

## 2.8. Algorytmy, schematy, programy

### DOWIESZ SIĘ

- co oznaczają pojęcia: *algorytm*, *schemat blokowy*, *język programowania*,
- jakie są sposoby obliczania największego wspólnego dzielnika (NWD).

### ALGORYTM EUKLIDESA

**Algorytm** opisuje sposób rozwiązania problemu krok po kroku. Opis ten powinien być dokładny i nie pozostawiać wątpliwości, co, jak i w jakiej kolejności trzeba zrobić. Z algorytmami masz często do czynienia na lekcjach matematyki. Jednym z przykładów algorytmu jest sposób znajdowania **największego wspólnego dzielnika** dwóch liczb całkowitych (**NWD**). NWD jest potrzebny np. przy skracaniu ułamków.

Algorytmem poznanym przez ciebie na lekcjach matematyki jest sprawdzanie podzielności dwóch liczb całkowitych przez kolejne liczby. Najpierw sprawdzamy podzielność przez 2, potem – przez 3 i tak dalej, aż do mniejszej z dwóch liczb. Największym wspólnym dzielnikiem dwóch liczb jest iloczyn ich wszystkich wspólnych dzielników pierwszych. Przyjrzyj się temu sposobowi na przykładzie szukania NWD liczb:  $a = 42$  i  $b = 24$ .

Liczba $a$	Podzielniki $a$	Liczba $b$	Podzielniki $b$
42	2	24	2
21	3	12	2
7	7	6	2
1	1	3	3
		1	1

NWD liczb  $a$  i  $b$  wynosi więc  $2 \cdot 3 \cdot 1 = 6$ .

Lepszym i szybszym rozwiązaniem problemu szukania NWD dwóch liczb całkowitych jest **algorytm Euklidesa**. To jeden z najstarszych algorytmów, opisany przez Euklidesa ok. 300 roku p.n.e. Opiera się na spostrzeżeniu: jeśli od większej liczby odejmiemy mniejszą, to mniejsza liczba i otrzymana różnica będą miały taki sam największy wspólny dzielnik jak pierwotne liczby. Gdy w wyniku kolejnego odejmowania otrzymamy parę takich samych liczb, to oznacza, że znaleźliśmy NWD.

Popatrz, jak przebiega znajdowanie NWD dwóch liczb  $a = 42$  i  $b = 24$  według tego algorytmu.

Liczba $a$	Liczba $b$	NWD
42	24	6
$42 - 24 = 18$	24	6
18	$24 - 18 = 6$	6
$18 - 6 = 12$	6	6
$12 - 6 = 6$	6	6

Oto algorytm Euklidesa zapisany w postaci planu działań.

1. Wybieramy dwie liczby naturalne.
2. Jeśli liczby są równe, to NWD jest np. pierwszą z nich i to oznacza koniec działań.
3. Jeśli liczby nie są sobie równe, to trzeba:
  - zbadać, która jest większa;
  - odjąć od niej mniejszą i zastąpić większą przez otrzymaną różnicę;
  - wrócić do sprawdzenia warunku w punkcie 2. (pętla).

■ Wykonaj ten algorytm dla następujących par liczb: 152 i 57, 1025 i 725, 132 i 44.

## SCHEMAT BLOKOWY ALGORYTMU

Jednym ze sposobów przedstawiania algorytmów są schematy blokowe. Można je budować z gotowych elementów w specjalnych programach. Posłużymy się jednym z takich programów – Elbox Laboratorium Informatyki (w wersji ELI Multi Plus 2.0). Schemat blokowy algorytmu zbudowany w programie ELI składa się z następujących klocków: początek algorytmu, wprowadzenie danych, wykonanie obliczeń, sprawdzenie warunku, wyprowadzenie wyniku, koniec algorytmu (rys. 1).

1. Wczytaniu dwóch liczb naturalnych **a** i **b** odpowiadają dwa kolejne klocki



2. Sprawdzenie warunku, czy  $a = b$  . Jeśli tak, to **NWD** jest np.

pierwszą z nich, więc następuje wyprowadzenie wyniku i koniec algorytmu .

3. Jeśli liczby nie są sobie równe, to następuje sprawdzenie warunku, czy  $a > b$  .

- Jeśli większa jest liczba **a**, to jest zastępowana przez  $a := a - b$

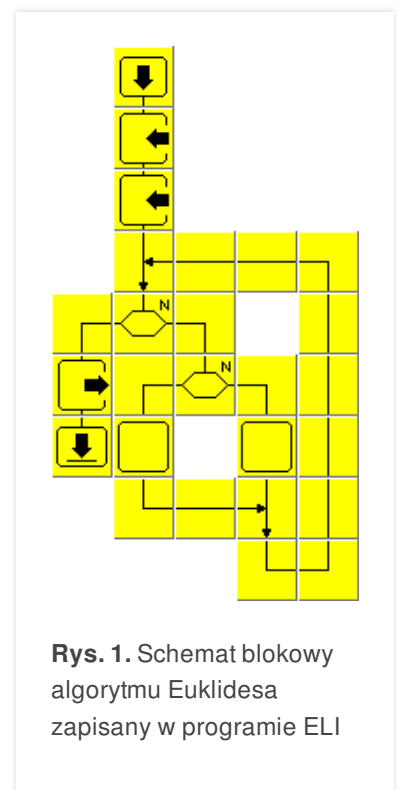


- Jeśli większa jest liczba **b**, to jest zastępowana przez  $b := b - a$



- Powrót do punktu 2. (pętla), czyli do sprawdzania warunku.

Tak zapisany algorytm można uruchomić w programie ELI i obserwować krok po kroku, jak jest wykonywany.

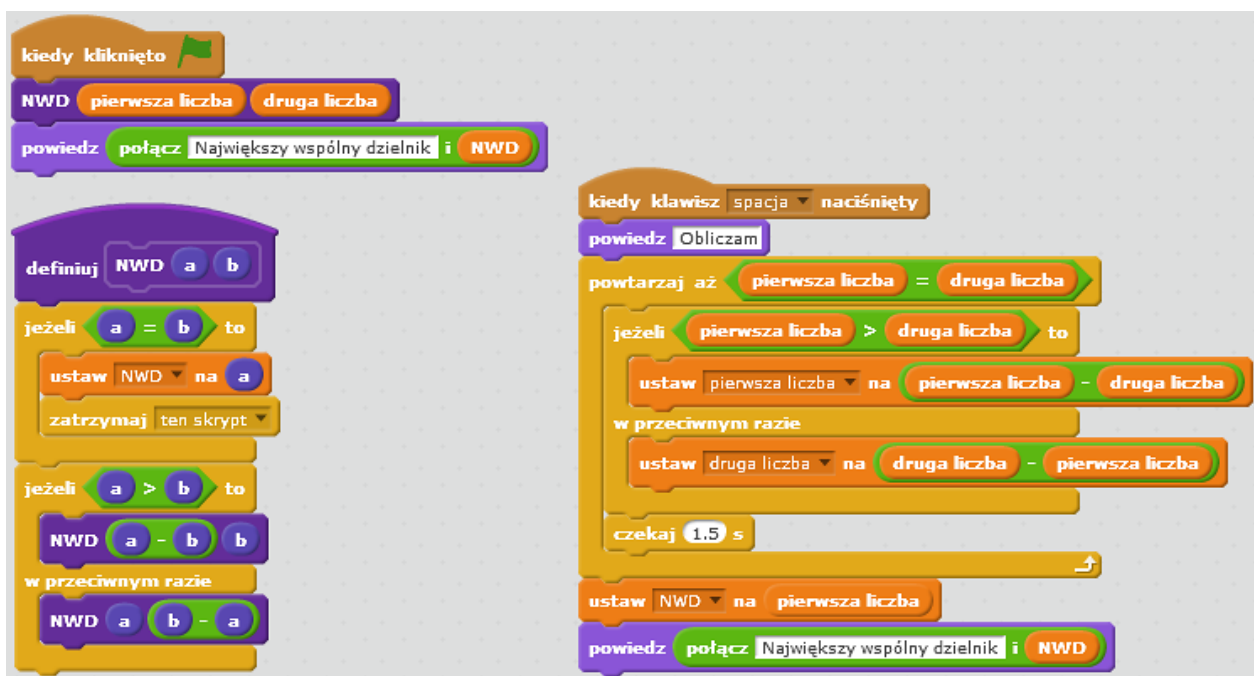


**Rys. 1.** Schemat blokowy algorytmu Euklidesa zapisany w programie ELI

## ZAPIS ALGORYTMU W SCRATCHU

Kolejnym sposobem zapisu algorytmu jest skrypt zbudowany w Scratchu. On także stanowi wizualizację algorytmu, którą można uruchomić. Na rysunku 2. są przedstawione dwie przykładowe realizacje algorytmu Euklidesa w Scratchu. Obie obliczają największy wspólny dzielnik. Pierwsza wykorzystuje rekurencję i zmienne lokalne, czyli parametry bloku **NWD** (niebieskie), a druga zawiera pętlę **powtarzaj ... aż** i działa na zmiennych globalnych, czyli zdefiniowanych za pomocą przycisku **Stwórz zmienną** w kategorii **Dane** (czerwone).

■ Jeśli chcesz uruchomić ten projekt, otwórz stronę: [scratch.mit.edu/projects/37501646](https://scratch.mit.edu/projects/37501646).



Rys. 2. Realizacje algorytmu Euklidesa w Scratchu

## ZAPIS ALGORYTMU W JĘZYKU PROGRAMOWANIA

Innym sposobem zapisu algorytmu jest program napisany w wybranym języku programowania. Pierwotnie komputery programowano w kodzie maszynowym, czyli za pomocą zestawu elementarnych poleceń zrozumiałych dla procesora. Było to zajęcie żmudne i trudne, w dodatku dla każdej kolejnej wersji procesora ten zestaw był nieco inny. W latach 50. XX wieku powstały pierwsze języki programowania wyższego poziomu (wyższego niż poziom poleceń procesora), o charakterze uniwersalnym, czyli te same dla różnych komputerów. Początkowo były to języki Fortran i Algol, później Cobol (czytaj: kobil), Pascal (czytaj: paskal), Basic (czytaj: bejzik) i Logo. Kolejna generacja języków to Java (czytaj: dżawa), C, Python (czytaj: pajton), Visual Basic (czytaj: wizual bejzik), PHP.

Poniżej przedstawiono dwie przykładowe realizacje algorytmu Euklidesa w języku Python i JavaScript. Zastosowano w nich pętlę **while warunek instrukcje**, co oznacza: dopóki jest spełniony warunek, wykonuj instrukcje. Zauważ, że warunkiem wykonywania odejmowania w pętli jest to, że liczby nie są sobie równe.

### Znaczenie elementów języka użytych w poniższych programach

**if** (czytaj: yf; jeżeli) – działa tak, jak blok **jeżeli ... to ...** w Scratchu.

**else** (czytaj: els; w przeciwnym razie) – dodane do **if** odpowiada blokowi **jeżeli ... to ... w przeciwnym razie ...** w Scratchu.

**print** (czytaj: prynt; wypisz) – odpowiada blokowi **powiedz ...** w Scratchu.

**var** (czytaj: wer; zmienna) – odpowiada przyciskowi **Stwórz zmienną** w Scratchu.

**return** (czytaj: riterm; zwróć) – przerywa działanie funkcji i zwraca podaną wartość; takiego bloku nie ma w Scratchu.

**fill** (czytaj: fyl; ustal kolor wypełnienia i kolor tekstu) – takiego bloku nie ma w Scratchu.

**text** (czytaj: tekst; wypisz tekst) – odpowiada blokowi **powiedz ...** w Scratchu.

Przyjrzyj się zapisom obu programów. Który wydaje ci się bardziej czytelny?

Python (środowisko PyScripter)	JavaScript (środowisko ProcessingJS, np. Khan Academy CS)
<pre> a=152 b=57 while a!=b:     if a&gt;b:         a=a-b     else:         b=b-a print(a)</pre>	<pre> var a=152; var b=57; var nwd = function(a,b){     while (a!=b){         if (a&gt;b){a=a-b;} else {b=b-a;}     }     return a; }; fill(0, 0, 0); text(nwd(a,b), 135, 100);</pre>

## ZADANIA

1. Istnieje wersja algorytmu Euklidesa szybsza i efektywniejsza od poznanej na lekcji. Polega ona na sprawdzaniu reszty z dzielenia większej liczby przez mniejszą zamiast odejmowania. Gdy reszta ta osiągnie zero, należy skończyć działanie w pętli i jako wynik wziąć drugą liczbę.

Zbuduj w Scratchu projekt realizujący szybszą wersję algorytmu Euklidesa. Do rozwiązania będzie potrzebny ci blok **mod** – zwracający resztę z dzielenia. Znajdziesz go w grupie **Wyrażenia**.