
	Politechnika Bydgoska im. J. J. Śniadeckich  <b>Wydział Telekomunikacji, Informatyki i Elektrotechniki</b>		
<b>Przedmiot</b>	Skryptowe języki programowania		
<b>Prowadzący</b>	mgr inż. Martyna Tarczewska		
<b>Temat</b>	Matplotlib		
<b>Student</b>	Marcin Ogórkiewicz		
<b>Nr ćw.</b>	10	<b>Data wykonania</b>	06.12.2023
<b>Ocena</b>		<b>Data oddania spr.</b>	06.12.2023

### Zadanie 1.

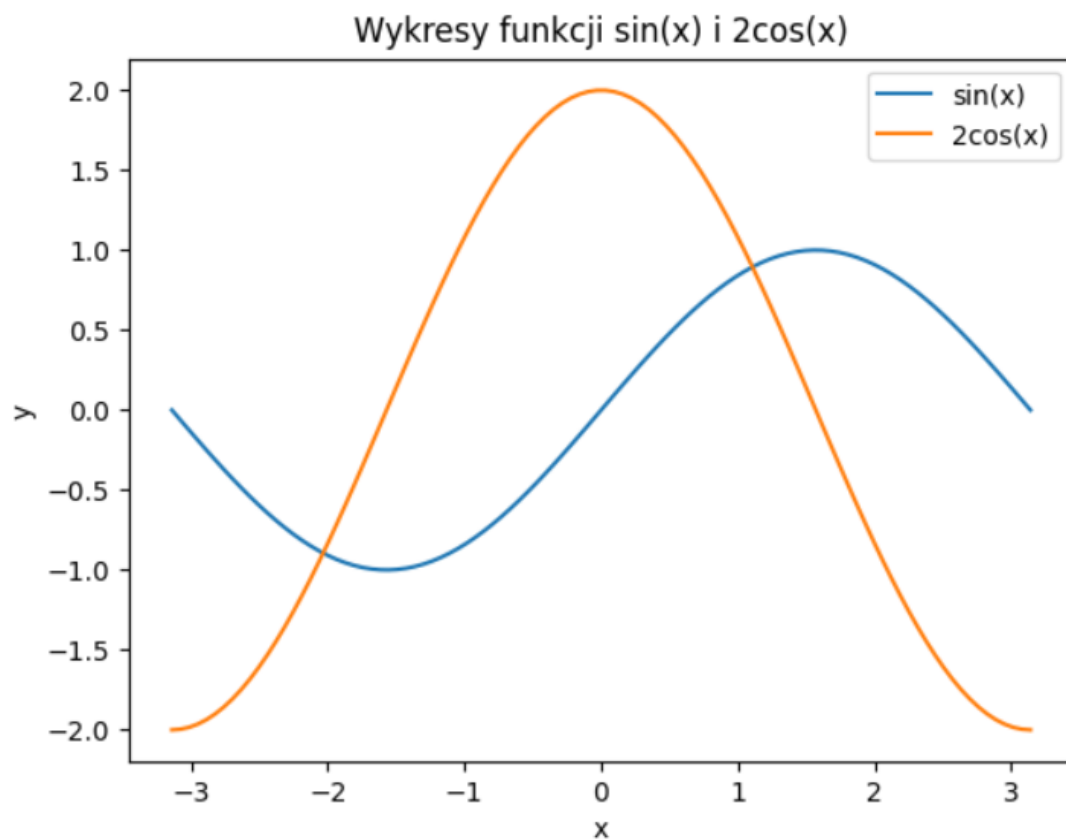
```
#!C:\Users\koksu\AppData\Local\Programs\Python\Python310

# importy
import typing
import matplotlib.pyplot as plt
import numpy as np
# stałe i zmienne globalne

# funkcje

def main() -> None:
    x = np.linspace(-np.pi, np.pi, 100)
    y_sin = np.sin(x)
    y_cos = 2 * np.cos(x)
    plt.plot(x, y_sin, label='sin(x)')
    plt.plot(x, y_cos, label='2cos(x)')
    plt.title('Wykresy funkcji sin(x) i 2cos(x)')
    plt.xlabel('x')
    plt.ylabel('y')
    plt.legend()
    plt.show()

if __name__ == "__main__":
    main()
```



## Zadanie 2

```
#!C:\Users\koksu\AppData\Local\Programs\Python\Python310

# importy
import typing
import matplotlib.pyplot as plt
# stałe i zmienne globalne

# funkcje

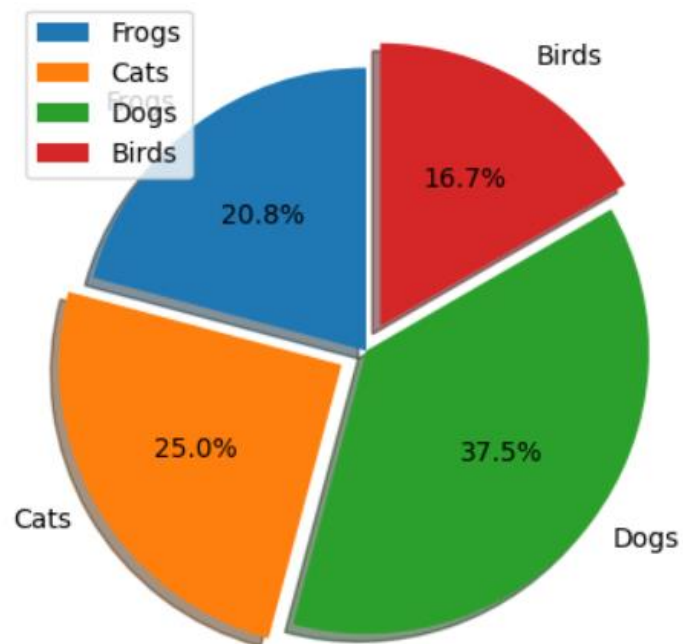
def main() -> None:
    labels = 'Frogs', 'Cats', 'Dogs', 'Birds'
    sizes = [25, 30, 45, 20]
    explode = (0, 0.1, 0, 0.1) # Dodane przesunięcia dla sektorów
    shadow = True # Dodany cień
    start_angle = 90 # Zmieniony kąt początkowy
    fig1, ax1 = plt.subplots()
    ax1.pie(
        sizes,
        explode=explode,
        labels=labels,
        autopct='%1.1f%%',
        shadow=shadow,
        startangle=start_angle
    )
```

```

ax1.legend(labels, loc="best")
plt.show()

if __name__ == "__main__":
    main()

```



### Zadanie 3.

Podkreślenie pomija dodanie nowej etykiety w spisie.

```

#!C:\Users\koksu\AppData\Local\Programs\Python\Python310

# importy
import typing
import matplotlib.pyplot as plt
# stałe i zmienne globalne

# funkcje

def main() -> None:

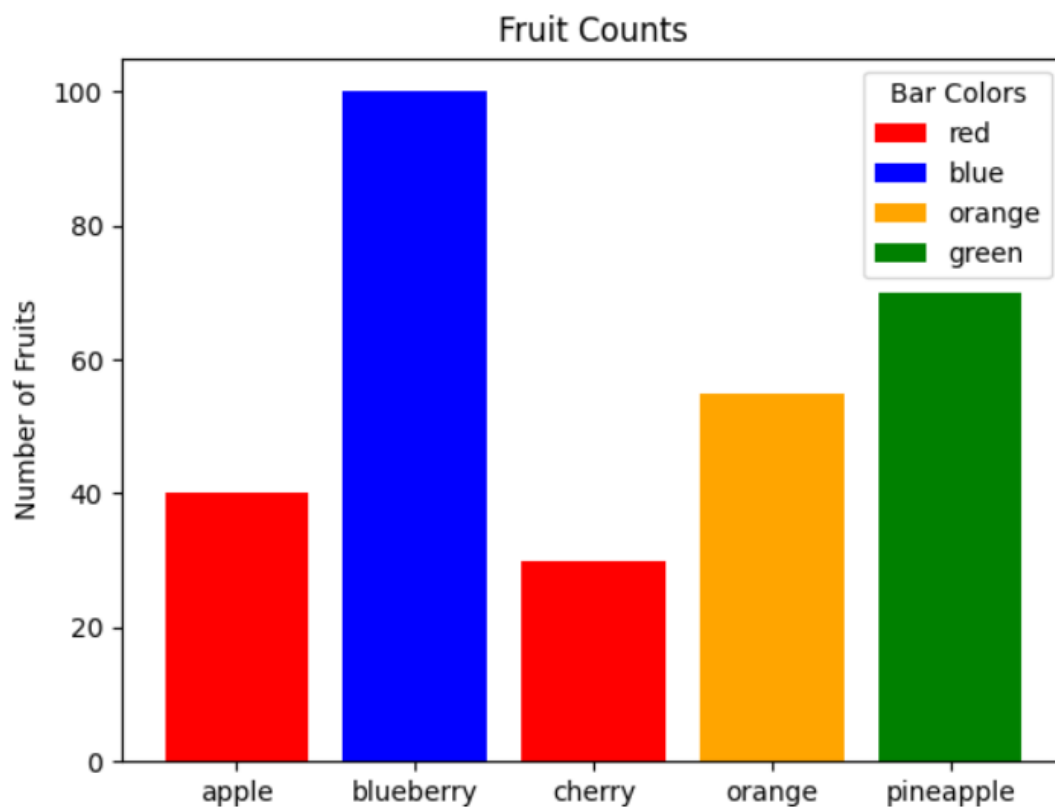
```

```

fig, ax = plt.subplots()
fruits = ['apple', 'blueberry', 'cherry', 'orange', 'pineapple']
counts = [40, 100, 30, 55, 70]
bar_labels = ['red', 'blue', 'red', 'orange', 'green']
bar_colors = ['red', 'blue', 'red', 'orange', 'green']
ax.bar(fruits, counts, label=bar_labels, color=bar_colors)
ax.set_ylabel('Number of Fruits')
ax.set_title('Fruit Counts')
ax.legend(title='Bar Colors')
plt.show()

if __name__ == "__main__":
    main()

```



#### Zadanie 4.

```

#!C:\Users\koksu\AppData\Local\Programs\Python\Python310

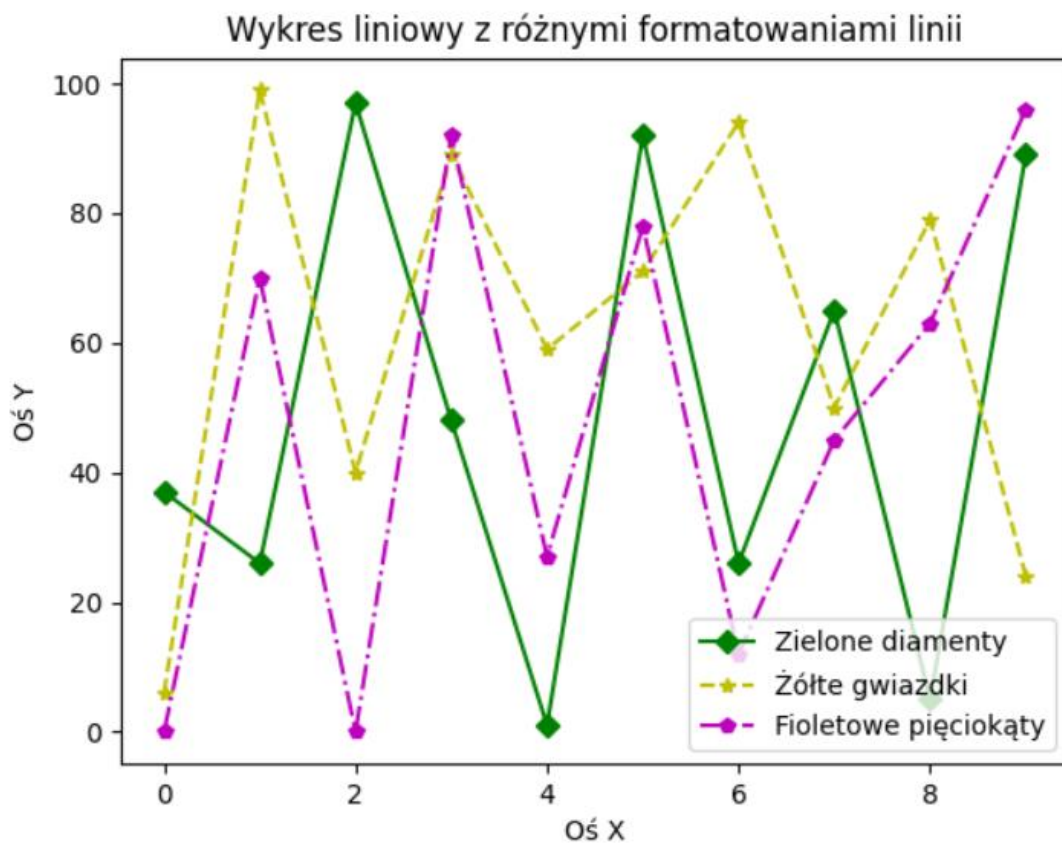
# importy
import typing
import matplotlib.pyplot as plt
import numpy as np
# stałe i zmienne globalne

```

```
# funkcje

def main() -> None:
    x = np.linspace(0, 9, 10)
    y1 = np.random.randint(0, 100, size=10)
    y2 = np.random.randint(0, 100, size=10)
    y3 = np.random.randint(0, 100, size=10)
    plt.plot(x, y1, 'gD-', label='Zielone diamenty')
    plt.plot(x, y2, 'y*--', label='Żółte gwiazdki')
    plt.plot(x, y3, 'mp-.', label='Fioletowe pięciokąty')
    plt.xlabel('Oś X')
    plt.ylabel('Oś Y')
    plt.title('Wykres liniowy z różnymi formatowaniami linii')
    plt.legend()
    plt.show()

if __name__ == "__main__":
    main()
```



**Zadanie 5.**

```

#!C:\Users\koksu\AppData\Local\Programs\Python\Python310

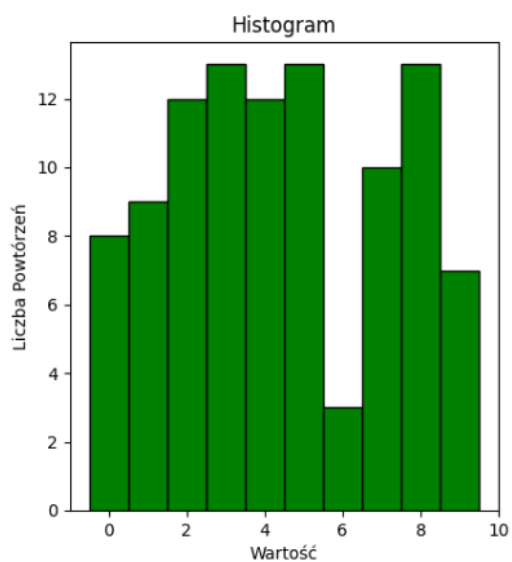
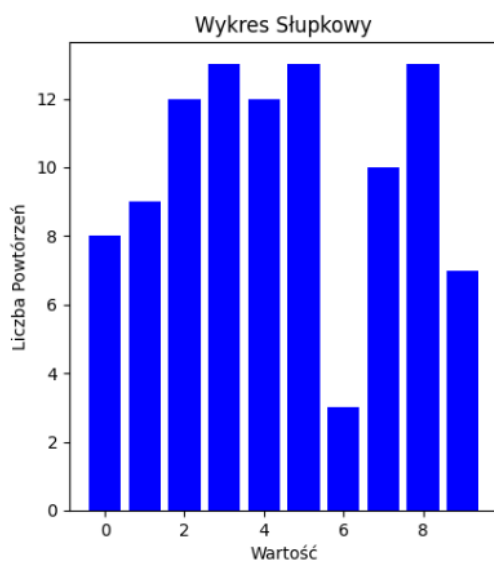
# importy
import typing
import matplotlib.pyplot as plt
import numpy as np
# stałe i zmienne globalne

# funkcje

def main() -> None:
    data = np.random.randint(0, 10, 100)
    unique_values, counts = np.unique(data, return_counts=True)
    plt.figure(figsize=(10, 5))
    plt.subplot(1, 2, 1)
    plt.bar(unique_values, counts, color='blue')
    plt.title('Wykres Słupkowy')
    plt.xlabel('Wartość')
    plt.ylabel('Liczba Powtórzeń')
    plt.subplot(1, 2, 2)
    plt.hist(data, bins=np.arange(11) - 0.5, color='green',
edgecolor='black')
    plt.title('Histogram')
    plt.xlabel('Wartość')
    plt.ylabel('Liczba Powtórzeń')
    plt.show()

if __name__ == "__main__":
    main()

```



**Zadanie 6.**

Parametr loc wykorzystuje się do ustalenia lokalizacji legendy na wygenerowanym rysunku. Wartości słowne wybierają jedną ze zdefiniowanych pozycji dla legendy, a cyfry pozwalają na wprowadzenie dokładnych współrzędnych.

### Zadanie 7.

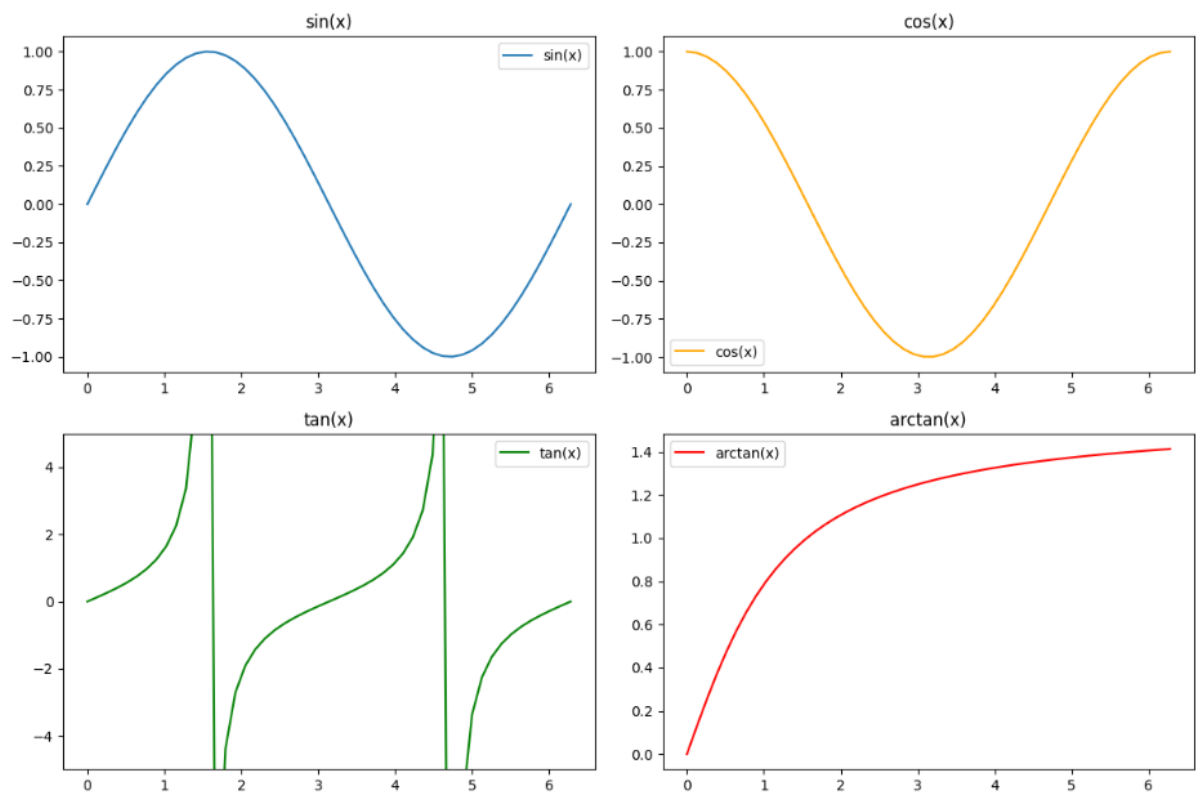
```
#!C:\Users\koku\AppData\Local\Programs\Python\Python310

# importy
import typing
import matplotlib.pyplot as plt
import numpy as np
# stałe i zmienne globalne

# funkcje

def main() -> None:
    x = np.linspace(0, 2 * np.pi)
    y1 = np.sin(x)
    y2 = np.cos(x)
    y3 = np.tan(x)
    y4 = np.arctan(x)
    plt.figure(figsize=(12, 8))
    plt.subplot(2, 2, 1)
    plt.plot(x, y1, label='sin(x)')
    plt.title('sin(x)')
    plt.legend()
    plt.subplot(2, 2, 2)
    plt.plot(x, y2, label='cos(x)', color='orange')
    plt.title('cos(x)')
    plt.legend()
    plt.subplot(2, 2, 3)
    plt.plot(x, y3, label='tan(x)', color='green')
    plt.title('tan(x)')
    plt.ylim(-5, 5) # Ustawienie ograniczenia osi Y dla lepszej
widoczności
    plt.legend()
    plt.subplot(2, 2, 4)
    plt.plot(x, y4, label='arctan(x)', color='red')
    plt.title('arctan(x)')
    plt.legend()
    plt.tight_layout()
    plt.show()

if __name__ == "__main__":
    main()
```



## Zadanie 8.

```
#!C:\Users\koksu\AppData\Local\Programs\Python\Python310

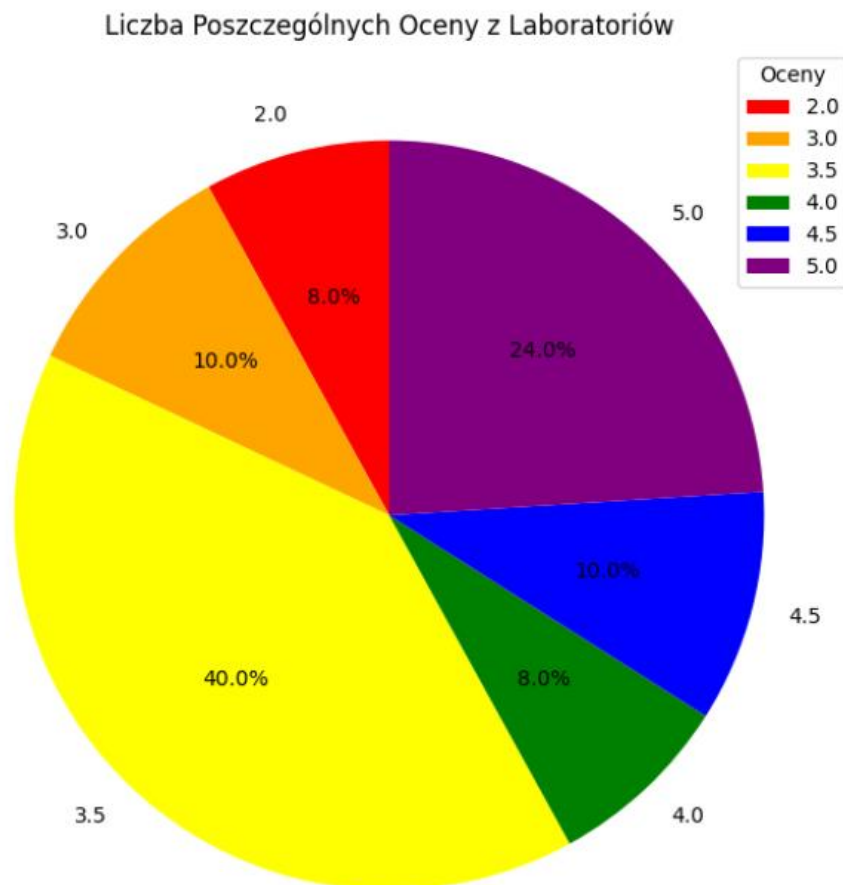
# importy
import typing
import matplotlib.pyplot as plt
import numpy as np
# stałe i zmienne globalne

# funkcje

def main() -> None:
    data = np.genfromtxt('../Lab09/oceny.csv', delimiter='\t',
skip_header=1)
    lab_grades = data[:, :-1]
    flat_lab_grades = lab_grades.flatten()
    grades, grades_counts = np.unique(flat_lab_grades, return_counts=True)
    plt.figure(figsize=(8, 8))
    plt.pie(grades_counts, labels=grades, autopct='%1.1f%%', startangle=90,
        colors=['red', 'orange', 'yellow', 'green', 'blue', 'purple'])
    plt.title('Liczba Poszczególnych Oceny z Laboratoriów')
    plt.legend(grades, title='Oceny', loc="best")
    plt.savefig('Ilość ocen.png')
    plt.show()

if __name__ == "__main__":
    main()
```





## Wnioski

Ćwiczenie nauczyło mnie posługiwania się modułem Matplotlib.