

ZADANIE 1.A

Cel zadania: Poznanie działania operatorów w Pythonie.

Przebieg zadania: Podanie w materiałach przykłady testujemy w interpreterze.

Kod programu:

```
1  #Amelia Kawik 73141
2
3  x1 = 1 + 2
4  print("Typ wyników", type(x1))
5  print ("Wynik x1:", x1)
6
7  x2 = 1 + 4.5
8  print(" \nTyp wyników", type(x2))
9  print ("Wynik x2:", x2)
10
11 x3 = 3 / 2
12 print(" \nTyp wyników", type(x3))
13 print ("Wynik x3:", x3)
14
15 x4 = 4 / 2
16 print(" \nTyp wyników", type(x4))
17 print ("Wynik x4:", x4)
18
19 x5 = 3 // 2
20 print(" \nTyp wyników", type(x5))
21 print ("Wynik x5:", x5)
22
23 x6 = -3 // 2
24 print(" \nTyp wyników", type(x6))
25 print ("Wynik x6:", x6)
26
27 x7 = 11 % 2
28 print(" \nTyp wyników", type(x7))
29 print ("Wynik x7:", x7)
30
31 x8 = 2 ** 10
32 print(" \nTyp wyników", type(x8))
33 print ("Wynik x8:", x8)
34
35 x9 = 8 ** 2
36 print(" \nTyp wyników", type(x9))
37 print ("Wynik x9:", x9)
38
39 #Amelia Kawik 73141
```

Wynik programu:

```
Typ wyników <class 'int'>
Wynik x1: 3

Typ wyników <class 'float'>
Wynik x2: 5.5

Typ wyników <class 'float'>
Wynik x3: 1.5

Typ wyników <class 'float'>
Wynik x4: 2.0

Typ wyników <class 'int'>
Wynik x5: 1

Typ wyników <class 'int'>
Wynik x6: -2

Typ wyników <class 'int'>
Wynik x7: 1

Typ wyników <class 'int'>
Wynik x8: 1024

Typ wyników <class 'int'>
Wynik x9: 64
```

Analiza wyników:

x1 - Operator + w typie **int** służy do dodawania liczb całkowitych i zwraca liczbę całkowitą.

x2 - Operator + w typie **float** służy do dodawania liczb całkowitych i zwraca liczbę zmiennoprzecinkową.

x3 - Operator / służy do dzielenia liczb całkowitych i zwraca liczbę zmiennoprzecinkową.

x4 - Operator / służy do dzielenia liczb całkowitych i zwraca liczbę zmiennoprzecinkową.

x5 - Operator // służy do dzielenia liczb całkowitych i zwraca liczbę całkowitą.

x6 - Operator // służy do dzielenia liczb całkowitych i zwraca liczbę całkowitą zaokrągloną w dół.

x7 - Operator % - modulo, służy do obliczania reszty z dzielenia liczb całkowitych.

x8 - Operator ** służy do potęgowania liczb całkowitych.

x9 - Operator ****** służy do potęgowania z wykładnikiem ułamkowym.

Wnioski: Każdy typ zmiennych służy do zwracania innych typów danych, w zależności od użytych operatorów arytmetycznych.

ZADANIE 1.B

Cel zadania: Wyjaśnienie działanie typów zmiennych.

Przebieg zadania: Podanie w materiałach przykłady testujemy w interpreterze.

Kod programu:

```
1  #Amelia Kawik 73141
2
3  print ("B")
4
5  x1 = int(3.0)
6
7  x2 = float(3)
8
9  x3 = float("3")
10
11 x4 = str(12.4)
12
13 x5 = bool(0)
14
15 print ("\n x1:",x1)
16
17 print ("\n x2:",x2)
18
19 print ("\n x3:",x3)
20
21 print ("\n x4:",x4)
22
23 print ("\n x5:",x5)
```

Wynik programu:

```
B

x1: 3

x2: 3.0

x3: 3.0

x4: 12.4

x5: False
```

Analiza wyników:

x1 – int zwraca liczbę całkowitą

x2 - float zwraca liczbę zmiennoprzecinkową

x3 - float zwraca liczbę zmiennoprzecinkową

x4 - str zwraca łańcuch znaków

x5 - bool zwraca wartość logiczną

Wnioski: Każdy typ zmiennych służy do zwracania innych typów danych.

ZADANIE 2.

Cel zadania: Sprawdzenie jak działa przypisanie danych (typ string) do zmiennych.

Przebieg zadania: Do zmiennej o nazwie *uczelnia* przypisujemy zdanie *Studiuję na WSIiZ*, Następnie drukujemy ten tekst do konsoli.

Kod programu:

```
1  #Amelia Kawik 73141
2
3  print("Zadanie drugie \n")
4
5  uczelnia = "Studiuję na WSIiZ"
6
7  print(uczelnia)
8
9  #Amelia Kawik 73141
```

Wynik programu:

```
Zadanie drugie
Studiuję na WSIiZ
```

Analiza wyników:

print("Zadanie drugie") – wypisuje tekst Zadanie drugie.

uczelnia = "Studiuję na WSIiZ" – tworzy zmienną uczelnia, przypisując jej ciąg znaków (string).

print(uczelnia) – wypisuje zawartość zmiennej, czyli Studiuję na WSIiZ.

Wnioski: Funkcją **print()** można zarówno drukować tekst (podany w „”) lub wcześniej zadeklarowane zmienne.

ZADANIE 3.

Cel zadania: Sprawdzenie działania funkcji **print()**.

Przebieg zadania: Podane w materiałach zmienne wypisujemy w konsoli w podany sposób.

Kod programu:

```
1 #Amelia Kawik 73141
2
3 print("Zadanie trzecie \n")
4
5 Imię = 'Jan'
6
7 Wiek = 20
8
9 Wzrost = 178
10
11 print ("Nazywam się", Imię, "i mam", Wiek, "lat. \nMój wzrost to", Wzrost, "cm.")
12
13 #Amelia Kawik 73141
```

Wynik programu:

```
Zadanie trzecie

Nazywam się Jan i mam 20 lat.
Mój wzrost to 178 cm.
```

Analiza wyników: Funkcja `print ()` drukuje w konsoli podane w cudzysłowie tekst, a po przecinku drukuje przypisane do zmiennych dane

Wnioski: Funkcja `print ()` służy do drukowania dowolnych tekstów oraz wcześniej zadeklarowanych zmiennych, można tego używać naprzemiennie.

ZADANIE 4

Cel zadania: Użycie i sprawdzenie działanie funkcji **round** .

Przebieg zadania: Napisanie skryptu na podstawie podanych w materiałach instrukcji.

Kod programu:

```
1  #Amelia Kawik 73141
2
3  print("Zadanie czwarte \n")
4
5  Cena = 39.99
6
7  Rabat = 0.2
8
9  Cena_po_rabacie = Cena - (Cena * Rabat)
10 print("Cena po rabacie wynosi:", round(Cena_po_rabacie, 2))
11
12 #Amelia Kawik 73141
```

Wynik programu:

Zadanie czwarte

Cena po rabacie wynosi: 31.99

Analiza wyników: Wykorzystano w tym kodzie operatora arytmetycznego `*` do mnożenia liczb, w tym wypadku do obliczenia rabatu. Następnie do funkcji `Cena_po_rabacie` przypisano wynik tego działania. Funkcja `round` zaokrągliła wynik do dwóch miejsc po przecinku (czyli do groszy).

Wnioski: Funkcja **round** służy do zaokrąglania liczb do podanego przez nas miejsca po przecinku.

ZADANIE 5.

Cel zadania: Napisanie skryptu do obliczenia pola i obwodu prostokąta z podanych przed użytkownika danych.

Przebieg zadania: Napisanie skryptu do obliczenia pola i obwodu prostokąta.

Kod programu:

```
1  #Amelia Kawik 73141
2
3  print("Zadanie piąte \n")
4
5  a = int(input("Podaj pierwszy bok prostokąta:"))
6
7  b = int(input(" \nPodaj drugi bok prostokąta:"))
8
9  pole = a * b
10
11 obwod = 2 * (a + b)
12
13 print("\nPole prostokąta:", pole, "\n")
14
15 print("\nObwód prostokąta:", obwod)
16
17 # #Amelia Kawik 73141
```

Wynik programu:

```
Zadanie piąte

Podaj pierwszy bok prostokąta:2

Podaj drugi bok prostokąta:4

Pole prostokąta: 8

Obwód prostokąta: 12
```

Analiza wyników:

print("Zadanie piąte \n") – wypisuje tytuł zadania i przechodzi do nowej linii.

a = int(input(...)) – prosi użytkownika o wpisanie długości pierwszego boku i zamienia wartość na liczbę całkowitą.

b = int(input(...)) – to samo dla drugiego boku.

pole = a * b – oblicza pole prostokąta.

obwod = 2 * (a + b) – oblicza obwód prostokąta.

Dwa ostatnie print() wypisują wyniki.

Wnioski: Funkcja input() pobiera dane wpisane przez użytkownika w konsoli jako tekst (string), funkcja int() zamienia ten tekst na liczbę całkowitą (integer), dzięki czemu można wykonywać działania matematyczne.

ZADANIE 6

Cel zadania: Obliczenie szacowanego zużycia paliwa w trakcie przebytej drogi. Użycie typów zmiennoprzecinkowych.

Przebieg zadania: Napisanie skryptu na podstawie instrukcji

Kod programu:

```
1  #Amelia Kawik 73141
2
3  print("Zadanie szoste")
4
5  droga = float(input("Podaj drogę (w kilometrach): "))
6
7  spalanie = float(input("Podaj spalanie (na 100 km): "))
8
9  cena_paliwa = 6.5  # zł za litr
10
11  ilosc_paliwa = (droga / 100) * spalanie
12
13  koszt_podrozy = ilosc_paliwa * cena_paliwa
14
15  print("Ilość potrzebnego paliwa:", round(ilosc_paliwa, 2), "litrów")
16
17  print("Koszt podróży wynosi:", round(koszt_podrozy, 2), "zł")
18
19  #Amelia Kawik 73141
```

Wynik programu:

```
Zadanie szoste

Podaj drogę (w kilometrach): 115

Podaj spalanie (na 100 km): 5.1

Ilość potrzebnego paliwa: 5.86 litrów

Koszt podróży wynosi: 38.12 zł
```

Analiza wyników: Program oblicza, ile paliwa potrzeba na przejechanie określonej trasy oraz jaki będzie koszt podróży przy podanym spalaniu i cenie paliwa (6.5 zł za litr jak w przykładzie). Wyniki są zaokrąglone do dwóch miejsc po przecinku.

Wnioski: Użycie `float(input())` pozwala na wprowadzenie wartości z przecinkiem. Wynik jest zaokrąglony do dwóch miejsc po przecinku, dzięki czemu przypomina rzeczywiste obliczenia kosztów.

ZADANIE 6.A

Przebieg zadania: Zmodyfikuj skrypt tak, aby długość przejechanej drogi była generowana losowo (liczba całkowita z zakresu), a użytkownik podawał aktualną cenę paliwa za litr.

Kod programu:

```
22  print("\nA")
23
24  losowa_droga = random.randint(50, 500)
25
26  print("\nDroga do przebycia wynosi:", losowa_droga, "km")
27
28  aktualna_cena_paliwa = float(input("\nPodaj aktualną cenę paliwa (za litr): "))
29
30  ilosc_paliwa2 = (losowa_droga / 100) * spalanie
31
32  przewidywany_koszt2 = ilosc_paliwa2 * aktualna_cena_paliwa
33
34  print("\nIlość potrzebnego paliwa wynosi:", round(ilosc_paliwa2, 2), "litrów")
35
36  print("\nPrzewidywany koszt podróży wynosi:", round(przewidywany_koszt2, 2), "zł")
37
```

Wynik programu:

```
A
Droga do przebycia wynosi: 171 km
Podaj aktualną cenę paliwa (za litr): 3.4
Ilość potrzebnego paliwa wynosi: 8.55 litrów
Przewidywany koszt podróży wynosi: 29.07 zł
```

Analiza wyników: `random.randint` wygenerował losową długość przejechanej drogi z zakresu od 50 do 500 km, następnie ta droga jest wyświetlana, potem kod pyta użytkownika o obecną cenę paliwa i zmienia wartość na liczbę zmiennoprzecinkową, funkcja `ilosc_paliwa2` oblicza ilość paliwa potrzebnego do pokonania tej drogi,

przewidywany_koszt2 oblicza koszt podróży, mnożąc potrzebne paliwo przez aktualną cenę. Round zaokrągla wynik do 2 miejsc po przecinku

Wnioski: Program pozwala użytkownikowi wprowadzić aktualną cenę paliwa, poprawnie oblicza ilość paliwa i koszt podróży a round zaokrągla wynik.

ZADANIE 6.B

Przebieg zadania: Zmodyfikuj zadania 6 tak, aby wyświetlanie wyników wykorzystywało f-string.

Kod programu:

```
39  print("\nB")
40
41  losowa_droga2 = random.randint(50, 500)
42
43  print("\nDroga do przebycia wynosi:", losowa_droga2, "km")
44
45  cena_aktualnego_paliwa = float(input("\nPodaj aktualną cenę paliwa (za litr): "))
46
47  ilosc_paliwa2 = (losowa_droga2 / 100) * spalanie
48
49  przewidywany_koszt2 = ilosc_paliwa2 * cena_aktualnego_paliwa
50
51  wynik1 = f"\nIlość potrzebnego paliwa wynosi: {round(ilosc_paliwa2, 2)} litrów"
52
53  wynik2 = f"\nPrzewidywany koszt podróży wynosi: {round(przewidywany_koszt2, 2)} zł"
54
55  print(wynik1)
56
57  print(wynik2)
58
59  #Amelia Kawik 73141
```

Wynik programu:

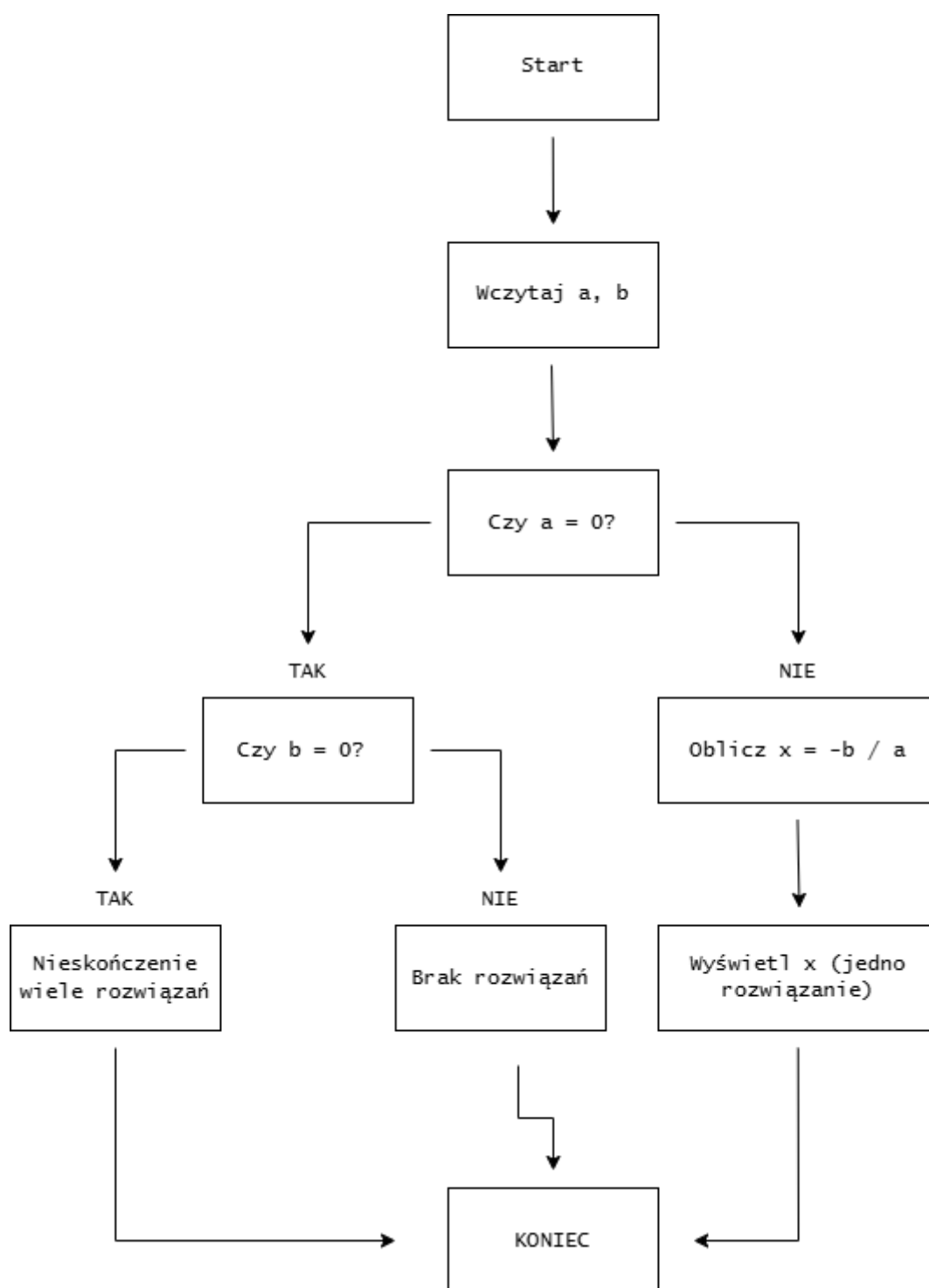
```
B
Droga do przebycia wynosi: 425 km
Podaj aktualną cenę paliwa (za litr): 4.2
Ilość potrzebnego paliwa wynosi: 21.25 litrów
Przewidywany koszt podróży wynosi: 89.25 zł
```

Analiza wyników: `random.randint` wygenerował losową długość przejechanej drogi z zakresu od 50 do 500 km, następnie ta droga jest wyświetlana, potem kod pyta użytkownika o obecną cenę paliwa i zmienia wartość na liczbę zmiennoprzecinkową, `ilosc_paliwa2` oblicza ilość paliwa potrzebnego do pokonania wylosowanej trasy, `przewidywany_koszt2` oblicza koszt w zł, `wynik1` i `wynik2` tworzą ładnie sformatowane łańcuchy tekstowe z wynikami, zaokrąglone do dwóch miejsc po przecinku. `Print()` je wyświetla

Wnioski: Użycie f-stringów sprawia, że wyniki są czytelne i estetyczne, program pozwala użytkownikowi wprowadzić aktualną cenę paliwa, poprawnie oblicza ilość paliwa i koszt podróży a `round` zaokrągla wynik.

ZADANIE 7

Cel zadania: Zastosowanie funkcji `if/else`

Schemat blokowy:

Przebieg zadania: Program pobiera od użytkownika a i b i sprawdza trzy przypadki, i wyświetla odpowiedni wynik.

Kod programu:

```

1  #Amelia Kawik 73141
2
3  print("Zadanie siodme ")
4
5
6  a = float(input("\nPodaj a: "))
7
8  b = float(input("\nPodaj b: "))
9
10 if a != 0:
11     x = -b / a
12
13     print(f"\nRozwiązanie równania: x = {x}")
14
15 else:
16     if b == 0:
17         print("\nRównanie ma nieskończenie wiele rozwiązań.")
18     else:
19         print("\nRównanie sprzeczne □ brak rozwiązania.")
20
21
22
23
24
25
26  #Amelia Kawik 73141

```

Wynik programu:

<p>Zadanie siodme</p> <p>Podaj a: 0</p> <p>Podaj b: 5</p> <p>Równanie sprzeczne – brak rozwiązania.</p>	<p>Zadanie siodme</p> <p>Podaj a: 2</p> <p>Podaj b: -4</p> <p>Rozwiązanie równania: x = 2.0</p>
<p>Zadanie siodme</p> <p>Podaj a: 0</p> <p>Podaj b: 0</p> <p>Równanie ma nieskończenie wiele rozwiązań.</p>	

Analiza wyników:

Przypadek, gdy $a \neq 0$

Program oblicza rozwiązanie równania:

$$x = -\frac{b}{a}$$

Przypadek, gdy $a = 0$ i $b = 0$

Równanie staje się tożsamościowe: $0 * x + 0 = 0$, program wyświetla informację: „*Równanie ma nieskończenie wiele rozwiązań.*”

Przypadek, gdy $a = 0$ i $b \neq 0$

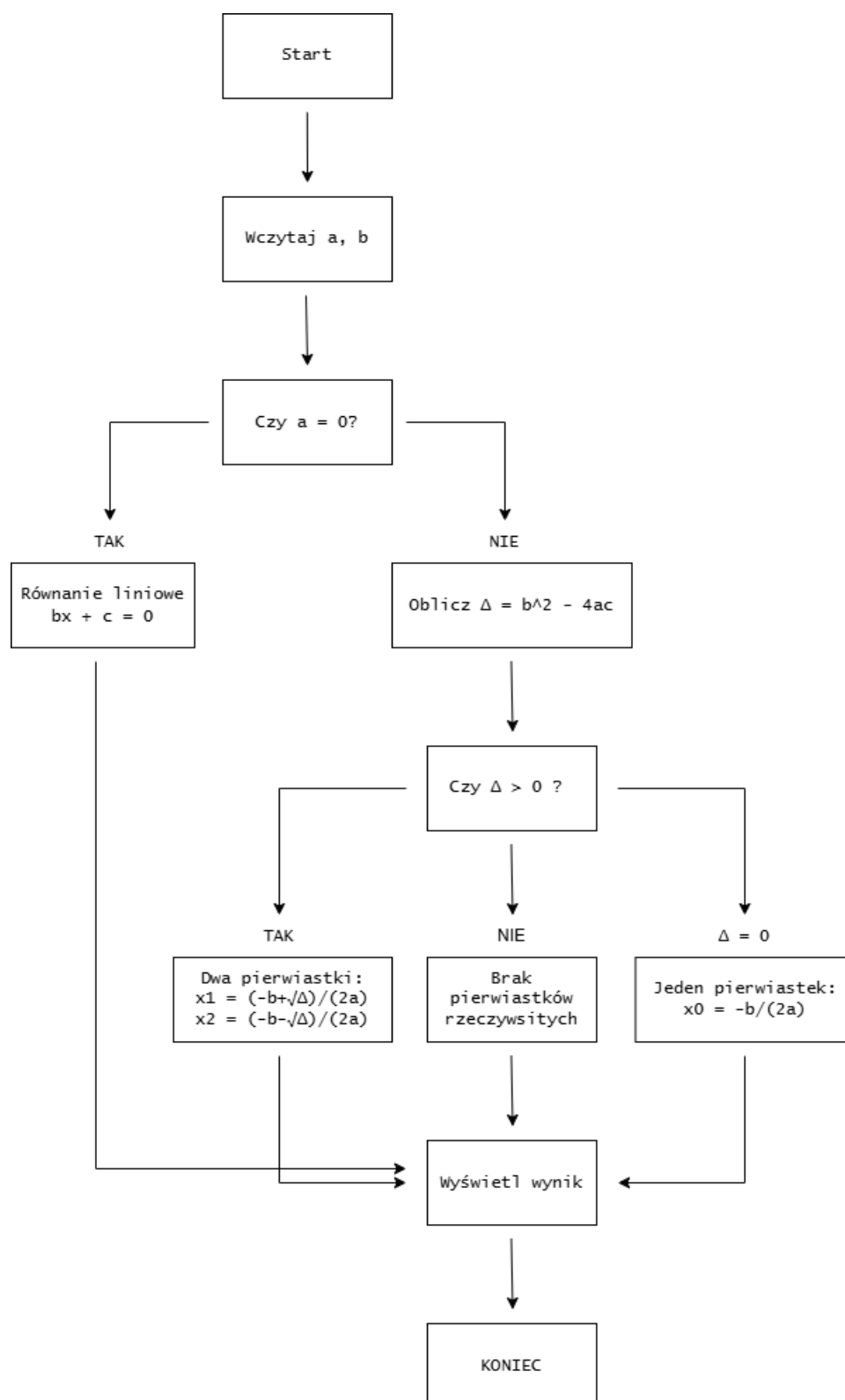
Równanie staje się sprzeczne: $0 * x + b = 0 \rightarrow$ niemożliwe do spełnienia, program wyświetla informację: „*Równanie sprzeczne – brak rozwiązania.*”

Wnioski: Program działa poprawnie dla wszystkich możliwych przypadków równania liniowego, dzięki warunkowi if $a \neq 0$ unikamy dzielenia przez zero

ZADANIE 8

Cel zadania: Zastosowanie funkcji if/else

Schemat blokowy:



Przebieg zadania: Program pobiera od użytkownika a, b i c sprawdza trzy przypadki, i wyświetla odpowiedni wynik.

Kod programu:

```
1  #Amelia Kawik 73141
2
3  print("Zadanie ósme")
4
5  a = float(input("\nPodaj a: "))
6
7  b = float(input("\nPodaj b: "))
8
9  c = float(input("\nPodaj c: "))
10
11 delta = b**2 - 4*a*c
12
13 if delta > 0:
14     x1 = (-b + delta**0.5) / (2*a)
15
16     x2 = (-b - delta**0.5) / (2*a)
17
18     print("\nRównanie ma dwa różne rozwiązania:")
19
20     print("x1 =", x1)
21
22     print("x2 =", x2)
23
24 elif delta == 0:
25
26     x = -b / (2*a)
27
28     print("\nRównanie ma jedno podwójne rozwiązanie:")
29
30     print("x =", x)
31
32 else:
33
34     print("\nRównanie nie ma rozwiązań rzeczywistych.")
35
36 #Amelia Kawik 73141
```

Wynik programu:

<pre> Zadanie ósme Podaj a: 1 Podaj b: -5 Podaj c: 6 Równanie ma dwa różne rozwiązania: x1 = 3.0 x2 = 2.0 </pre>	<pre> Zadanie ósme Podaj a: 1 Podaj b: -2 Podaj c: 1 Równanie ma jedno podwójne rozwiązanie: x = 1.0 </pre>
<pre> Zadanie ósme Podaj a: 1 Podaj b: 2 Podaj c: 5 Równanie nie ma rozwiązań rzeczywistych. </pre>	

Analiza wyników:

$\Delta > 0 \rightarrow$ dwa różne pierwiastki:

$$x_1 = \frac{-b + \sqrt{\Delta}}{2a}, x_2 = \frac{-b - \sqrt{\Delta}}{2a}$$

$\Delta == 0 \rightarrow$ jedno podwójne rozwiązanie:

$$x = \frac{-b}{2a}$$

$\Delta < 0 \rightarrow$ brak rozwiązań rzeczywistych.

Wnioski: Program obsługuje wszystkie możliwe przypadki równania kwadratowego,

ZADANIE 9

Cel zadania: Stworzenie prostego kalkulatora

Przebieg zadania: Użycie funkcji if oraz operatorów arytmetycznych

Kod programu:

```
1  #Amelia Kawik 73141
2
3  print("Zadanie dziewiąte")
4
5  print("\nProsty kalkulator")
6
7  a = float(input("\nPodaj pierwszą liczbę: "))
8
9  b = float(input("\nPodaj drugą liczbę: "))
10
11  Dodawanie = a + b
12
13  Odejmowanie = a - b
14
15  Mnozenie = a * b
16
17  if b != 0:
18      Dzielenie = a / b
19
20
21  else:
22      print("\nNie można dzielić przez zero.")
23
24
25  Potegowanie = a ** b
26
27  print("\nDodawanie:", Dodawanie)
28
29  print("\nOdejmowanie:", Odejmowanie)
30
31  print("\nMnozenie:", Mnozenie)
32
33  if b != 0:
34      print("\nDzielenie:", Dzielenie)
35
36  print("\nPotęgowanie:", Potegowanie)
37
38  #Amelia Kawik 73141
```

Wynik programu:

Zadanie dziewiąte	Zadanie dziewiąte
Prosty kalkulator	Prosty kalkulator
Podaj pierwszą liczbę: 4	Podaj pierwszą liczbę: 13
Podaj drugą liczbę: 6	Podaj drugą liczbę: 0
Dodawanie: 10.0	Nie można dzielić przez zero.
Odejmowanie: -2.0	Dodawanie: 13.0
Mnożenie: 24.0	Odejmowanie: 13.0
Dzielenie: 0.6666666666666666	Mnożenie: 0.0
Potęgowanie: 4096.0	Potęgowanie: 1.0

Analiza wyników: Przy użyciu operatora arytmetycznego „+” kalkulator dodaje podane liczby, „-” odejmuje je, „*” mnoży liczby a „/” dzieli je.

Wnioski: Program wykonuje wszystkie podstawowe działania arytmetyczne dla dwóch liczb rzeczywistych, użycie `if b != 0` zapobiega błędom przy dzieleniu.

Link do GitHub: <https://github.com/gombok>