|  |  |  |  |
| --- | --- | --- | --- |
|  | Politechnika Bydgoska im. J. J. Śniadeckich  Wydział Telekomunikacji,  Informatyki i Elektrotechniki  **Zakład Systemów Teleinformatycznych** | |  |
| **Przedmiot** | Skryptowe języki programowania | | |
| **Prowadzący** | mgr inż. Martyna Tarczewska | | |
| **Temat** | Numpy | | |
| **Student** | Paweł Jońca | | |
| **Nr lab.** | 9 | **Data wykonania** | 14.12.2024 |
| **Ocena** |  | **Data oddania spr.** | 14.12.2024 |

Zad 1

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Obraz zawierający tekst, Czcionka, zrzut ekranu, design

Opis wygenerowany automatycznie

*import* numpy *as* np  
  
*def* replace\_zeros(A, x):  
 A = np.array(A)  
 A[A == 0] = x  
 *return* A  
  
*# Przykład użycia*mat = np.array([[0, 2, 0], [4, 0, 6], [0, 8, 9]])  
x\_val = 3  
result = replace\_zeros(mat, x\_val)  
print("Macierz po zamianie zer na", x\_val, ":")  
print(result)

Zad 2

Obraz zawierający tekst, elektronika, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

*def* medianize(A):  
 *if not* A: *# Sprawdzenie, czy lista nie jest pusta  
 return* []  
 *# Obliczenie średniej wartości tablicy* average = sum(A) / len(A)  
 *# Odjęcie średniej od każdego elementu tablicy* result = [x - average *for* x *in* A]  
 *return* result  
*# Przykład użycia*A = [1, 2, 3, 4, 5]  
print(medianize(A))

Zad 3

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

*import* numpy *as* np  
  
*def* medianize(A):  
 *if not* A: *# Sprawdzenie, czy lista nie jest pusta  
 return* []  
 *# Obliczenie średniej wartości tablicy* average = sum(A) / len(A)  
 *# Odjęcie średniej od każdego elementu tablicy* result = [x - average *for* x *in* A]  
 *return* result  
  
*def* matrix():  
 *# Generowanie macierzy 5x5 z liczbami naturalnymi mniejszymi od 100* matrix = np.random.randint(0, 100, (5, 5))  
 print("Macierz:")  
 print(matrix)  
 *# Największy element globalnie* max\_global = np.max(matrix)  
 print("Największy element globalnie:", max\_global)  
 *# Największy element w każdym wierszu* max\_rows = np.max(matrix, axis=1)  
 print("Największy element w każdym wierszu:", max\_rows)  
 *# Największy element w każdej kolumnie* max\_cols = np.max(matrix, axis=0)  
 print("Największy element w każdej kolumnie:", max\_cols)  
*# Przykład użycia*A = [1, 2, 3, 4, 5]  
print(medianize(A))  
matrix()

Zad 4

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, System operacyjny

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie

*import* numpy *as* np  
  
*# Przygotowanie przykładowej tablicy*array = np.arange(12) *# Tablica z liczbami od 0 do 11  
  
# Test 1: -1 jako pierwszy parametr*reshaped1 = array.reshape(-1, 3)  
print("Test 1: -1 jako pierwszy parametr")  
print(reshaped1)  
  
*# Test 2: -1 jako drugi parametr*reshaped2 = array.reshape(4, -1)  
print("\nTest 2: -1 jako drugi parametr")  
print(reshaped2)

Zad 5

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

*import* pandas *as* pd  
*# Wczytanie pliku CSV z separatorem tabulatora*df = pd.read\_csv("oceny.csv", sep='\t')  
*# Sprawdzenie nazw kolumn i pierwszych wierszy*print("Nazwy kolumn w DataFrame:", df.columns)  
print(df.head())  
*# Najniższa ocena z laboratoriów dla każdego studenta*df['Min\_LAB'] = df[['Lab1', 'Lab2', 'Lab3', 'Lab4', 'Lab5']].min(axis=1)  
print("Najniższa ocena z laboratoriów dla każdego studenta:")  
print(df[['Min\_LAB']])  
*# Średnia ocena z egzaminu*average\_exam = df['Exam'].mean()  
print(f"Średnia ocena z egzaminu: {average\_exam:.2f}")  
*# Liczba 2 z egzaminu*count\_exam\_2 = (df['Exam'] == 2).sum()  
print(f"Liczba ocen 2 z egzaminu: {count\_exam\_2}")  
*# Czy jest student, który miał same 5 z laboratoriów*all\_5\_in\_labs = any((df[['Lab1', 'Lab2', 'Lab3', 'Lab4', 'Lab5']] == 5).all(axis=1))  
print(f"Czy jest student, który miał same 5 z laboratoriów: {'Tak' *if* all\_5\_in\_labs *else* 'Nie'}")  
*# Czy jest student, który miał 2 z Lab2 i Lab3*student\_2\_in\_lab2\_lab3 = any((df['Lab2'] == 2) & (df['Lab3'] == 2))  
print(f"Czy jest student, który miał 2 z Lab2 i Lab3: {'Tak' *if* student\_2\_in\_lab2\_lab3 *else* 'Nie'}")  
*# Liczba studentów, którzy dostali wyższą ocenę z egzaminu niż średnia z laboratoriów*df['Average\_LAB'] = df[['Lab1', 'Lab2', 'Lab3', 'Lab4', 'Lab5']].mean(axis=1)  
students\_higher\_exam = (df['Exam'] > df['Average\_LAB']).sum()  
print(f"Liczba studentów, którzy dostali wyższą ocenę z egzaminu niż średnia z laboratoriów: {students\_higher\_exam}")  
*# Liczba piątek uzyskanych przez studenta z największą liczbą 5*df['Count\_5'] = (df[['Lab1', 'Lab2', 'Lab3', 'Lab4', 'Lab5', 'Exam']] == 5).sum(axis=1)  
max\_5\_student = df['Count\_5'].max()  
print(f"Liczba piątek uzyskanych przez studenta z największą liczbą 5: {max\_5\_student}")

Zad 6

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

*import* numpy *as* np  
*# Wylosowanie 10 liczb do tablicy Numpy*array = np.random.rand(10) *# Liczby losowe z zakresu [0, 1)*print("Oryginalna tablica:", array)  
*# Sortowanie rosnąco*sorted\_array\_ascending = np.sort(array)  
print("Tablica posortowana rosnąco:", sorted\_array\_ascending)  
*# Sortowanie malejąco*sorted\_array\_descending = np.sort(array)[::-1] *# Odwrócenie kolejności*print("Tablica posortowana malejąco:", sorted\_array\_descending)

Zad 7

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie

*import* numpy *as* np  
*# Tworzenie przykładowej tablicy 5x5*tablica = np.random.randint(1, 10, (5, 5)) *# Tablica z losowymi wartościami od 1 do 9  
# Definiowanie wag*wagi = np.array([1, 2, 3, 2, 1])  
*# Obliczanie średniej ważonej dla każdego wiersza*srednie\_wazone = np.dot(tablica, wagi) / sum(wagi)  
*# Wyświetlenie wyników*print("Tablica:")  
print(tablica)  
print("\nŚrednie ważone dla każdego wiersza:")  
print(srednie\_wazone)

Zad 8

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie

*import* random  
*from* collections *import* Counter  
*# Tworzenie tablicy 10x10 z losowymi liczbami*tablica = [[random.randint(0, 10) *for* \_ *in* range(10)] *for* \_ *in* range(10)]  
*# Zliczanie wystąpień liczb w tablicy*flat\_tablica = [element *for* row *in* tablica *for* element *in* row] *# Spłaszczamy tablicę*liczba\_wystapien = Counter(flat\_tablica)  
*# Wypisanie tablicy*print("Tablica 10x10:")  
*for* row *in* tablica:  
 print(row)  
*# Wypisanie liczności wystąpień elementów*print("\nLiczność wystąpień elementów:")  
*for* liczba, wystapienia *in* liczba\_wystapien.items():  
 print(f"Liczba {liczba}: {wystapienia} razy")

Zad 9

Przetestować operator \* oraz operator @. Do czego służą?

Między liczbami \* służy do mnożenia dwóch liczb

Do rozpakowywania argumentów w funkcjach

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Rozpakowywanie w przypisaniach: Użycie \* w przypisaniach umożliwia rozpakowanie elementów z listy lub krotki.



Mnożenie elementów w iterowalnych obiektach (np. listach, ciągach):



W Pythonie operator @ jest używany w matematyce do mnożenia macierzy, zwłaszcza w bibliotece numpy.

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznieObraz zawierający Czcionka, zrzut ekranu, tekst, design

Opis wygenerowany automatycznie

Operator @ jest również używany w Pythonie do dekorowania funkcji, tzn. przypisania funkcji dekorującej do innej funkcji.

***Zad 10 Wypróbować funkcję set\_printoptions. Jakie daje możliwości?***

Precision - Ustawia liczbę cyfr po przecinku dla liczb zmiennoprzecinkowych.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Suppress -Kontroluje, czy bardzo małe liczby (w notacji wykładniczej) mają być pokazywane w zwykłym formacie.

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Threshold - Określa liczbę elementów tablicy, które mają być wyświetlane. Gdy liczba elementów przekracza próg, tablica jest skracana.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Linewidth – określa maksymalną szerokość linii wyświetlania tablicy

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Pozwala zdefiniować własne funkcje formatowania dla różnych typów danych.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Wnioski: W trakcie realizacji zadań z modułu Numpy zyskałem praktyczne umiejętności w zakresie manipulacji danymi oraz wykonywania obliczeń numerycznych na macierzach. Użycie funkcji takich jak reshape, medianize oraz operacje na tablicach wielowymiarowych pozwoliło mi lepiej zrozumieć, jak efektywnie przetwarzać duże zbiory danych w Pythonie. Dodatkowo, nauczyłem się korzystać z wczytywania danych z plików oraz ich analizy, co jest niezbędne w praktycznych zastosowaniach programowania.