|  |  |  |  |
| --- | --- | --- | --- |
| Obraz zawierający symbol, design  Opis wygenerowany automatycznie | Politechnika Bydgoska im. J. J. Śniadeckich  **Wydział Telekomunikacji,**  **Informatyki i Elektrotechniki** | | Obraz zawierający symbol, logo, Grafika, Czcionka  Opis wygenerowany automatycznie |
| **Przedmiot** | Skryptowe języki programowania | | |
| **Prowadzący** | mgr inż. Martyna Tarczewska | | |
| **Temat** | Więcej liczb | | |
| **Student** | Ivan Fomin | | |
| **Nr ćw.** | 5 | **Data wykonania** | 11.11.2023 |
| **Ocena** |  | **Data oddania spr.** | 11.11.2023 |

1. **Cel ćwiczenia:**

Celem ćwiczenia jest dalsze poznawanie sposobów zapisu i przetwarzania liczb, jakie oferuje język programowania Python oraz wykonanie zadań utrwalających przyswojoną wiedzę.

1. **Przygotowanie do ćwiczeń:**

Zapoznałem się z dokumentacją pliku LAB05.pdf i z modułami math oraz random.

1. **Zadania/przebieg ćwiczenia:**

Zadanie 1:

def get\_base()->int:

    while True:

        base = int(input('Podaj podstawę systemu liczbowego (2, 8, 10, 16): '))

        if base in [2, 8, 10, 16]:

            return base

        else:

            print('Podano nieprawidłową wartość!')

def convert\_and\_print(base)->None:

    num = int(input('Podaj liczbę: '), base)

    print('Dwójkowy:', bin(num))

    print('Ósemkowy:', oct(num))

    print('Dziesiętny:', num)

    print('Szesnastkowy:', hex(num))

convert\_and\_print(get\_base())

Wyniki:  
Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Zadanie 2:

def get\_bit\_value(number:int, bit\_index:int)->int:

    if number < 0 or number > 255:

        raise ValueError("Number must be in range <0, 255>.")

    if bit\_index < 0 or bit\_index > 7:

        raise ValueError("Bit index must be in range <0, 7>.")

    return (number >> bit\_index) & 1

number = int(input("Give a number (0-255): "))

bit\_index = int(input("Give indeks of bit (0-7): "))

bit\_value = get\_bit\_value(number, bit\_index)

print(f"Value of bit on position {bit\_index} in number {number} is {bit\_value}.")

sprawdzanie ostatniego bitu po przesunięciu za pomocą AND z 1

Wyniki:  
Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie

Zadanie 3:

import math

import random

def rand3()->tuple:

    a = random.randint(3, 10)

    b = random.randint(3, 10)

    c = random.randint(3, 10)

    return a, b, c

a,b,c = rand3()

print (a,b,c)

def chek(a,b,c)->bool:

    if a+b>c and a+c>b and b+c>a:

        return True

    else:

        return False

def calculate\_area(a, b, c)->float:

    s = (a + b + c) / 2

    return math.sqrt(s \* (s - a) \* (s - b) \* (s - c))

if(chek(a,b,c)):

    print("Yes")

    print(f'Area: {format(calculate\_area(a,b,c), ".2f")}')

else:

    print("No")

trójkąt istnieje tylko wtedy , gdy suma dwóch jego stron jest większa za trzecią.

Wyniki:  
Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie

Obraz zawierający tekst, Czcionka, zrzut ekranu, Grafika

Opis wygenerowany automatycznie

Zadanie 4:

import random

coin = ['o', 'r']

def coinflip\_result()->str:

    return random.choice(coin)

def user\_choice()->str:

    while True:

        user\_input = input("Wybierz orzeł('o') lub reszka('r') lub 'X' żeby skończyć grę: ")

        if user\_input in coin or user\_input == 'X':

            return user\_input

        else:

            print("Nieprawidłowy wybór! Spróbuj ponownie.")

def game()->None:

    game\_counter = 0

    win\_counter = 0

    while True:

        game\_counter += 1

        user\_coin = user\_choice()

        if user\_coin == 'X':

            print('GAME OVER')

            break

        if coinflip\_result() == user\_coin:

            win\_counter += 1

            print(f'Wygrałeś! Wygrałeś {win\_counter} razy na {game\_counter} gry')

        else:

            print(f'Przegrałeś! Wygrałeś {win\_counter} razy na {game\_counter} gry')

if \_\_name\_\_ == "\_\_main\_\_":

    game()

zamiast pełnych nazw zrobiłem po prostu pierwsze litery słów dla identyfikacji wyboru

Wyniki:  
Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie  
Zadanie 5:

import random

tools = ['k', 'p', 'n']

def user\_choice()->str:

    while True:

        user\_input = input("Wybierz kamień('k') / papier('p') / nożyce('n') lub 'X' żeby skończyć grę: ")

        if user\_input in tools or user\_input == 'X':

            return user\_input

        else:

            print("Nieprawidłowy wybór! Spróbuj ponownie.")

def computer\_choice()->str:

    return random.choice(tools)

def game()->None:

    game\_counter = 0

    win\_counter = 0

    draw\_counter = 0

    while True:

        user\_tool = user\_choice()

        if user\_tool == 'X':

            print(f'GAME OVER\n Wyniki: Wygrałeś {win\_counter} razy i zegrałeś w remis {draw\_counter} razy na {game\_counter} gry')

            break

        pc\_tool = computer\_choice()

        game\_counter += 1

        if pc\_tool == 'k' and user\_tool == 'p' or pc\_tool == 'p' and user\_tool == 'n' or pc\_tool == 'n' and user\_tool == 'k':

            win\_counter += 1

            print(f'Wygrałeś! {user\_tool} pokonuje {pc\_tool}')

        elif pc\_tool == user\_tool:

            draw\_counter += 1

            print(f'Remis! {user\_choice} = {pc\_tool}')

        else:

            print(f'Przegrałeś! {pc\_tool} pokonuje {user\_tool}')

if \_\_name\_\_ == "\_\_main\_\_":

    game()

zamiast pełnych nazw zrobiłem po prostu pierwsze litery słów dla identyfikacji wyboru

zrobiłem jeszcze licznik remisów

Wyniki:  
Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Zadanie 6:

import math

def ladder\_height(length: float, angle: float) -> float:

    height = length \* math.sin(math.radians(angle))

    return round(height, 8)

l = float(input("Podaj długość drabiny: "))

a = float(input("Podaj kąt (w stopniach): "))

print(f"Wysokość wynosi: {ladder\_height(l, a)}")

po prostu szukam wysokość przez sinus kąta i długość

Wyniki:  
Obraz zawierający tekst, Czcionka, typografia

Opis wygenerowany automatycznie

Zadanie 7:

import math as m

def check\_one():

    for angle in range(91):

        a\_r = m.radians(angle)

        result = round(m.sin(a\_r)\*\*2 + m.cos(a\_r)\*\*2, 5)

        print(f"Kąt: {angle} stopni, wynik: {result}")

        if result != 1:

            print(f"Jedynka trygonometryczna nie sprawdza się dla kąta {angle} stopni. Wynik: {result}")

check\_one()

Wyniki:

Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie Obraz zawierający tekst, Czcionka, zrzut ekranu, typografia

Opis wygenerowany automatycznie

Zadanie 8:

def xor\_letters(text: str, key: str) -> str:

    coded\_text = ""

    for i in range(len(text)):

        xor\_result = ord(text[i]) ^ ord(key[i])

        coded\_text += chr(xor\_result)

    return coded\_text

text = "algorytm"

key = "kodykody"

coded\_text = xor\_letters(text, key)

print(f"Niejawny tekst: {coded\_text}")

decoded\_text = xor\_letters(coded\_text, key)

print(f"Jawny tekst po dekodowaniu: {decoded\_text}")

Obraz zawierający tekst, pismo odręczne, szkic, rysowanie

Opis wygenerowany automatyczniekoduję i dekoduję każdą literę słowa za pomocą XOR:

Wyniki:  
Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

Zadanie 9:

def power\_of\_two(p: int):

    if p > 0:

        return 1 << p

    elif p == 0:

        return 1

    else:

        print("Wykładnik potęgi nie może być ujemny")

if \_\_name\_\_ == "\_\_main\_\_":

    try:

        p = int(input("Podaj wykładnik potęgi: "))

        print(power\_of\_two(p))

    except ValueError:

        print("Podana wartość nie jest liczbą całkowitą!")

Wyniki:

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

Zadanie 10:

import math

import platform

import sys

a = float(input("give the number a: "))

b = int(input("give the number b: "))

def testowanief(a,b):

    print(math.trunc(a))

    print(math.floor(a))

    print(math.ceil(a))

    print(sys.version\_info)

    if sys.version\_info > (3, 9):

        print(math.lcm(int(a),b)) *#najmniejsza wspólna wielokrotność*

        print(math.gcd(int(a),b)) *#największy wspólny dzielnik*

*# Obliczanie wartości bezwzględnej*

    print(f'Wartość bezwzględna: {math.fabs(a)}')

if \_\_name\_\_ == "\_\_main\_\_":

    testowanief(a,b)

trunk różni się od floor tym że działa zależnie od 0 gdy liczba jest > 0 to działa jak floor , a gdy < 0 jak ceil

math.fabs() jest do obliczania wartości bezwzględnej

Wyniki:  
Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Zadanie 11:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznieTesty:

**4. Wnioski:**W tym labie zapoznałem się możliwościami modułów math i random, wykorzystałem operację bitowe dla wykonywania działań.