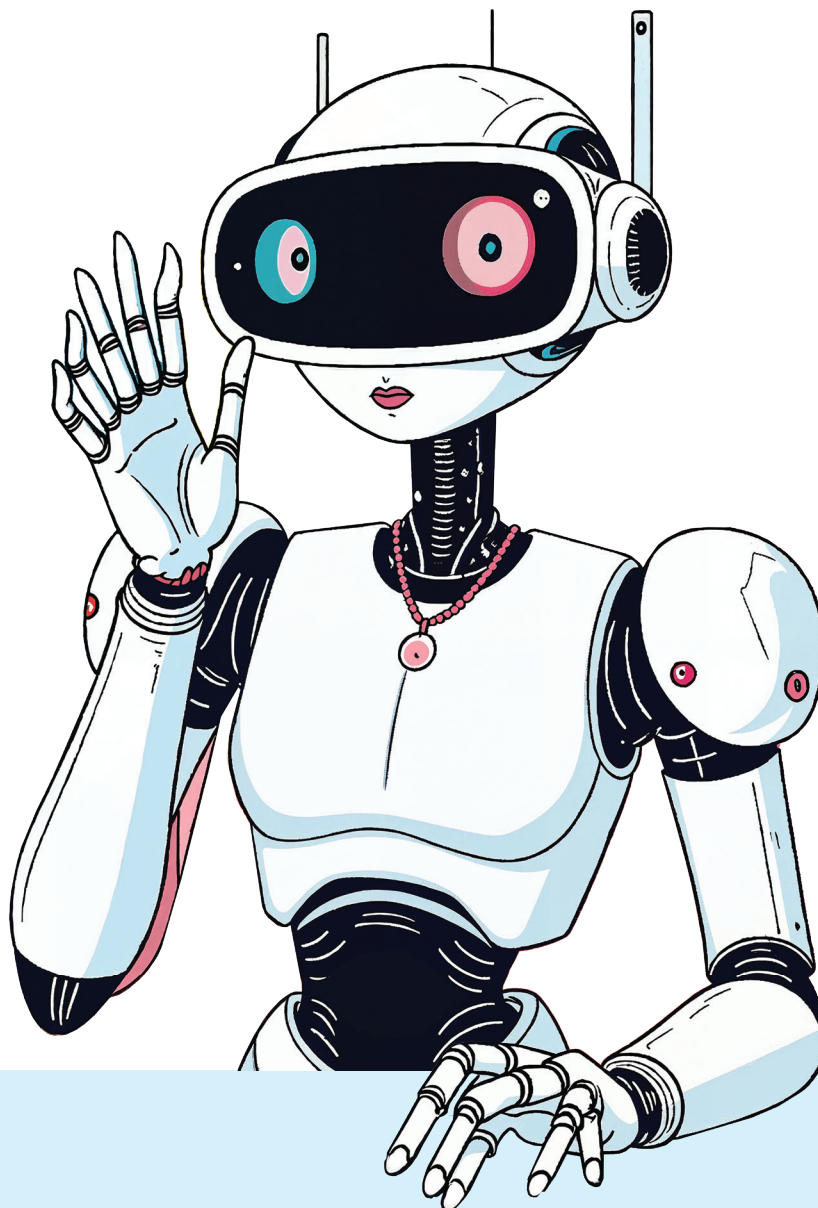




Lekcja
o Funduszach
Europejskich IX



Robotyka

- świat pełen możliwości!



Fundusze Europejskie
dla Polski Wschodniej



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Programowanie - język robotów

Robot nie rozumie naszego języka, dlatego musimy mówić do niego w specjalnych językach programowania. Dzięki nim człowiek może wydawać maszynie polecenia. Oto przykłady takich języków:

Scratch

Wizualny język programowania, idealny dla początkujących, w którym programowanie odbywa się za pomocą kolorowych bloczków.

C++

Język programowania zapewniający bardzo szybkie działanie programów i efektywne wykorzystanie pamięci; stosowany w zaawansowanych robotach i urządzeniach technicznych, np. w sterowaniu łazikami marsjańskimi.

Python

Prosty i czytelny język tekstowy, bardzo popularny wśród programistów. Używany w robotyce, grach oraz mający zastosowanie w sztucznej inteligencji.

Trening 1. Logiczne myślenie!

Budowa i programowanie robotów wymagają solidnych podstaw logicznego myślenia. Sprawdź się!

Zeskanuj kod QR lub wejdź na
<https://www.umiemyinformatyke.pl/decydowanie-wspolna-cecha-emoji-2>



Zadanie 1. Ciąg liczbowy.

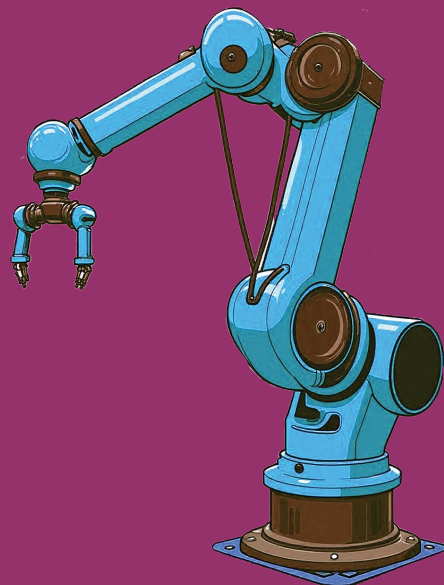
Wybierz z ramki obok brakujące liczby do ciągu liczbowego.

1	2		5	8	13	21	
---	---	--	---	---	----	----	--

42	3	4	34
----	---	---	----

Ciekawostka

Robotyka to dziedzina nauki i techniki zajmująca się projektowaniem, budowaniem i programowaniem robotów. Roboty mogą wykonywać zadania, które są dla ludzi zbyt trudne, niebezpieczne lub po prostu nudne.



Czym jest algorytm?

Czy wiesz, że każdego dnia korzystasz z algorytmów? Brzmi poważnie, a to nic trudnego! Algorytm to po prostu dokładny plan działania krok po kroku.

Wyobraź sobie, że robisz kanapkę z serem:



1. Kroisz chleb.



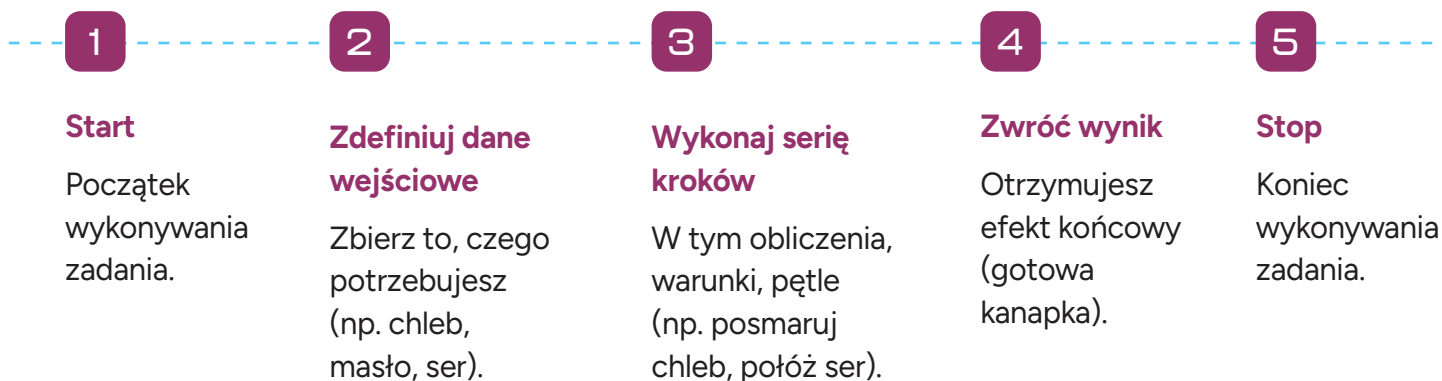
2. Smarujesz kromkę masłem.



3. Kładasz plaster sera.

Jeśli pomylisz kolejność (np. najpierw ser, a potem masło), całość raczej się nie uda. **Roboty działają dokładnie tak samo – potrzebują algorytmów, żeby wiedzieć, co zrobić najpierw, a co potem. Bez nich byłyby kompletnie zagubione.**

Oto uniwersalny wzór działania każdego algorytmu:



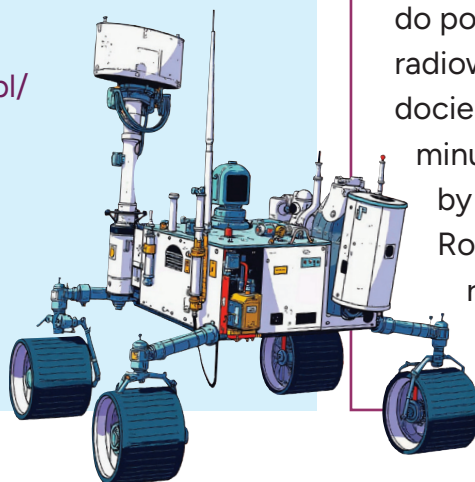
Opisowy przykład algorytmu obliczania pola prostokąta:

1. Wprowadź długość a.
2. Wprowadź szerokość b.
3. Oblicz pole: $P = a \times b$.
4. Wyświetl

Trening 2. Myślenie algorytmiczne.

Im lepiej potrafisz układać algorytmy, tym łatwiej zaprogramujesz swojego robota. Śledź drogę myszy i wskaż, do którego przedmiotu dotrze.

Zeskanuj kod QR lub wejdź na <https://www.umiemyinformatyke.pl/przesuwanie-ciag-mysz-aktywna>



Ciekawostka

Roboty pracujące na Marsie mają specjalne algorytmy do poruszania się, gdyż sygnał radiowy z Ziemi na Marsa dociera nawet dopiero po 20 minutach. To zbyt długo, by czekać na komendę. Robot musi więc umieć radzić sobie sam!

Zadanie 2. Własny algorytm.

Stwórz algorytm opisujący, co robisz rano. Wpisz poniżej kroki algorytmu w odpowiedniej kolejności:

- 1 _____
- 2 _____
- 3 _____
- 4 _____
- 5 _____

Trening 3. Myślenie algorytmiczne.

Pomóż skrzatowi dotrzeć do drzwi, omijając drzewa. Przejdź jak najwięcej plansz w wyznaczonym czasie.

Zeskanuj kod QR lub wejdź na
<https://www.umiemyinformatyke.pl/strzalki-rozgrzewka/2>



Co to jest zmienna?

Zmienna w programowaniu to coś w rodzaju pamięci robota – pozwala mu zapamiętać ważne informacje, np. wynik w grze czy odległość do przeszkody. Możesz wyobrazić ją sobie jako kieszeń w plecaku robota. Do tej kieszeni wkładasz coś, co robot ma zapamiętać, np. liczbę punktów w grze. Kieszeń ma swoją nazwę (np. „punkty”), żeby robot wiedział, co tam jest. W trakcie gry można zmieniać zawartość kieszeni — dodawać punkty albo je zabierać.

Trening 4. Opanuj zmienne!

Sprawdź, czy rozumiesz działanie zmiennych. Zeskanuj kod QR lub wejdź na:
<https://www.umiemyinformatyke.pl/memory-zmienne-glowne-zasady-2>



Przykład rozwiązania z wyjaśnieniem:

$x \leftarrow 2$ Do zmiennej x wkładamy liczbę 2.
 $y \leftarrow 3$ Do zmiennej y wkładamy liczbę 3.
 $x \leftarrow y - x$ Teraz zmieniamy wartość w x .

Liczymy $y - x \rightarrow$ czyli $3 - 2 = 1$.

Do zmiennej x wpisujemy wynik, więc teraz $x = 1$.

Podsumowanie wartości po tych krokach:

$x = 1$

$y = 3$

Jak zaprogramować robota?

Zanim napiszesz prawdziwy program w komputerze, dobrze jest najpierw zaplanować, co ma zrobić robot – krok po kroku. Zamiast od razu pisać kod, możesz użyć czegoś prostszego – tzw. pseudokodu.

Co to jest pseudokod?

To sposób zapisywania pomysłów na program w prosty, zrozumiały sposób. Nie jest to prawdziwy kod, choć wygląda trochę podobnie do niego. Nie musisz znać języka programowania, żeby zrozumieć pseudokod. Pseudokod pomaga:

- ✓ w prosty sposób zrozumieć logikę działania programu (co się dzieje po kolei);
- ✓ stworzyć „plan działania” – przygotować się do pisania prawdziwego kodu;
- ✓ wytłumaczyć innym osobom, niemającym wiedzy programistycznej, co robi robot.

Przykład w pseudokodzie: liczenie kroków robota

START

← Tak oznaczamy początek programu.

ZADEKLARUJ kroki jako liczbę

← Tworzymy zmienną „kieszeń” o nazwie „kroki” do przechowywania informacji o liczbie zrobionych kroków.

USTAW kroki ← 0

← Wkładamy do „kieszeni” liczbę 0, ponieważ robot jeszcze nie zrobił kroku i dopiero zaczyna się poruszać.

POWTARZAJ 4 RAZY

← Wskazujemy robotowi, żeby zrobił to, co poniżej, cztery razy; to wygodny sposób na uniknięcie wydawania czterech osobnych poleceń.

IDŹ do przodu

← Dajemy robotowi komendę, aby zrobił jeden krok do przodu.

ZWIĘKSZ kroki o 1

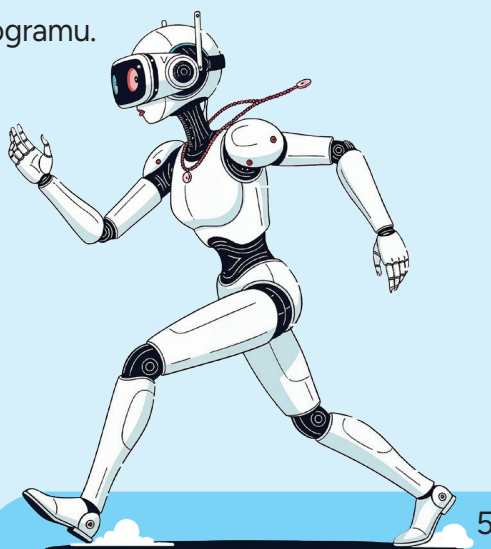
← Dodajemy 1 do liczby w „kieszeni”, żeby robot pamiętał, że zrobił kolejny krok.

STOP

← Oznaczamy koniec programu.

Ciekawostka

Twoja pamięć krótkotrwała może przechowywać około siedmiu informacji jednocześnie. Robot może natomiast zapamiętać tysiące informacji – pod warunkiem, że mu na to pozwolisz, tworząc odpowiednią liczbę zmiennych!



Czym jest warunek?

Warunek to zasada typu „jeśli coś się stanie, to zrób to”. **Jeśli pada deszcz, to biorę parasol**. Roboty działają podobnie! Dzięki warunkom mogą podejmować decyzje – ale tylko takie, które wcześniej zaprogramował człowiek.

Schemat ogólny warunków:

- Jeśli (ang. *if*) coś jest prawdą, ➡ zrób **A** (określone zadanie).
- W przeciwnym razie (ang. *else*), ➡ zrób **B** (inne zadanie).

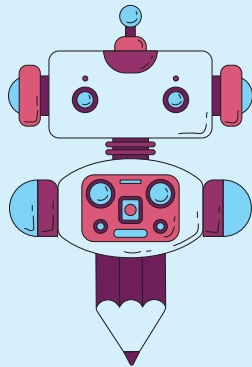
Przykład:

Jeśli pada deszcz, ➡ weź parasol.
W przeciwnym razie ➡ idź bez parasola.

Zadanie 3. Naucz robota podejmować decyzje.

Twój robot jeździ po pokoju. Przed nim znajduje się przeszkoda. Pomóż mu podjąć decyzję:

Napisz, co powinien zrobić robot, jeżeli zobaczy przeszkodę przed sobą:
Jeżeli robot widzi przeszkodę, to:



A co powinien zrobić, jeśli nie widzi przeszkody? **Jeżeli robot nie widzi przeszkody, to**

Pętle - roboty lubią powtarzać

Pętla to bardzo sprytny sposób, żeby nakazać robotowi powtarzanie jakiejś czynności wiele razy, bez konieczności ciągłego powielania tej samej komendy. Pomyśl, jakby wyglądało Twoje życie bez pętli:

umyj pierwszy ząb, umyj drugi ząb, umyj trzeci ząb...

Na szczęście, zamiast pisać osobną instrukcję dla każdego zęba, możesz użyć pętli: **Powtarzaj 32 razy:** „umyj ząb”.

Przykład pętli w pseudokodzie: robot rysuje kwadrat	Wyjaśnienie:
POWTARZAJ 4 RAZY:	➔ Robot powtórzy te same czynności 4 razy.
IDŹ do przodu 10 kroków	➔ Każdy bok kwadratu ma 10 kroków.
SKRĘĆ w prawo o 90 stopni	➔ Po każdym boku robot skręca, żeby narysować następny bok.
STOP	➔ Koniec rysowania.

Trening 5. Pętle i warunki.
Warunki i pętle to dwa uzupełniające się zagadnienia. Zeskanuj kod QR lub wejdź na <https://www.umiemyinformatyke.pl/memory-zolw-powtorzenia-2>



Funkcje – jak uprościć pracę robotowi

Funkcja ułatwia programowanie – zamiast powtarzać te same kroki wiele razy, zapisujesz je raz i potem tylko wywołujesz.

Przykład:

Zamiast za każdym razem pisać: „odkręć wodę, weź mydło, umyj dłonie, spłucz, zakręć wodę, wytrzyj ręce”, stworzysz funkcję mieszczącą wszystkie potrzebne kroki: „umyj ręce” i używasz jej, kiedy trzeba.

Przykład funkcji

Definiowanie funkcji	Wyjaśnienie
START	→ Tu rozpoczyna się definicja funkcji.
FUNKCJA ObliczPole(dlugosc, szerokosc)	→ Tworzymy funkcję o nazwie ObliczPole, która potrzebuje dwóch danych: długości i szerokości.
pole ← dlugosc * szerokosc	→ Robot mnoży te dwie wartości i zapisuje wynik w zmiennej pole.
ZWRÓĆ pole	→ Funkcja oddaje wynik, żeby można go było później wykorzystać.
KONIEC FUNKCJI	→ Tu kończy się definicja funkcji.
Wykorzystanie funkcji	Wyjaśnienie
START	→ Tu rozpoczyna się wykorzystanie funkcji.
Wczytaj a, b	→ Program prosi użytkownika o podanie długości i szerokości prostokąta.
wynik ← ObliczPole(a, b)	→ Tutaj program wywołuje naszą funkcję i liczy pole na podstawie danych a i b.
Wypisz „Pole prostokąta: ” wynik	→ Pokazuje wynik na ekranie.
STOP	→ Tu kończy się wykorzystanie funkcji.

Przykład funkcji w praktyce

Użytkownik wpisuje:

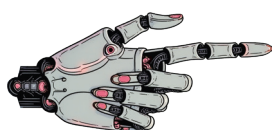
- a = 5
- b = 3

→ Funkcja liczy: $5 \times 3 = 15$.

→ Program wypisuje: Pole prostokąta: 15.

Zadanie 4. Napisz własną funkcję.

Twój robot sprząta pokój. Chcesz, żeby co jakiś czas wykonywał zestaw czynności: podnosił zabawkę, odkładał ją na półkę i odkurzał miejsce. Opisz powyższą funkcję w pseudokodzie.



Trening 6. Uogólnianie funkcji.

Funkcje są tym lepsze, im bardziej uniwersalne. Dobrze zaprojektowana funkcja sprawia, że robot działa szybciej i sprawniej. Zeskanuj kod QR lub wejdź na <https://www.umiemyinformatyke.pl/przesuwanie-domino-abstrakcja-2/1321>



START

POWTARZAJ 3 RAZY

Wczytaj liczba_krokov

Jeśli liczba_krokov > 5

Wypisz „duży krok”

W przeciwnym razie

Wypisz „mały krok”

KONIEC POWTARZAJ

STOP

Zadanie 5. Analiza programu

Wyobraź sobie robota, któremu podajesz liczbę kroków, a on odpowiada czy to krok duży, czy mały. Odczytaj program i sprawdź, jak robot podejmuje decyzje – a potem uzupełnij zdania.

Uzupełnij zdania:

1. Program zaczyna się od słowa
2. Następnie włącza się pętla, która powtarza się razy.
3. Za każdym razem użytkownik wpisuje kroków.
4. Program sprawdza, czy liczba kroków jest większa niż
5. Jeśli liczba kroków jest większa niż 5, stawia
6. W przeciwnym razie stawia
7. Program kończy się poleceniem

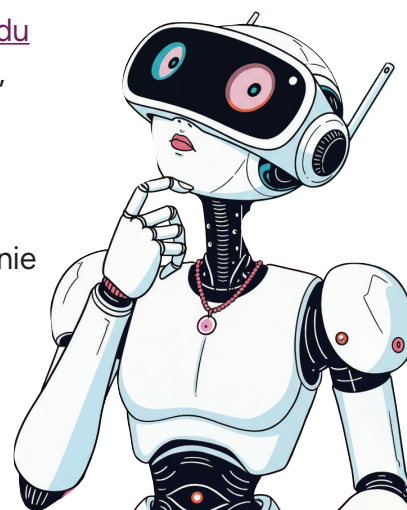
Trening 7. Czas na test.

Dowiedz się jeszcze więcej o programowaniu i robotyce. Zeskanuj kod QR lub wejdź na <https://www.umiemyinformatyke.pl/decydowanie-robotyka-2>



Możesz dalej rozwijać swoje umiejętności, korzystając z poniższych linków:

- ➔ www.umiemyto.pl
– rozwiązuj zadania i kursy z programowania;
- ➔ www.scratch.mit.edu
– twórz własne gry, animacje i roboty w Scratchu;
- ➔ www.replit.com
– pisz kod w Pythonie i innych językach bez instalowania programów.



Organizatorem akcji „Lekcja o Funduszach Europejskich IX” jest Ministerstwo Funduszy i Polityki Regionalnej – Departament Programów Ponadregionalnych.

Więcej informacji: www.fepw.gov.pl/lekcja

Publikacja współfinansowana ze środków Unii Europejskiej.



Fundusze Europejskie
dla Polski Wschodniej



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską

