

Zadanie 1

CEL ZADANIA

Celem zadania było zapoznanie się z podstawowymi operatorami i typami danych w języku Python oraz sprawdzenie ich działania przy użyciu funkcji type().

PRZEBIEG ZADANIA

W interpreterze Pythona wykonano działania arytmetyczne, takie jak dodawanie, dzielenie, potęgowanie i reszta z dzielenia. Następnie zastosowano funkcje konwersji typów: int(), float(), str() i bool() w celu sprawdzenia ich efektu.

KOD PROGRAMU

a)

```
Zad1.py X
D: > Pulpit > Lab.Pdst.Prog > Zad1.py
1 print("1. ",type(1 + 2)) # "+" - Operator dodawania
2 print("2. ",type(1 + 4.5)) # "+" - Operator dodawania
3 print("3. ",type(3/2)) # "/" - Operator dzielenia (z resztą)
4 print("4. ",type(4/2)) # "/" - Operator dzielenia (z resztą)
5 print("5. ",type(3//2)) # "/" - Operator dzielenia całkowitego (bez reszty)
6 print("6. ",type(-3//2)) # "/" - Operator dzielenia całkowitego (bez reszty)
7 print("7. ",type(11 % 2)) # "%" - Operator modulo (wypisanie tylko reszty z dzielenia podanych liczb)
8 print("8. ",type(2 ** 10)) # "**" - Operator potęgowania
9 print("9. ",type(8 ** (1/3))) # "**" - Operator potęgowania, "/" - Operator potęgowania (z resztą), "()" - służy do grupowania wyrażeń i określania kolejności działań
```

b)

```
Zad1.py Zad1_b.py X
D: > Pulpit > Lab.Pdst.Prog > Zad1_b.py
1 print("1. ",int(3.0)) # Funkcja "int()" zmienia liczbę zmiennoprzecinkową (float) na liczbę całkowitą (int)
2 print("2. ",float(3)) # Funkcja "float()" zmienia liczbę całkowitą (int) na liczbę zmiennoprzecinkową (float)
3 print("3. ",float("3.0")) # Funkcja "float()" zamienia napis (string), który ma w sobie liczbę, na liczbę zmiennoprzecinkową (float)
4 print("4. ",str(12.4)) # Funkcja "str()" zamienia liczbę (w tym przypadku zmiennoprzecinkową) na ciąg znaków/tekst (string)
5 print("5. ",bool(0)) # Funkcja "bool()" to logiczny typ danych, przyjmuje tylko dwie wartości (true/false) - wartość 0 to fałsz (false) a każda inna to prawda (true)
```

WYNIK PROGRAMU

a)

```
1. <class 'int'>
2. <class 'float'>
3. <class 'float'>
4. <class 'float'>
5. <class 'int'>
6. <class 'int'>
7. <class 'int'>
8. <class 'int'>
9. <class 'float'>
PS C:\Users\wiki_>
```

b)

```
1. 3
2. 3.0
3. 3.0
4. 12.4
5. False
PS C:\Users\wiki_>
```

ANALIZA WYNIKÓW

Działania zwracały wyniki o odpowiednich typach (int, float). Funkcje konwersji poprawnie zmieniały typy danych.

Przykładowo:

- $1 + 2 \rightarrow$ wynik: 3, typ int
- $1 + 4.5 \rightarrow$ wynik: 5.5, typ float
- `float("3")` \rightarrow konwersja tekstu na liczbę
- `bool(0)` \rightarrow wynik False

WNIOSKI

Python automatycznie rozpoznaje typ danych na podstawie wartości. Funkcje konwersji umożliwiają łatwą zmianę typów. Działanie operatorów jest intuicyjne i spójne z zasadami matematyki.

Zadanie 2

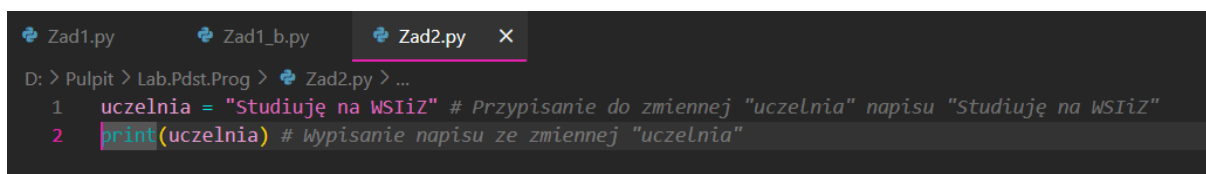
CEL ZADANIA

Celem zadania było zapoznanie się z funkcją `print()` i sposobem wyświetlania tekstu w Pythonie.

PRZEBIEG ZADANIA

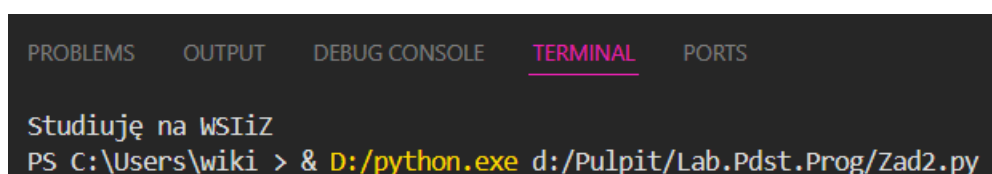
Utworzono zmienną `uczelnia` o wartości "Studiuję na WSIiZ", a następnie za pomocą `print(uczelnia)` wypisano jej zawartość w konsoli.

KOD PROGRAMU



```
Zad1.py  Zad1_b.py  Zad2.py  X
D: > Pulpit > Lab.Pdst.Prog > Zad2.py > ...
1  uczelnia = "Studiuję na WSIiZ" # Przypisanie do zmiennej "uczelnia" napisu "Studiuję na WSIiZ"
2  print(uczelnia) # Wypisanie napisu ze zmiennej "uczelnia"
```

WYNIK PROGRAMU



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Studiuję na WSIiZ
PS C:\Users\wiki > & D:/python.exe d:/Pulpit/Lab.Pdst.Prog/Zad2.py
```

ANALIZA WYNIKÓW

Program poprawnie wyświetlił tekst:

“ Studiuję na WSiIZ”

Nie wystąpiły żadne błędy składniowe ani logiczne.

WNIOSKI

Instrukcja “print()” służy do prezentacji wyników programu i jest podstawową funkcją wyjścia w Pythonie.

Zadanie 3

CEL ZADANIA

Celem zadania było wykorzystanie zmiennych i funkcji print() do tworzenia dynamicznego tekstu.

PRZEBIEG ZADANIA

Zdefiniowano zmienne:

imie = 'Jan'

wiek = 20

wzrost = 178

Następnie wydrukowano zdania przy pomocy print():

Nazywam się Jan i mam 20 lat.

Mój wzrost to 178 cm.

KOD PROGRAMU

```
Zad1.py  Zad1_b.py  Zad2.py  Zad3.py  X
D: > Pulpit > Lab.Pdst.Prog > Zad3.py > ...
1  imie = 'Jan' # Przypisanie do zmiennej "imie" tekst "Jan"
2  wiek = 20 # Przypisanie do zmiennej "wiek" liczbe 20
3  wzrost = 178 # Przypisanie do zmiennej "wzrost" liczby 178
4
5  print("Nazywam się", imie, "i mam", wiek, "lat.\n Mój wzrost to", wzrost, "cm.") # wypisanie tekstu z użyciem powyższych zmiennych oraz wykorzystanie "\n" (nowa linia)
```

WYNIK PROGRAMU

```
Nazywam się Jan i mam 20 lat.
Mój wzrost to 178 cm.
PS C:\Users\wiki_>
```

ANALIZA WYNIKÓW

Program poprawnie połączył tekst ze zmiennymi liczbowymi. Wynik był zgodny z oczekiwaniami.

WNIOSKI

Python umożliwia łatwe łączenie tekstu i wartości zmiennych w jednym komunikacie. Umiejętność formatowania wyjścia jest niezbędna przy tworzeniu przejrzystych programów.

Zadanie 4

CEL ZADANIA

Celem zadania było zastosowanie działań arytmetycznych i formatowania tekstu do obliczenia ceny po rabacie.

PRZEBIEG ZADANIA

Zdefiniowano zmienne:

cena = 39.99

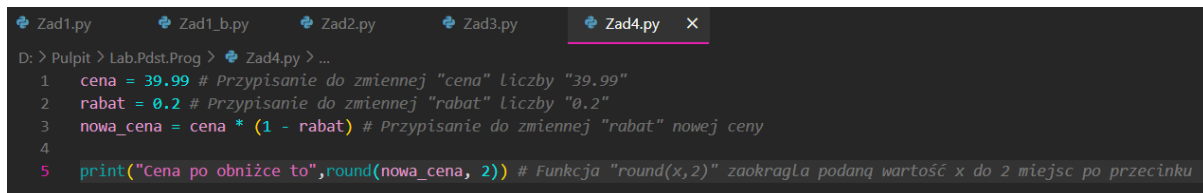
rabat = 0.2

Obliczono cenę po rabacie:

```
nowa_cena = cena * (1 - rabat)

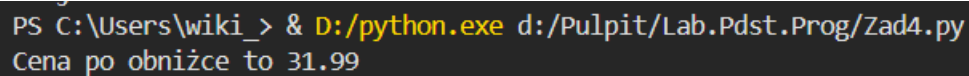
print("Cena po obniżce to", round(nowa_cena, 2))
```

KOD PROGRAMU



```
Zad1.py Zad1_b.py Zad2.py Zad3.py Zad4.py X
D: > Pulpit > Lab.Pdst.Prog > Zad4.py > ...
1  cena = 39.99 # Przypisanie do zmiennej "cena" liczby "39.99"
2  rabat = 0.2 # Przypisanie do zmiennej "rabat" liczby "0.2"
3  nowa_cena = cena * (1 - rabat) # Przypisanie do zmiennej "rabat" nowej ceny
4
5  print("Cena po obniżce to", round(nowa_cena, 2)) # Funkcja "round(x,2)" zaokrągla podaną wartość x do 2 miejsc po przecinku
```

WYNIK PROGRAMU



```
PS C:\Users\wiki_ > & D:/python.exe d:/Pulpit/Lab.Pdst.Prog/Zad4.py
Cena po obniżce to 31.99
```

ANALIZA WYNIKÓW

Program obliczył i wyświetlił wynik 31.99. Formatowanie "round(x,2)" ograniczyło wynik do dwóch miejsc po przecinku.

WNIOSKI

Zadanie 5

CEL ZADANIA

Celem zadania było stworzenie prostego programu obliczającego pole i obwód prostokąta na podstawie danych od użytkownika.

PRZEBIEG ZADANIA

Użytkownik wprowadzał długości boków:

```
a = float(input("Podaj długość boku a: "))
```

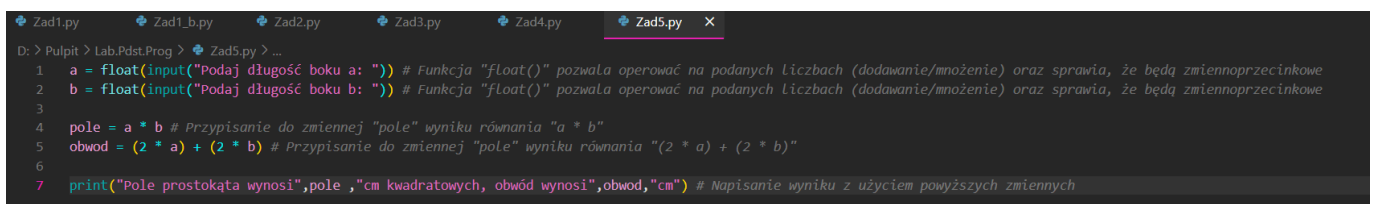
```
b = float(input("Podaj długość boku b: "))

pole = a * b

obwod = 2 * (a + b)

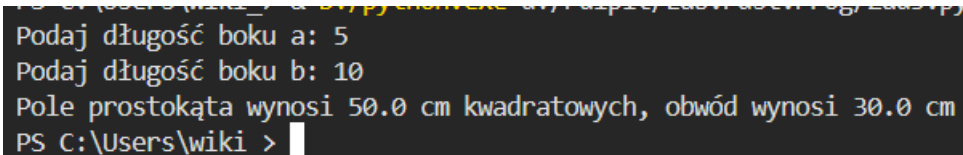
print("Pole prostokąta wynosi", pole, "cm kwadratowych, obwód wynosi", obwod,
      "cm")
```

KOD PROGRAMU



```
Zad1.py Zad1_b.py Zad2.py Zad3.py Zad4.py Zad5.py X
D:\Pulpit > Lab.Pdst.Prog > Zad5.py > ...
1 a = float(input("Podaj długość boku a: ")) # Funkcja "float()" pozwala operować na podanych liczbach (dodawanie/mnożenie) oraz sprawia, że będą zmiennoprzecinkowe
2 b = float(input("Podaj długość boku b: ")) # Funkcja "float()" pozwala operować na podanych liczbach (dodawanie/mnożenie) oraz sprawia, że będą zmiennoprzecinkowe
3
4 pole = a * b # Przypisanie do zmiennej "pole" wyniku równania "a * b"
5 obwod = (2 * a) + (2 * b) # Przypisanie do zmiennej "pole" wyniku równania "(2 * a) + (2 * b)"
6
7 print("Pole prostokąta wynosi",pole, "cm kwadratowych, obwód wynosi",obwod,"cm") # Napisanie wyniku z użyciem powyższych zmiennych
```

WYNIK PROGRAMU



```
Podaj długość boku a: 5
Podaj długość boku b: 10
Pole prostokąta wynosi 50.0 cm kwadratowych, obwód wynosi 30.0 cm
PS C:\Users\wiki_ >
```

ANALIZA WYNIKÓW

Program poprawnie obliczał wartości i zwracał wynik zgodny z danymi wejściowymi.

WNIOSKI

Instrukcja input() umożliwia interakcję z użytkownikiem. Prosty program pokazuje praktyczne zastosowanie podstaw arytmetyki i formatowania danych w Pythonie.

Zadanie 6

CEL ZADANIA

Celem zadania było stworzenie programu obliczającego spalanie i koszt podróży na podstawie danych użytkownika.

PRZEBIEG ZADANIA

Użytkownik podawał długość drogi, średnie spalanie i cenę paliwa. Program obliczał:

$$\text{zuzycie} = (\text{średnie_spalanie} / 100) * \text{droga}$$
$$\text{koszt_podrozy} = \text{zuzycie} * \text{cena_paliwa}$$

Wersja rozszerzona używała "random.randint()" do losowania długości drogi i "f-stringów" do formatowania wyników.

KOD PROGRAMU

```
Zad1.py Zad1_b.py Zad2.py Zad3.py Zad4.py Zad5.py Zad6.py X Zad6_a.py Zad6_b.py
D: > Pulpit > Lab.Pdst.Prog > Zad6.py > ...
1 droga = float(input("Podaj pokonany dystans: ")) # Funkcja "float()" pozwala operować na podanych liczbach (dodawanie/mnożenie/dzielenie) oraz sprawia, że będą zmiennopr.
2 średnie_spalanie = float(input("Podaj średnie spalanie samochodu (w litrach na 100km): ")) # Funkcja "float()" pozwala operować na podanych liczbach (dodawanie/mnożenie/d.
3 cena_paliwa = 6.5 # Cena paliwa za litr
4 zuzycie = (średnie_spalanie / 100) * droga # Obliczanie zużycia paliwa
5 koszt_podrozy = zuzycie * cena_paliwa # Obliczenie kosztu podróży
6
7 print("Przewidywane zużycie paliwa wynosi:", zuzycie, "litrów", "\nSzacunkowy koszt podróży wynosi:", koszt_podrozy, "zł") # Wypisanie wyników z użyciem zmiennych
```

```
Zad1.py Zad1_b.py Zad2.py Zad3.py Zad4.py Zad5.py Zad6.py X Zad6_a.py X Zad6_b.py
D: > Pulpit > Lab.Pdst.Prog > Zad6_a.py > ...
1 import random # Importowanie biblioteki "math" w celu użycia funkcji random.randint()
2
3 droga = random.randint(1, 10000) # Używamy funkcji random.randint() aby wylosować liczbę spośród podanego zakresu
4 średnie_spalanie = float(input("Podaj średnie spalanie samochodu (l/100km): ")) # Funkcja "float()" pozwala operować na podanych liczbach (dodawanie/mnożenie/dzielenie) o.
5 cena_paliwa = float(input("Podaj cenę paliwa (zł/l): ")) # Funkcja "input()" pozwoli użytkownikowi wprowadzić cenę paliwa do zmiennej
6 zuzycie = round(średnie_spalanie / 100 * droga, 2) # Obliczanie zużycia paliwa, funkcja "round()" zaokrągli wynik do 2 miejsc po przecinku
7 koszt_podrozy = round(zuzycie * cena_paliwa, 2) # Obliczenie kosztu podróży, funkcja "round()" zaokrągli wynik do 2 miejsc po przecinku
8
9 print("Przewidywane zużycie paliwa na", droga, "km wynosi:", zuzycie, "litrów", "\nSzacunkowy koszt podróży na", droga, "km wynosi:", koszt_podrozy, "zł") # Wypisanie wyników
```

b)

```
Zad1.py X Zad1_b.py Zad2.py Zad3.py Zad4.py Zad5.py Zad6.py X Zad6_a.py X Zad6_b.py X
D: > Pulpit > Lab.Pdst.Prog > Zad6_b.py > ...
1 import random # Importowanie biblioteki "math" w celu użycia funkcji random.randint()
2
3 droga = random.randint(1, 10000) # Używamy funkcji random.randint() aby wylosować liczbę spośród podanego zakresu
4 średnie_spalanie = float(input("Podaj średnie spalanie samochodu (l/100km): ")) # Funkcja "float()" pozwala operować na podanych liczbach (dodawanie/mnożenie/dzielenie) o.
5 cena_paliwa = float(input("Podaj cenę paliwa (zł/l): ")) # Funkcja "input()" pozwoli użytkownikowi wprowadzić cenę paliwa do zmiennej
6 zuzycie = round(średnie_spalanie / 100 * droga, 2) # Obliczanie zużycia paliwa, funkcja "round()" zaokrągli wynik do 2 miejsc po przecinku
7 koszt_podrozy = round(zuzycie * cena_paliwa, 2) # Obliczenie kosztu podróży, funkcja "round()" zaokrągli wynik do 2 miejsc po przecinku
8
9 print(f"Przewidywane zużycie paliwa wynosi: {zuzycie} litrów, a szacunkowy koszt podróży wynosi: {koszt_podrozy} zł") # Wypisanie wyniku z użyciem zmiennych oraz użycie f-
```

WYNIK PROGRAMU

```
Szacunkowy koszt podróży wynosi: 13.65
PS C:\Users\wiki_ > & D:/python.exe d:/Pulpit/Lab.Pdst.Prog/Zad6.py
Podaj pokonany dystans: 30
Podaj średnie spalanie samochodu (w litrach na 100km): 7
Przewidywane zużycie paliwa wynosi: 2.1 litrów
Szacunkowy koszt podróży wynosi: 13.65 zł
PS C:\Users\wiki_ > 
```

```
Podaj średnie spalanie samochodu (l/100km): 7
Podaj cenę paliwa (zł/l): 5.58
Przewidywane zużycie paliwa na 8139 km wynosi: 569.73 litrów
Szacunkowy koszt podróży na 8139 km wynosi: 3179.09 zł
PS C:\Users\wiki_ > 
```

```
PS C:\Users\wiki_ > & D:/python.exe c:/Users/wiki_/Downloads/zadanie_6b.py
Podaj średnie spalanie samochodu (l/100km): 11
Podaj cenę paliwa (zł/l): 2.69
Przewidywane zużycie paliwa wynosi: 673.31 litrów, a szacunkowy koszt podróży wynosi: 1811.2 zł
PS C:\Users\wiki_ > 
```

ANALIZA WYNIKÓW

Program poprawnie obliczał zużycie paliwa i koszty podróży. Losowanie działało prawidłowo, a formatowanie zapewniało czytelność wyników.

WNIOSKI

Ćwiczenie pokazało zastosowanie biblioteki random, obliczeń matematycznych i dynamicznego wyświetlania wyników przy użyciu f-stringów

Zadanie 7

CEL ZADANIA

Celem zadania było napisanie programu rozwiązującego równanie liniowe $ax + b = 0$ oraz narysowanie schematu blokowego algorytmu.

PRZEBIEG ZADANIA

Użytkownik wprowadzał współczynniki a i b. Program obliczał:

if a == 0:

if b == 0

print("Równanie tożsamościowe")

else:

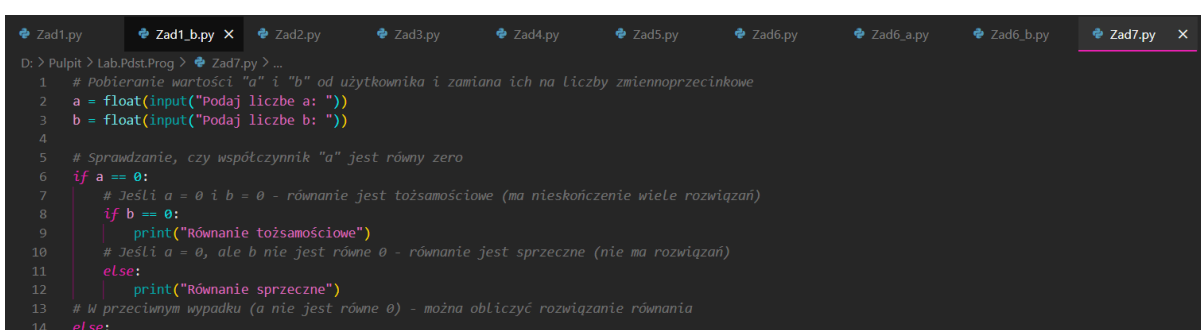
print("Równanie sprzeczne")

else:

X = (b * -1)/a

Schemat blokowy wykonano w draw.io zgodnie z logiką warunkową.

KOD PROGRAMU

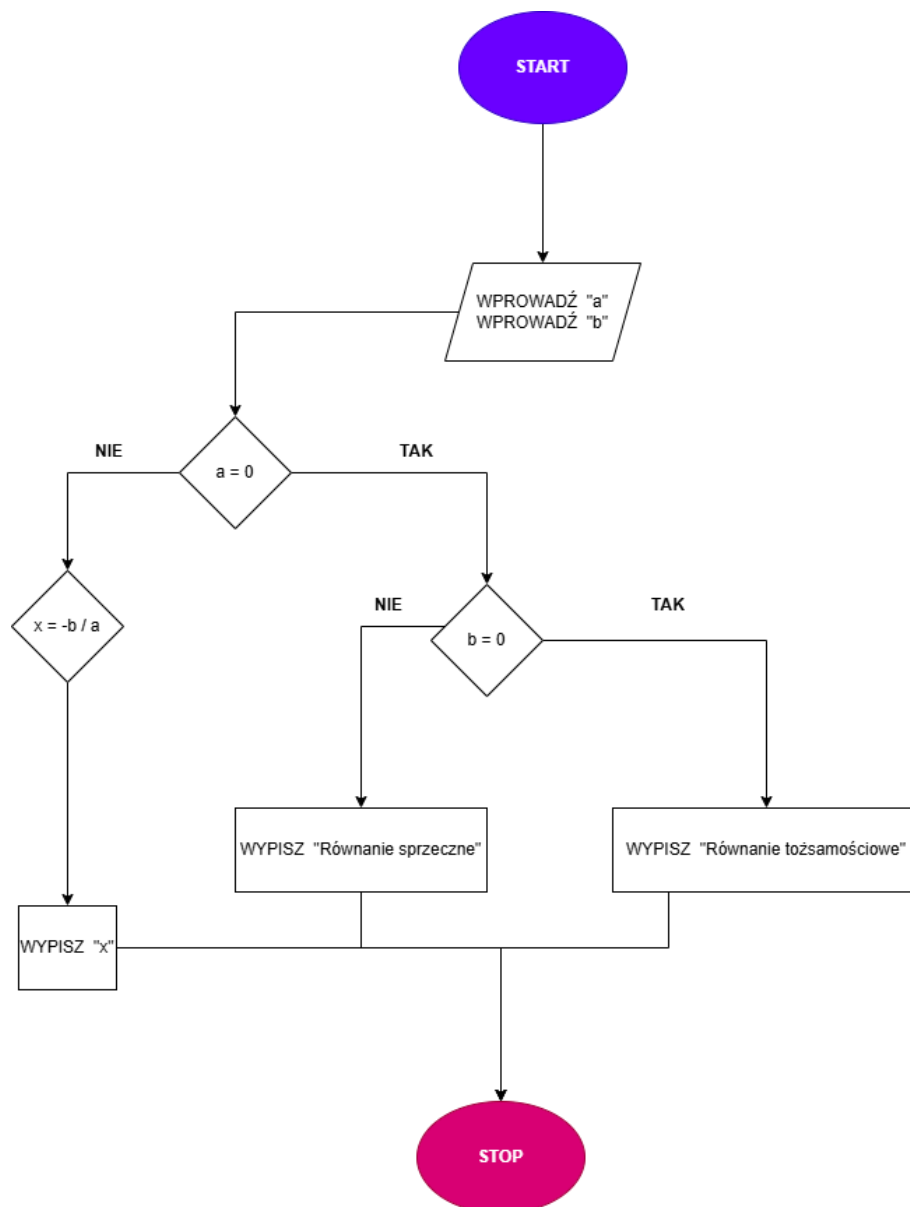


```
Zad1.py  Zad1_b.py X  Zad2.py  Zad3.py  Zad4.py  Zad5.py  Zad6.py  Zad6_a.py  Zad6_b.py  Zad7.py X
D: > Pulpit > Lab.Pdst.Prog > Zad7.py > ...
1 # Pobieranie wartości "a" i "b" od użytkownika i zamiana ich na liczby zmiennoprzecinkowe
2 a = float(input("Podaj liczbę a: "))
3 b = float(input("Podaj liczbę b: "))
4
5 # Sprawdzanie, czy współczynnik "a" jest równy zero
6 if a == 0:
7     # Jeśli a = 0 i b = 0 - równanie jest tożsamościowe (ma nieskończenie wiele rozwiązań)
8     if b == 0:
9         print("Równanie tożsamościowe")
10    # Jeśli a = 0, ale b nie jest równe 0 - równanie jest sprzeczne (nie ma rozwiązań)
11    else:
12        print("Równanie sprzeczne")
13    # W przeciwnym wypadku (a nie jest równe 0) - można obliczyć rozwiązanie równania
14    else:
```


WYNIK PROGRAMU

```
Podaj liczbe a: 5  
Podaj liczbe b: 5  
-1.0  
PS C:\Users\wiki_> █
```

SHCEMAT BLOKOWY



ANALIZA WYNIKÓW

Dla poprawnych danych program zwracał jedno rozwiązanie. Przy $a = 0$ wyświetlał odpowiedni komunikat.

WNIOSKI

Zadanie pozwoliło zrozumieć strukturę warunkową if-else i sposób reprezentacji algorytmu na schemacie blokowym.

CEL ZADANIA

Celem zadania było napisanie programu rozwiązującego równanie kwadratowe $ax^2 + bx + c = 0$ oraz opracowanie jego schematu blokowego.

PRZEBIEG ZADANIA

Program pobierał wartości a, b, c, następnie obliczał deltę i pierwiastki.

Schemat blokowy wykonano w draw.io zgodnie z logiką warunkową.

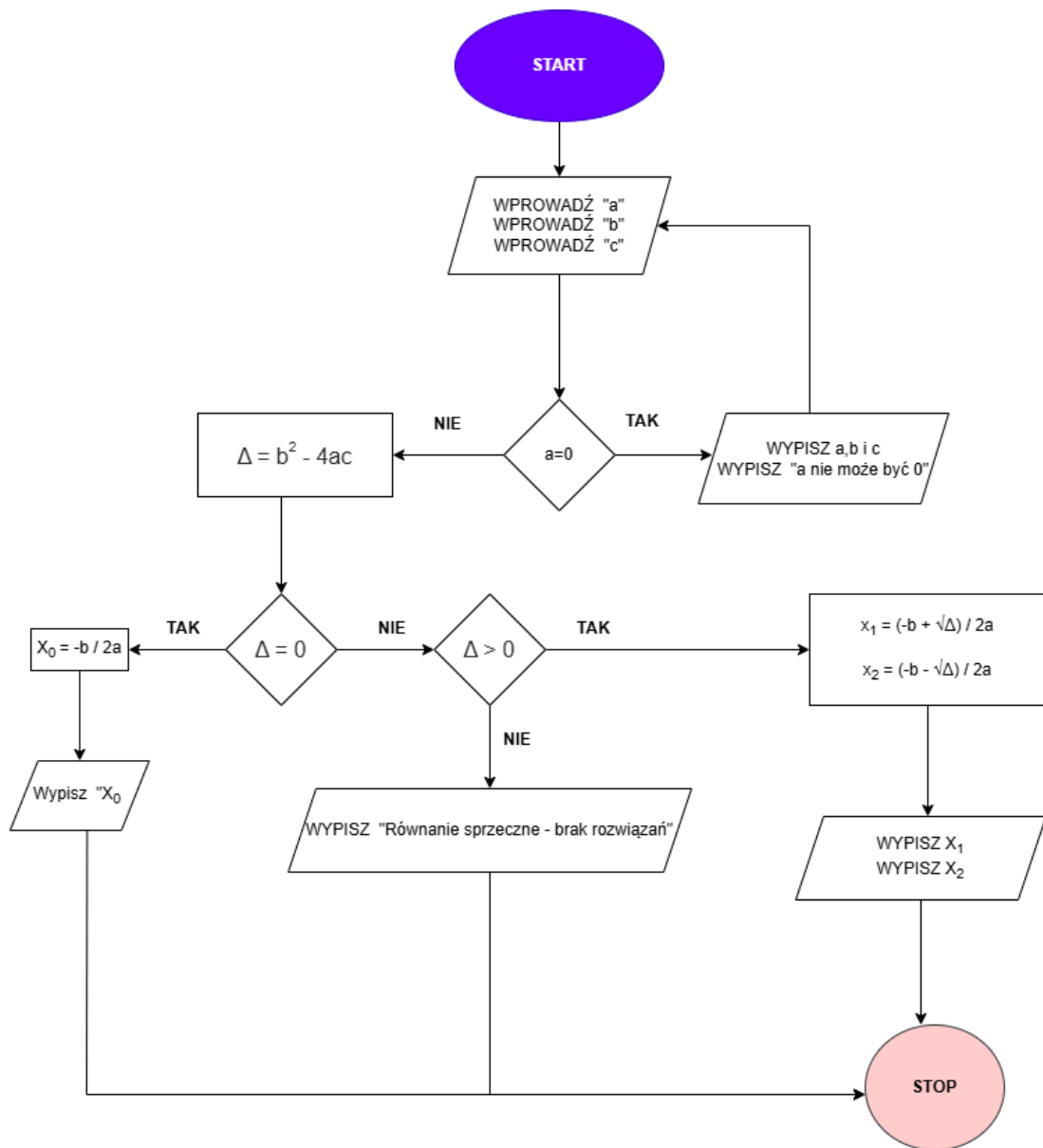
KOD PROGRAMU

```
Zad1.py x Zad1_b.py Zad2.py Zad3.py Zad4.py Zad5.py Zad6.py Zad6_a.py Zad6_b.py Zad7.py Zad8.py x
D: > Pulpit > Lab.Pdst.Prog > Zad8.py > ...
1 from math import sqrt # Importujemy funkcje "sqrt" (pierwiastek kwadratowy) z modulu "math"
2 # Pobranie współczynników równania kwadratowego od użytkownika
3 a = float(input("Podaj a: "))
4 b = float(input("Podaj b: "))
5 c = float(input("Podaj c: "))
6
7 # Sprawdzenie czy a = 0 (bo wtedy równanie nie jest kwadratowe)
8 if a == 0:
9     print("a nie może być równe 0")
10 else:
11     delta = (b ** 2) - (4 * a * c) # Obliczanie delty
12     # Jeśli delta = 0 - równanie ma jedno rozwiązanie
13     if delta == 0:
14         x_zero = (b * -1) / (2 * a)
15         print(f"x0 = {x_zero}")
16     else:
17         # Jeśli delta > 0 - równanie ma dwa rozwiązania
18         if delta > 0:
19             pierwiastek_z_delta = sqrt(delta)
20             x_jeden = round(((b * -1) + pierwiastek_z_delta) / (2 * a),2) # Obliczanie pierwszego pierwiastka równania
21             x_dwa = round(((b * -1) - pierwiastek_z_delta) / (2 * a),2) # Obliczanie drugiego pierwiastka równania
22             print(f"x1 = {x_jeden} x2 = {x_dwa}") # Wypisanie wyników
23         # Jeśli delta < 0 - brak rozwiązań
24     else:
25         print("Równanie sprzeczne")
26
```

WYNIK PROGRAMU

```
PS C:\Users\wiki_> & D:/python.exe c:/Users/wiki_/Downloads/zadanie_8.py
Podaj a: 7
Podaj a: 7
Podaj b: 5
Podaj c: 2
Równanie sprzeczne
PS C:\Users\wiki_> & D:/python.exe c:/Users/wiki_/Downloads/zadanie_8.py
Podaj a: 16
Podaj b: 8
Podaj c: 1
X0 = -0.25
PS C:\Users\wiki_> 
```

SCHEMAT BLOKOWY



ANALIZA WYNIKÓW

Program poprawnie rozpoznawał liczbę rozwiązań i obliczał je z użyciem wzoru kwadratowego. Schemat blokowy był zgodny z logiką programu.

WNIOSKI

Zadanie utrwaliło wiedzę o instrukcjach warunkowych i działaniach matematycznych w Pythonie. Pozwoliło zrozumieć zastosowanie algorytmu w praktyce.

Zadanie 9

CEL ZADANIA

Celem zadania było stworzenie prostego kalkulatora wykonującego podstawowe działania matematyczne.

PRZEBIEG ZADANIA

Użytkownik podawał dwie liczby. Program wykonywał i wyświetlał:

dodawanie = $a + b$

odejmowanie = $a - b$

mnożenie = $a * b$

dzielenie = a / b

dzielenie_calkowite = $a // b$

potegowanie = $a ** b$

KOD PROGRAMU

```
py  Zad1_b.py X  Zad2.py  Zad3.py  Zad4.py  Zad5.py  Zad6.py  Zad6_a.py  Zad6_b.py  Zad7.py  Zad8.py  Zad9.py X
D:\> Pulpit > LabPdat.Prog > Zad9.py > ...
1  # Pobieranie wartości "a" i "b" od użytkownika i zamiana ich na liczby zmiennoprzecinkowe
2  a = float(input("Podaj pierwszą liczbę: "))
3  b = float(input("Podaj drugą liczbę: "))
4  # Sprawdzenie czy użytkownik nie dzieli przez 0
5  if b == 0:
6      print("Nie można dzielić przez 0")
7  # Jeśli b nie jest zerem - zostają wykonane wszystkie działania matematyczne
8  else:
9      dodawanie = a + b
10     odejmowanie = a - b
11     mnozenie = a * b
12     dzielenie = a / b
13     dzielenie_calkowite = a // b
14     potegowanie = a ** b
15     # Wypisanie wyników działań
16     print()
17     print("Dodawanie:", dodawanie)
18     print("Odejmowanie: ", odejmowanie)
19     print("Mnozenie:", mnozenie)
20     print("Dzielenie z resztą:", dzielenie)
21     print("Dzielenie całkowite:", dzielenie_calkowite)
22     print("Potegowanie:", potegowanie)
```

WYNIK PROGRAMU

```
PS C:\Users\wiki_> & D:/python.exe d:/Pulpit/Lab.Pdst.Prog/Zad9.py
Podaj pierwszą liczbę: 10
Podaj drugą liczbę: 5

Dodawanie: 15.0
Odejmowanie: 5.0
Mnożenie: 50.0
Dzielenie z resztą: 2.0
Dzielenie całkowite: 2.0
Potęgowanie: 100000.0
PS C:\Users\wiki_> █
```

ANALIZA WYNIKÓW

Program poprawnie wykonywał obliczenia dla wszystkich operatorów. Przy dzieleniu przez 0 zwracał błąd, co pozwoliło zrozumieć znaczenie obsługi wyjątków.

WNIOSKI

Kalkulator jest praktycznym przykładem wykorzystania operatorów arytmetycznych. Pokazuje znaczenie walidacji danych i obsługi błędów.