

Tytuł ćwiczenia: Jednostka 1 - Paradygmat imperatywny

Autor: Joanna Dagil

Grupa: TCH-1

Data: 07.10.2025

1 Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z klasyfikacją programowania na paradygmaty i z założeniami paradygmatu imperatywnego. A następnie zapoznanie się z kolejnymi elementami programowania imperatywnego:

1. zmiennymi i typami danych,
2. instrukcjami przypisania i sterującymi,
3. instrukcjami wejścia/wyjścia,
4. procedurami i funkcjami,
5. sekwenencyjnością tego programowania.

2 Przebieg ćwiczenia

W tym ćwiczeniu posłużę się językiem Python w edytorze Visual Studio Code. W celu wykonania zadań zapoznaję się z tutorialami, a następnie przechodzę do stworzenia plików:

zad1.py

```
1 # zdefiniowanie funkcji main
2 def main():
3     # pobranie liczby N przez instrukcję wejścia i przypisanie jej do zmiennej N
4     N_str = input('Podaj liczbę całkowitą N: ')
5     # sprawdzenie czy N jest liczba całkowita przez instrukcję sterującą warunkową if
6     if N_str.lstrip('-').isdigit():
7         N = int(N_str)
8     else:
9         print("Błąd: N musi być liczbą całkowitą!")
10        return
11    # zainicjowanie zmiennej wynikowej suma
12    suma = 0
13    # iteracja po liczbach parzystych od 0 do N włącznie instrukcją sterującą pętlą
14    for i in range(0, N + 1, 2):
15        # dodanie kolejnej liczby parzystej do sumy
16        suma += i
17    # wypisanie wyniku instrukcją wyjścia
18    print(f"Suma liczb parzystych od 0 do {N} wynosi {suma}")
19    pass
20
21 # wywołanie funkcji main
22 if __name__ == '__main__':
23     main()
```

zad2.py

```
1 # zdefiniowanie funkcji main
2 def main():
3     # pobranie liczby N przez instrukcję wejścia i przypisanie jej do zmiennej N
4     N = int(input('Podaj liczbę N: '))
5     # iteracja instrukcją sterującą pętlą while dopóki N jest większe od 0
6     while N > 0:
7         # wypisanie aktualnej wartości N instrukcją wyjścia
8         print(N, end=' ')
9         # zmniejszenie wartości N o 1
10        N -= 1
```

```

11     # wypisanie pustej linii instrukcją wyjścia (w celach estetycznych)
12     print('')
13     pass
14
15 # wywołanie funkcji main
16 if __name__ == '__main__':
17     main()

```

zad3.py

```

1 # zdefiniowanie funkcji main
2 def main():
3     # pobranie ilości liczb przez instrukcję wejścia i przypisanie jej do zmiennej N
4     N = int(input('Podaj ilość liczb do sortowania: '))
5     # zainicjowanie pustej listy numbers
6     numbers = []
7     # iteracja instrukcją sterującą pętlą for i wczytanie N liczb do listy numbers
8     # przez instrukcję wejścia
9     for i in range(N):
10         number = float(input(f'Podaj {i+1} liczbę: '))
11         numbers.append(number)
12     # iteracja instrukcją sterującą pętlą for
13     for i in range(N-1):
14         # iteracja zagnieżdzoną instrukcją sterującą pętlą for
15         for j in range(N-1-i):
16             # porównanie dwóch sąsiednich elementów listy przy pomocy instrukcji
17             warunkowej if
18             if numbers[j] > numbers[j+1]:
19                 # zamiana miejscami dwóch sąsiednich elementów listy za pomocą
20                 # przypisania wielokrotnego
21                 numbers[j], numbers[j+1] = numbers[j+1], numbers[j]
22     # wypisanie posortowanej listy numbers instrukcją wyjścia
23     print(numbers)
24     pass
25
26 # wywołanie funkcji main
27 if __name__ == '__main__':
28     main()

```

Pliki uruchamiam polecienniem

```
1 python zad{N}.py
```

3 Wyniki działania

Dla Zadania 1:

```

1 PS C:\Users\asiad\Desktop\paradygmaty> python zad1.py
2 Podaj liczbę całkowitą N: 3
3 Suma liczb parzystych od 0 do 3 wynosi 2
4 PS C:\Users\asiad\Desktop\paradygmaty> python zad1.py
5 Podaj liczbę całkowitą N: 5000
6 Suma liczb parzystych od 0 do 5000 wynosi 6252500
7 PS C:\Users\asiad\Desktop\paradygmaty> python zad1.py
8 Podaj liczbę całkowitą N: -20
9 Suma liczb parzystych od 0 do -20 wynosi 0
10 PS C:\Users\asiad\Desktop\paradygmaty> python zad1.py
11 Podaj liczbę całkowitą N: 0
12 Suma liczb parzystych od 0 do 0 wynosi 0
13 PS C:\Users\asiad\Desktop\paradygmaty> python zad1.py
14 Podaj liczbę całkowitą N: 1
15 Suma liczb parzystych od 0 do 1 wynosi 0
16 PS C:\Users\asiad\Desktop\paradygmaty> python zad1.py
17 Podaj liczbę całkowitą N: 1.5
18 Błąd: N musi być liczbą całkowitą!

```

Dla Zadania 2:

```
1 PS C:\Users\asiad\Desktop\paradygmaty> python zad2.py
2 Podaj liczbę N: 16
```

```
3 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
4 PS C:\Users\asiad\Desktop\paradygmaty> python zad2.py
5 Podaj liczbę N: -30
6
7 PS C:\Users\asiad\Desktop\paradygmaty> python zad2.py
8 Podaj liczbę N: 0
9
10 PS C:\Users\asiad\Desktop\paradygmaty> python zad2.py
11 Podaj liczbę N: 1
12 1
```

Dla Zadania 3:

```
1 PS C:\Users\asiad\Desktop\paradygmaty> python zad3.py
2 Podaj ilość liczb do sortowania: 3
3 Podaj 1 liczbę: 10
4 Podaj 2 liczbę: 3
5 Podaj 3 liczbę: 11
6 [3.0, 10.0, 11.0]
7 PS C:\Users\asiad\Desktop\paradygmaty> python zad3.py
8 Podaj ilość liczb do sortowania: 1
9 Podaj 1 liczbę: 2
10 [2.0]
11 PS C:\Users\asiad\Desktop\paradygmaty> python zad3.py
12 Podaj ilość liczb do sortowania: 4
13 Podaj 1 liczbę: 1.2
14 Podaj 2 liczbę: 1
15 Podaj 3 liczbę: 3.5
16 Podaj 4 liczbę: 1.2
17 [1.0, 1.2, 1.2, 3.5]
```

4 Wnioski

Paradygmat imperatywny wykonuje polecenia sekwencyjnie - po kolej, z wyjątkiem instrukcji sterujących - warunkowych lub pętli. Wartości zmiennych w trakcie trwania programu mogą ulegać zmianie - nazywane jest to "zmianą stanu programu".

W mojej ocenie jest to najbardziej podstawowy paradygmat, przeznaczony do małych, prostych projektów.