

Sprawozdanie z Laboratorium nr 2

Oskar Wójs 73167

Zadanie 1

Cel zadania: Stworzenie programu, który na podstawie liczby zdobytych punktów określa wynik egzaminu studenta.

Kod programu:

```
punkty = int(input("Podaj liczbę punktów: "))

if punkty > 80:
    print("Egzamin zaliczony w terminie 0!")
elif punkty >= 50:
    print("Masz możliwość poprawy wyniku.")
else:
    print("Musisz poprawić egzamin.")
```

Wynik programu: Dla punktów = 75 wynik to: Masz możliwość poprawy wyniku.

Wnioski: Program poprawnie rozpoznaje zakresy punktów i wyświetla odpowiednie komunikaty.

Zadanie 2

Cel zadania: Uporządkowanie trzech liczb od najmniejszej do największej bez użycia funkcji sortujących.

Kod programu:

```
x = int(input("Podaj x: "))
y = int(input("Podaj y: "))
z = int(input("Podaj z: "))

if x > y:
    x, y = y, x
if y > z:
    y, z = z, y
if x > y:
    x, y = y, x
```

```
print("Posortowane liczby:", x, y, z)
```

Wynik programu: Dla danych 7, 2, 5 wynik to: 2, 5, 7.

Wnioski: Zadanie pokazało, jak można wykonać sortowanie prostą metodą porównań.

Zadanie 3

Cel zadania: Sprawdzenie, czy plik ma rozszerzenie '.xlsx'.

Kod programu:

```
nazwa_pliku = "Raport_maj.xlsx"
```

```
if nazwa_pliku.endswith(".xlsx"):
    print("Tak")
else:
    print("Nie")
```

Wynik programu: Tak

Wnioski: Metoda `endswith()` pozwala sprawdzić rozszerzenie pliku w prosty sposób.

Zadanie 5a

Cel zadania: Odczytanie pliku `notowania_gieldowe.txt` i wyświetlenie jego zawartości.

Kod programu:

```
with open("notowania_gieldowe.txt", "r") as plik:
    for linia in plik:
        print(linia.strip())
```

Wynik programu: Wyświetlenie zawartości pliku z notowaniami spółek giełdowych.

Wnioski: Instrukcja `with open()` pozwala w prosty sposób odczytywać dane z pliku linia po linii.

Zadanie 5b

Cel zadania: Dopisanie nowej spółki do pliku i ponowne wyświetlenie jego zawartości.

Kod programu:

```
with open("notowania_gieldowe.txt", "a") as plik:  
    plik.write("\nALR, 113")
```

```
with open("notowania_gieldowe.txt", "r") as plik:  
    for linia in plik:  
        print(linia.strip())
```

Wynik programu: Do pliku dodano spółkę ALR, 113 i wyświetlono całą zawartość pliku.

Wnioski: Tryb 'a' w funkcji open() pozwala dopisywać dane bez nadpisywania istniejących.

Zadanie 6

Cel zadania: Sprawdzenie, czy wprowadzona litera jest duża czy mała.

Kod programu:

```
litera = input("Podaj literę: ")  
  
if litera.isupper():  
    print("Litera jest duża.")  
elif litera.islower():  
    print("Litera jest mała.")  
else:  
    print("To nie jest litera.")
```

Wynik programu: Dla litery 'A' wynik to: Litera jest duża.

Wnioski: Metody isupper() i islower() są przydatne do analizy znaków.

Zadanie 7

Cel zadania: Sprawdzenie poprawności hasła pod względem długości i obecności znaku '!'.

Kod programu:

```
haslo = "pk47!jy0893"  
  
if len(haslo) == 11 and "!" in haslo:  
    print("Hasło jest poprawne")  
else:  
    print("Hasło jest niepoprawne")
```

Wynik programu: Hasło jest poprawne.

Wnioski: Użycie operatora 'in' oraz funkcji len() pozwala łatwo weryfikować złożone warunki logiczne.

Zadanie 8

Cel zadania: Wyodrębnienie fragmentów tekstu za pomocą operatora wycinania.

Kod programu:

```
text = "Studiuje-Informatykę"

print("Pierwsze trzy znaki:", text[:3])
print("Ostatnie dwa znaki:", text[-2:])
```

Wynik programu: Pierwsze trzy znaki: Stu | Ostatnie dwa znaki: kę

Wnioski: Operator wycinania (slicing) jest bardzo użyteczny przy pracy z ciągami znaków.

Podsumowanie

Podczas wykonywania laboratorium nauczyłem się, jak działa instrukcja warunkowa if-elif-else, jak odczytywać i zapisywać pliki tekstowe oraz jak wykorzystywać metody operujące na ciągach znaków. Ćwiczenia były praktyczne i pozwoliły utrwalić podstawy Pythona.