciel #3:',line3)

main()
```



Zapis danych liczbowych

- Funkcja `write()` zapisuje ciągi znaków, zatem dane liczbowe przed zapisaniem do pliku funkcją `write()` należy skonwertować na łańcuch znakowy funkcją `str()`



Zapis danych liczbowych

```
06_file_write_numbers.py - C:\Users\karol\Desktop\201920\python-examples\pliki\06_fi...
File Edit Format Run Options Window Help
# Przykładowy program pokazuje
# zamianę danych liczbowych na tekst
# przed zapisaniem ich do pliku.

def main():
    print('Podaj trzy liczby całkowite')
    num1 = int(input('Pierwsza liczba: '))
    num2 = int(input('Druga liczba: '))
    num3 = int(input('Trzecia liczba: '))

    #otwarcie pliku "liczby.txt" do zapisu
    outfile = open('liczby.txt','w')
    #zapisanie 3 liczb po skonwertowaniu
    #na łańcuch tekstowy
    outfile.write(str(num1) + '\n')
    outfile.write(str(num2) + '\n')
    outfile.write(str(num3) + '\n')
    #zamknięcie pliku
    outfile.close()

main()
```



Zapis danych liczbowych

```
06_file_write_numbers.py - C:\Users\karol\Desktop\201920\python-examples\pliki\06_fi...
File Edit Format Run Options Window Help
# Przykładowy program pokazuje
# zamianę danych liczbowych na tekst
# przed zapisaniem ich do pliku.

def main():
    print('Podaj trzy liczby całkowite')
    num1 = int(input('Pierwsza liczba: '))
    num2 = int(input('Druga liczba: '))
    num3 = int(input('Trzecia liczba: '))

    #otwarcie pliku "liczby.txt" do zapisu
    outfile = open('liczby.txt','w')
    #zapisanie 3 liczb po skonwertowaniu
    #na łańcuch tekstowy
    outfile.write(str(num1) + '\n')
    outfile.write(str(num2) + '\n')
    outfile.write(str(num3) + '\n')
    #zamknięcie pliku
    outfile.close()

main()
```



Odczyt danych liczbowych

```
07_file_read_numbers.py
File Edit Format Run Options Window Help
# Przykładowy program pokazuje
# odczyt danych liczbowych z pliku
# zapisanych w postaci tekstu.

def main():
    print('Program wczytuje trzy liczby całkowite')
    print('z pliku "liczby.txt" i oblicza ich sumę.')

    #otwarcie pliku "liczby.txt" do odczytu
    infile = open('liczby.txt','r')
    #odczytanie 3 linii tekstu
    #zamiana liczb na int
    num1 = int(infile.readline())
    num2 = int(infile.readline())
    num3 = int(infile.readline())
    #zamknięcie pliku
    infile.close()

    #obliczenie sumy liczb
    total = num1 + num2 + num3

    #wyświetlenie liczb i stosowanego komunikatu
    print('Odczytane liczby to:',num1, num2, num3)
    print('Suma liczb wynosi:', total)

main()
```



Odczyt danych liczbowych

```
07_file_read_numbers.py
File Edit Format Run Options Window Help
# Przykładowy program pokazuje
# odczyt danych liczbowych z pliku
# zapisanych w postaci tekstu.

def main():
    print('Program wczytuje trzy liczby całkowite')
    print('z pliku "liczby.txt" i oblicza ich sumę.')

    #otwarcie pliku "liczby.txt" do odczytu
    infile = open('liczby.txt','r')
    #odczytanie 3 linii tekstu
    #zamiana liczb na int
    num1 = int(infile.readline())
    num2 = int(infile.readline())
    num3 = int(infile.readline())
    #zamknięcie pliku
    infile.close()

    #obliczenie sumy liczb
    total = num1 + num2 + num3

    #wyświetlenie liczb i stosowanego komunikatu
    print('Odczytane liczby to:',num1, num2, num3)
    print('Suma liczb wynosi:', total)

main()
```

Przetwarzanie plików za pomocą pętli

- Zwykle pliki zawierają duże ilości informacji, które są przetwarzane z wykorzystaniem pętli

Przetwarzanie plików za pomocą pętli

```
08_file_write_grades.py
File Edit Format Run Options Window Help
# Przykładowy program pokazuje
# zapis danych liczbowych do pliku
# z wykorzystaniem pętli.

def main():
    print('Program zapisuje do pliku "oceny.txt"')
    print('oceny podane przez użytkownika.')

    number = int(input('Ile ocen chcesz podać? '))

    #otwarcie pliku "oceny.txt" do zapisu
    outfile = open('oceny.txt','w')

    #pobranie od użytkownika ocen
    #i zapisanie ich do pliku.
    for count in range(1,number+1):
        grade = float(input(
            'Podaj ' + str(count) + ' ocenę: '))
        outfile.write(str(grade) + '\n')

    #zamknięcie pliku
    outfile.close()

main()
```




Przetwarzanie plików za pomocą pętli

```
08_file_write_grades.py
File Edit Format Run Options Window Help
# Przykładowy program pokazuje
# zapis danych liczbowych do pliku
# z wykorzystaniem pętli.

def main():
    print('Program zapisuje do pliku "oceny.txt"')
    print('oceny podane przez użytkownika.')

    number = int(input('Ile ocen chcesz podać? '))

    #otwarcie pliku "oceny.txt" do zapisu
    outfile = open('oceny.txt','w')

    #pobranie od użytkownika ocen
    #i zapisanie ich do pliku.
    for count in range(1,number+1):
        grade = float(input(
            'Podaj ' + str(count) + ' ocenę: '))
        outfile.write(str(grade) + '\n')

    #zamknięcie pliku
    outfile.close()

main()
```



Przetwarzanie plików za pomocą pętli

- Funkcja `readline()` po osiągnięciu końca pliku zwraca pusty łańcuch



Przetwarzanie plików za pomocą pętli

```
09_file_read_grades.py
File Edit Format Run Options Window Help

# Przykładowy program pokazuje odczyt
# danych liczbowych do pliku
# z wykorzystaniem pętli.
# Funkcja readline() zwraca pusty łańcuch
# jeśli osiągnięty zostanie koniec pliku.

def main():
    print('Program odczytuje oceny')
    print('z pliku "oceny.txt".')

    #otwarcie pliku "oceny.txt" do zapisu
    infile = open('oceny.txt','r')

    #odczytanie z pliku linii tekstu
    line = infile.readline()

    #dopóki linia nie jest pusta
    while line != '':
        #skonwertuj tekst na liczbę
        amount = float(line)
        #wyświetl ocenę
        print(format(amount, '.1f'))
        #wczytaj kolejną linię
        line = infile.readline()

    #zamknięcie pliku
    infile.close()

main()
```

Przetwarzanie plików za pomocą pętli

```
09_file_read_grades.py
File Edit Format Run Options Window Help

# Przykładowy program pokazuje odczyt
# danych liczbowych do pliku
# z wykorzystaniem pętli.
# Funkcja readline() zwraca pusty łańcuch
# jeśli osiągnięty zostanie koniec pliku.

def main():
    print('Program odczytuje oceny')
    print('z pliku "oceny.txt".')

    #otwarcie pliku "oceny.txt" do zapisu
    infile = open('oceny.txt','r')

    #odczytanie z pliku linii tekstu
    line = infile.readline()

    #dopóki linia nie jest pusta
    while line != '':
        #skonwertuj tekst na liczbę
        amount = float(line)
        #wyświetl ocenę
        print(format(amount, '.1f'))
        #wczytaj kolejną linię
        line = infile.readline()

    #zamknięcie pliku
    infile.close()

main()
```



Przetwarzanie plików za pomocą pętli

- Do odczytu danych można użyć również pętli `for`
- Obiekt reprezentujący plik udostępnia iterator, który może być wykorzystany w pętli `for`



Przetwarzanie plików za pomocą pętli

```
10_file_read_grades.py
File Edit Format Run Options Window Help

# Przykładowy program pokazuje odczyt
# danych liczbowych do pliku
# z wykorzystaniem pętli for.
# Pętla for "przechodzi" przez plik
# dzięki temu, że obiekt reprezentujący
# plik udostępnia iterator.

def main():
    print('Program odczytuje oceny')
    print('z pliku "oceny.txt".')

    #otwarcie pliku "oceny.txt" do zapisu
    infile = open('oceny.txt','r')

    #dla kolejnych linii pliku
    for line in infile:
        #skonwertuj tekst na liczbę
        amount = float(line)
        #wyświetl ocenę
        print(format(amount, '.1f'))

    #zamknięcie pliku
    infile.close()

main()
```

Przetwarzanie plików za pomocą pętli

```
10_file_read_grades.py
File Edit Format Run Options Window Help

# Przykładowy program pokazuje odczyt
# danych liczbowych do pliku
# z wykorzystaniem pętli for.
# Pętla for "przechodzi" przez plik
# dzięki temu, że obiekt reprezentujący
# plik udostępnia iterator.

def main():
    print('Program odczytuje oceny')
    print('z pliku "oceny.txt".')

    #otwarcie pliku "oceny.txt" do zapisu
    infile = open('oceny.txt','r')

    #dla kolejnych linii pliku
    for line in infile:
        #skonwertuj tekst na liczbę
        amount = float(line)
        #wyświetl ocenę
        print(format(amount, '.1f'))

    #zamknięcie pliku
    infile.close()

main()
```



Podsumowanie (1)

- Funkcje pozwalające na otwarcie i zamknięcie plików:
 - `open()`
 - `close()`
- Funkcje pozwalające na zapis i odczyt danych tekstowych:
 - `write()`
 - `read()`
 - `readline()`



Podsumowanie (2)

- Funkcja `rstrip()` służąca do pracy z łańcuchami znakowymi
- Wykorzystanie pętli `while` i `for` do pracy z plikami