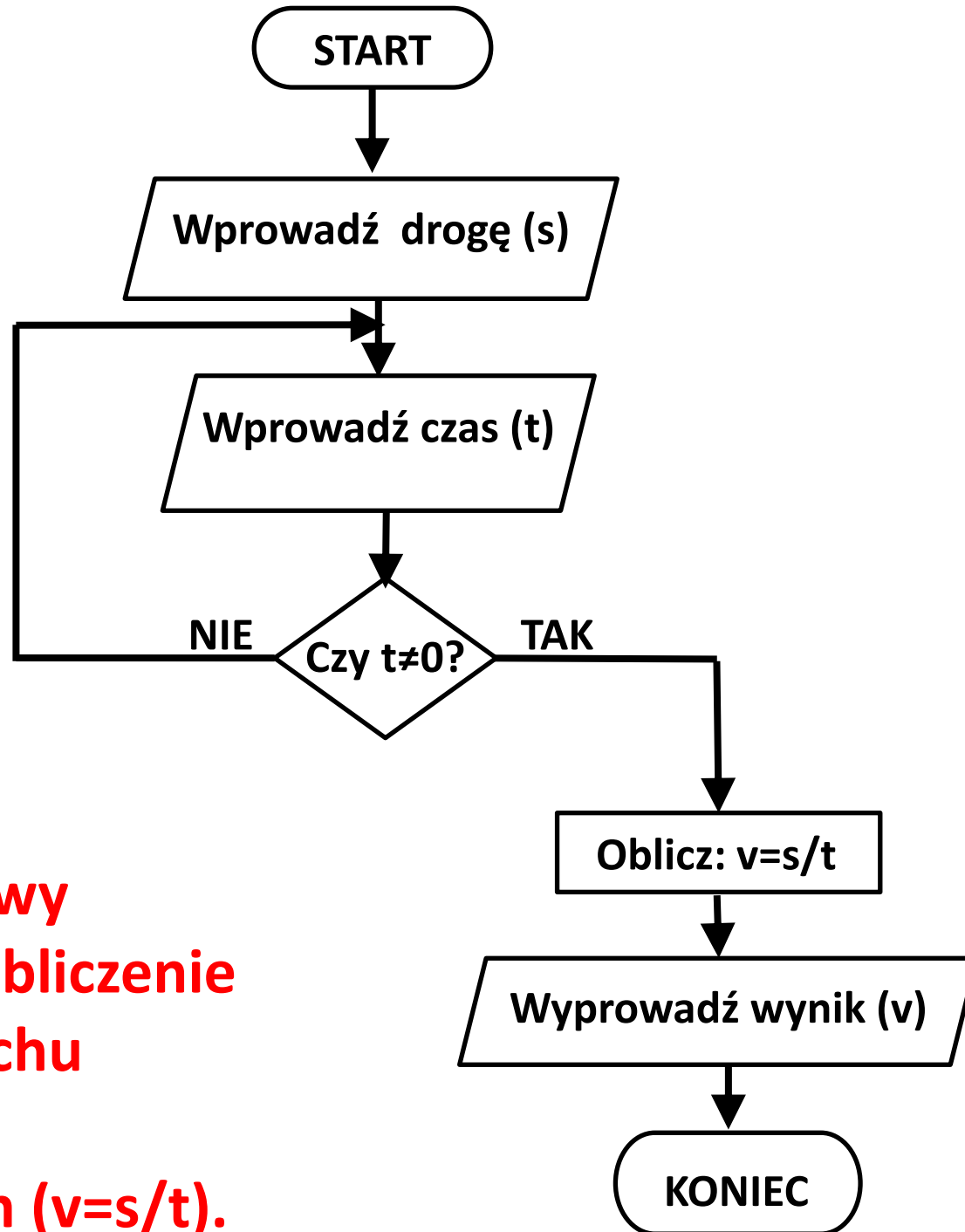


**Algorytm na obliczenie prędkości  
w ruchu jednostajnym  
prostoliniowym.**



**Schemat blokowy  
algorytmu na obliczenie  
prędkości w ruchu  
jednostajnym  
prostoliniowym ( $v=s/t$ ).**

# Blok wejścia, wprowadzanie danych przez użytkownika.

JavaBlock

Plik Edycja Schematy Ustawienia Pomoc

Interpreter: JavaScript 200 Ukryj konsolę

Edytor bloku Wejścia/Wyjścia

Zmienna: s

Wiadomość: Podaj drogę

☒ Rysuj standardowy kształt

☒ Wczytaj ☐ Wypisz

**Wczytaj**

☐ Tablica: 0 elementów

Typ: Liczba

☒ Rysuj wiadomość

Wstaw nowy blok tego typu

Start

wypisz: Podaj drogę  
wczytaj s

# Blok wejścia, wprowadzanie danych przez użytkownika.

JavaBlock

Plik Edycja Schematy Ustawienia Pomoc

Interpreter: JavaScript 200 Ukryj konsolę

Edytor bloku Wejścia/Wyjścia

Zmienna: t

Wiadomość: Podaj czas

☒ Rysuj standardowy kształt

☒ Wczytaj ☐ Wypisz

Wczytaj

☐ Tablica: 0 elementów

Typ: Liczba

☒ Rysuj wiadomość

Wstaw nowy blok tego typu

Start

wypisz: Podaj drogę  
wczytaj s

wypisz: Podaj czas  
wczytaj t

```
graph TD; Start((Start)) --> Output[/wypisz: Podaj drogę  
wczytaj s/]; Output --> Input[/wypisz: Podaj czas  
wczytaj t/];
```

# Blok decyzyjny (warunkowy). Wpisz kod: $t \neq 0$ (tzn. $t \neq 0$ ).

JavaBlock

Plik Edycja Schematy Ustawienia Pomoc

Interpreter: JavaScript 200 Ukryj konsolę

Kod: Komentarz

$t \neq 0$

Wstaw nowy blok tego typu

Start

wypisz: Podaj drogę  
wczytaj s

wypisz: Podaj czas  
wczytaj t

$t \neq 0$

The screenshot shows the JavaBlock IDE interface. On the left is a toolbar with various block icons. The main workspace contains a flowchart with the following steps: a 'Start' terminal block, a process block with 'wypisz: Podaj drogę' and 'wczytaj s', another process block with 'wypisz: Podaj czas' and 'wczytaj t', and a decision block with the condition ' $t \neq 0$ '. A red arrow originates from the title bar text and points to the decision block. The code editor on the left shows the code ' $t \neq 0$ '.

# Blok przetwarzania, wykonywanie obliczeń.

JavaBlock

Plik Edycja Schematy Ustawienia Pomoc

Interpreter: JavaScript 200 Ukryj konsolę

Kod: Komentarz

$v = s/t$

Wstaw nowy blok tego typu

Start

wypisz: Podaj drogę  
wczytaj s

wypisz: Podaj czas  
wczytaj t

$t \neq 0$

$v \leftarrow s/t$

```
graph TD; Start([Start]) --> IO1[/wypisz: Podaj drogę  
wczytaj s/]; IO1 --> IO2[/wypisz: Podaj czas  
wczytaj t/]; IO2 --> Decision{t ≠ 0}; Decision --> End([End]);
```

$v \leftarrow s/t$

# Wstawianie połączeń, linii łamanych („przeskok”).

JavaBlock

Plik Edycja Schematy Ustawienia Pomoc

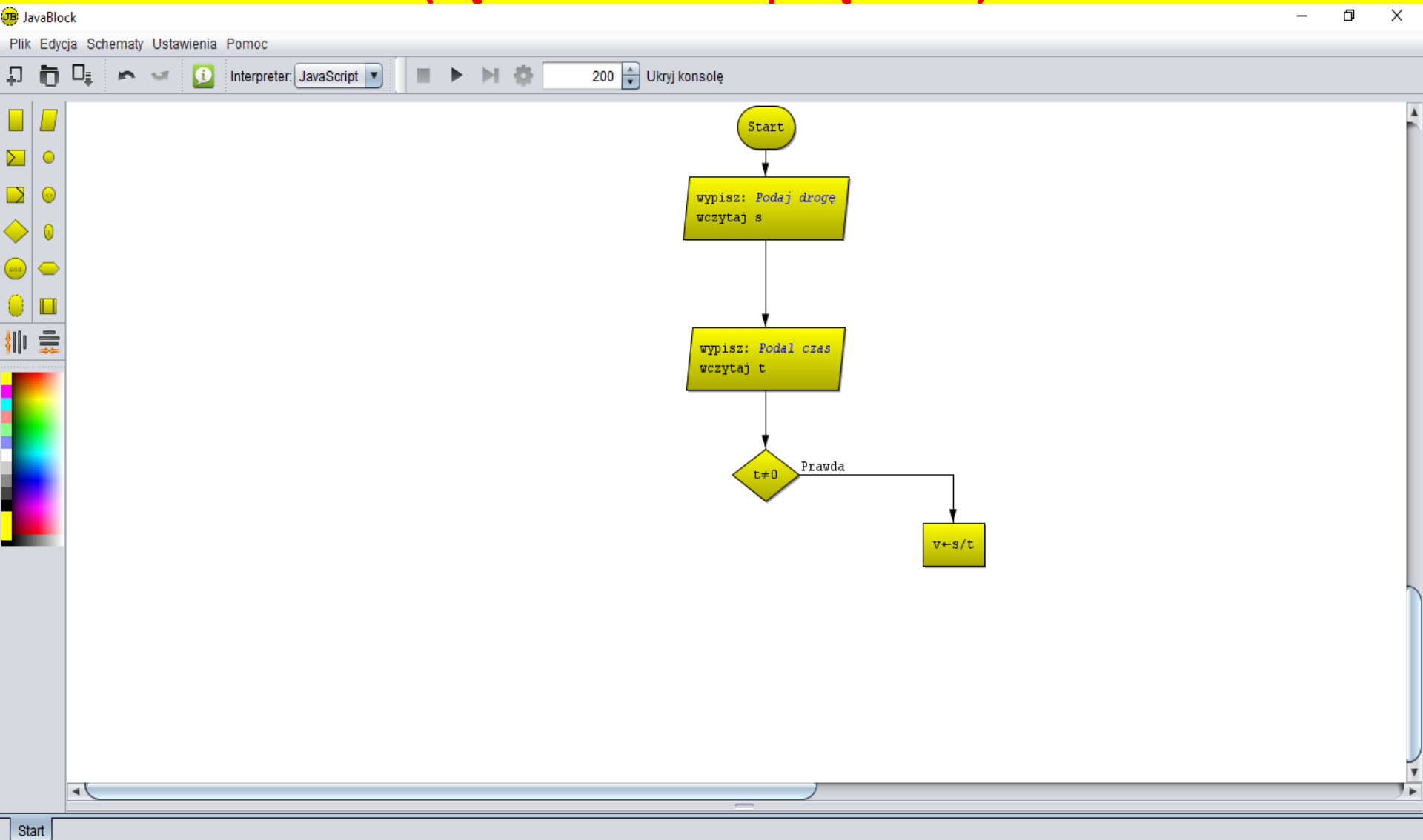
Interpreter: JavaScript 200 Ukryj konsolę

```
graph TD; Start([Start]) --> IO1[/wypisz: Podaj drogę  
wczytaj s/]; IO1 --> IO2[/wypisz: Podaj czas  
wczytaj t/]; IO2 --> Cond{t ≠ 0}; Cond --> Calc[v ← s/t]; Calc --> IO1;
```

The flowchart illustrates a loop for calculating average speed. It begins with a 'Start' terminal, followed by an input/output block for distance 's'. Then, it asks for time 't'. A decision diamond checks if 't' is not equal to 0. If true, it calculates the average speed  $v \leftarrow s/t$  and loops back to the input/output block. A red arrow points to the connector icon in the left toolbar, which is used to create the loop connection.

Start

**Wstawianie połączeń, linii łamanych („przeskok”): kliknij na bloku decyzyjnym (będzie to początek połączenia), przyciśnij lewy przycisk „ctrl” i kliknij na „przeskoku”, a następnie na bloku przetwarzania (będzie to koniec połączenia).**





# Blok wyjścia, wyprowadzenie wyników.

JavaBlock

Plik Edycja Schematy Ustawienia Pomoc

Interpreter: JavaScript 200 Ukryj konsolę

Edytor bloku Wejścia/Wyjścia

Zmienna: v

Wiadomość: dkość wynosi:

☒ Rysuj standardowy kształt

☐ Wczytaj ☒ Wypisz

Sufix wiadomości

☒ Wstaw nową linię

Wstaw nowy blok tego typu

Start

wypisz: Podaj drogę  
wczytaj s

wypisz: Podaj czas  
wczytaj t

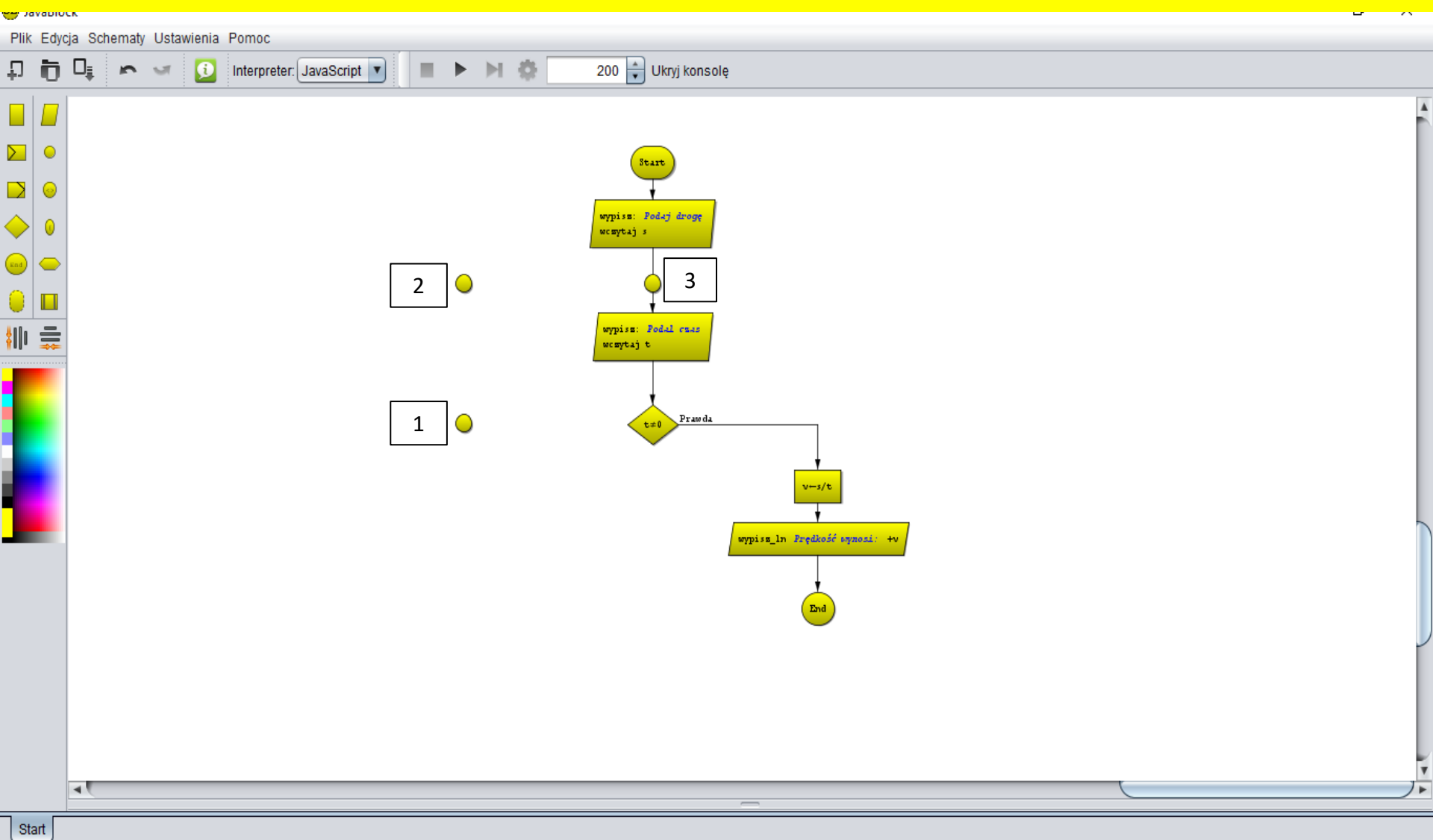
t ≠ 0 Prawda

$v \leftarrow s/t$

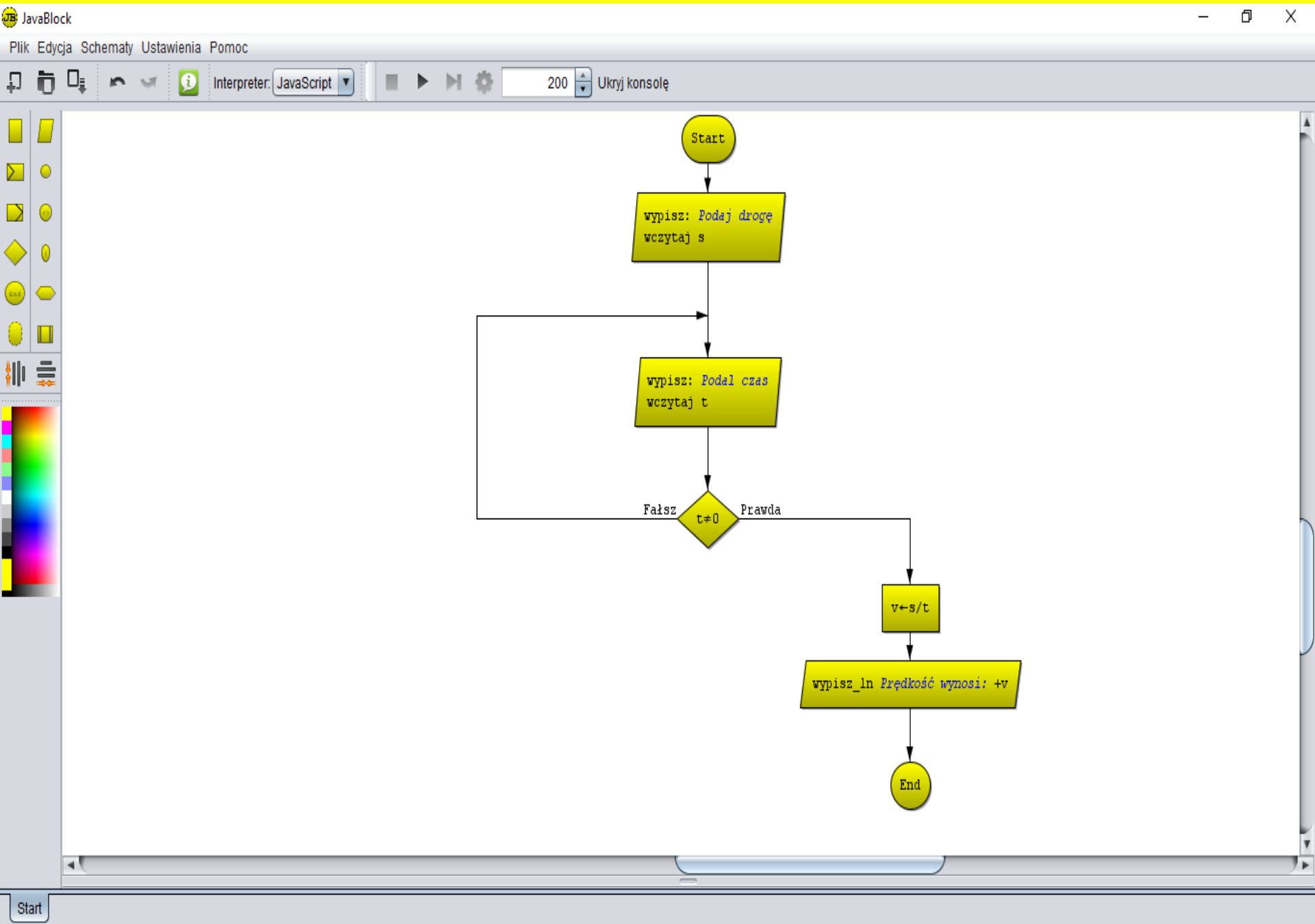
wypisz\_ln Prędkość wynosi: +v

End

**Wstawianie połączeń, linii łamanych („przeskok”): kliknij na bloku decyzyjnym (będzie to początek połączenia), przyciśnij lewy przycisk „ctrl” i kliknij na „przeskoku 1”, ustaw wskaźnik myszy i kliknij w miejscu „przeskoku 2 i „przeskoku 3” – pojawią się połączenia.**



# Gotowy algorytm.



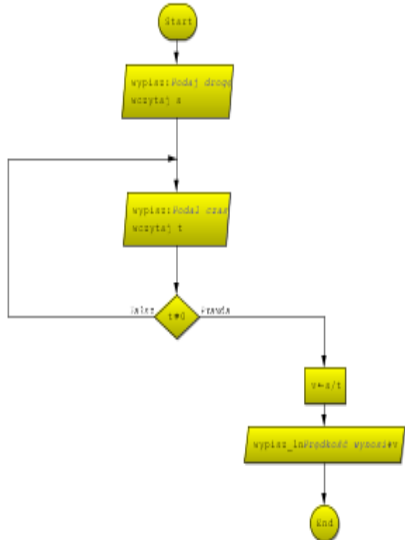
**Aby zobaczyć czy algorytm jest poprawnie zrobiony lub jaki jest wynik działania matematycznego kliknij w polecenie „Ukryj konsolę”.**

JavaBlock

Plik Edycja Schematy Ustawienia Pomoc

Interpreter: JavaScript

200 Ukryj konsolę



```
graph TD; Start((Start)) --> ReadA[/wczytaj a/]; ReadA --> ReadT[/wczytaj t/]; ReadT --> Decision{t <= a}; Decision -- tak --> ReadT; Decision -- nie --> Calc[1 + a / t]; Calc --> Print[/wypisz_indeks_i_wynosi/]; Print --> End((End));
```

Wyjście Zmienne Wejście

**Aby przetestować algorytm jest poprawnie kliknij w przycisk polecenie „Uruchom”, a następnie testowanie „Krokowe” lub „Automatyczne”.**

JavaBlock

Plik Edycja Schematy Ustawienia Pomoc

Interpreter: JavaScript

200 Ukryj konsolę

```
graph TD; Start([Start]) --> Wypisz1[/Wypisz: Prędkość wynosi: 1/]; Wypisz1 --> Wczytaj1[/Wczytaj: a/]; Wczytaj1 --> Wypisz2[/Wypisz: Prędkość wynosi: a/]; Wypisz2 --> Dec1{a < 100}; Dec1 -- Nie --> Wczytaj1; Dec1 -- Tak --> Dziel[/a = a / 2/]; Dziel --> Wypisz3[/Wypisz: Prędkość wynosi: a/]; Wypisz3 --> Koniec([KONIEC]);
```

Wyjście Zmienne Wejście

Prędkość wynosi:1

==KONIEC==

Wyczyść

Start