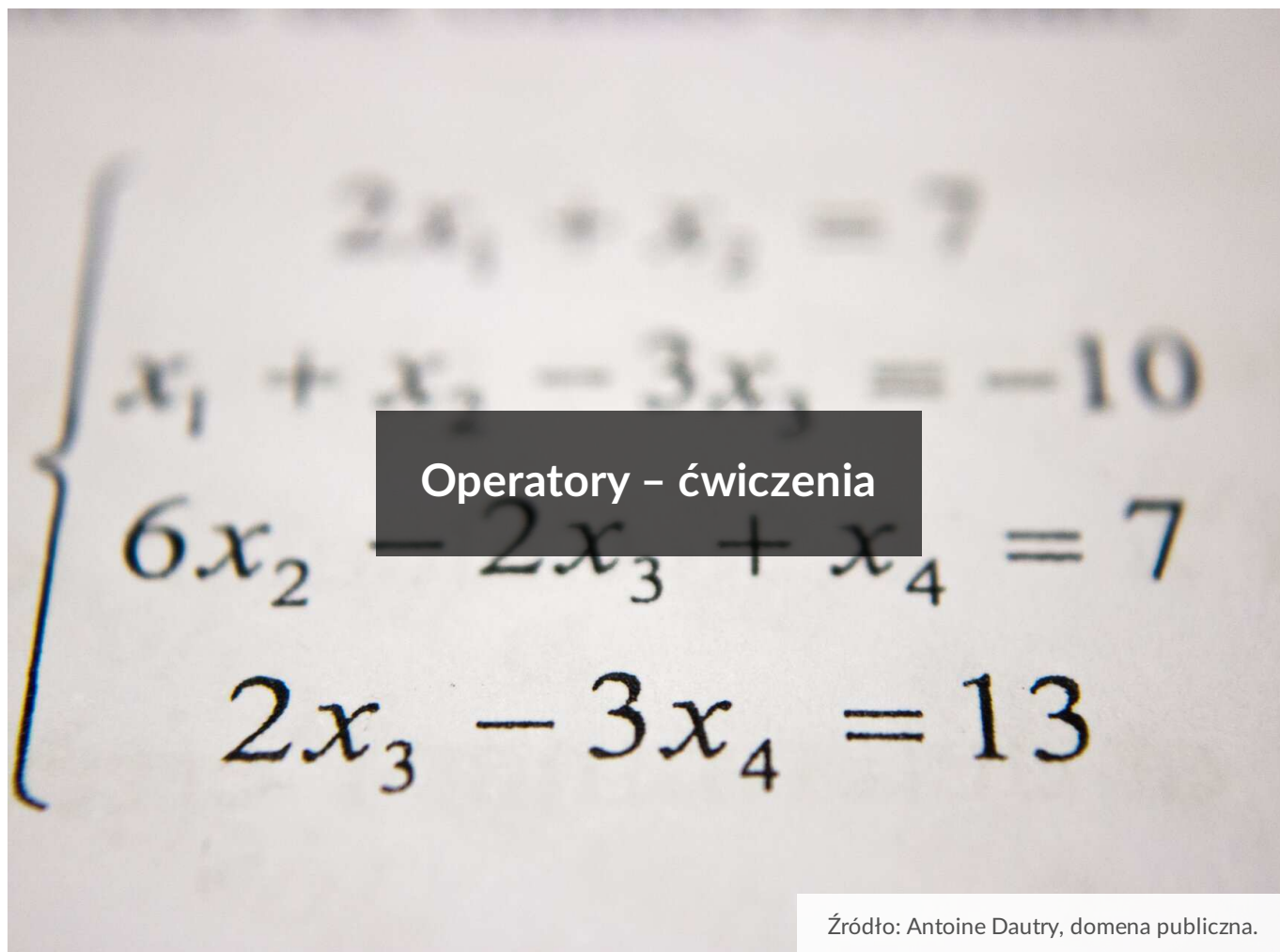


$$\begin{cases} 2x_1 + x_2 = 7 \\ x_1 + x_2 - 3x_3 = -10 \\ 6x_2 - 2x_3 + x_4 = 7 \\ 2x_3 - 3x_4 = 13 \end{cases}$$

Operatory – ćwiczenia

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Schemat interaktywny](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Poznaliśmy już różne operatory, których możemy używać w wybranych językach programowania.

Podstawowe informacje na ich temat znajdziesz w e-materiale [Operatory](#).

W tym e-materiale zapoznamy się z przykładowymi zadaniami dotyczącymi tego zagadnienia.

Operatory w poszczególnych językach programowania omówione są w e-materiałach:

- [Operatory w języku C++](#),
- [Operatory w języku Java](#),
- [Operatory w języku Python](#).

Twoje cele

- Przeanalizujesz sposób zapisywania i wykonywania działań w programach.
- Scharakteryzujesz sposób wykonywania operacji arytmetycznych oraz logicznych w zadaniach.
- Zapiszesz działania arytmetyczne oraz wyrażenia logiczne.

Przeczytaj

Przykładowe zadanie

Zadanie 1

Treść zadania:

Sprawdź, jaki będzie rezultat zastosowania przedstawionego algorytmu.

```
1 a = 2
2 b = 2
3 c = a + b * a
4 wypisz c
5 wypisz c++
6 a = 0
7 d = a * (21 * b + c * 3) + b
8 wypisz d
9 b = 14 % 3
10 wypisz b
```

Rozwiązanie:

Przeanalizujemy wykonywane czynności.

1. Zmiennej `a` przypisujemy wartość 2.
2. Zmienna `b` również przyjmuje wartość 2.
3. Zmiennej `c` przypisujemy wynik działania $2 + 2 * 2$. Pamiętajmy o kolejności wykonywania operacji matematycznych (mnożenie ma wyższy priorytet niż dodawanie). Zmienna `c` ma zatem wartość 6.
4. Wypisujemy 6.
5. W linii nr 5 znalazły się dwa polecenia: wypisania wartości zmiennej `c` oraz wykonania postinkrementacji (czyli zwiększenia `c` o 1). Ponownie wypisujemy wartość 6. Dzieje się tak dlatego, że w wyniku postinkrementacji wartość zmiennej zostaje zwiększona na końcu. Polecenie zapisane w linii 5. brzmi:
„wypisz wartość `c`, a następnie zwiększ ją o 1”. Gdybyśmy mieli do czynienia z preinkrementacją (czyli poleceniem `++c`), wartość `c` zostałaby powiększona o 1 przed jej wyświetleniem (pojawiłby się napis 7). W obydwu przypadkach po wykonaniu polecenia z linii numer 5 zmienna `c` wynosi 7.

6. Zmienna a przyjmuje wartość 0.
7. Zmiennej d przypisywany jest wynik skomplikowanego działania. Ponieważ jednak mamy do czynienia z mnożeniem przez 0 (bowiem $a = 0$), zmienna d przyjmuje wartość zmiennej b , czyli 2.
8. Wypisujemy 2.
9. Zmiennej b przypisywana jest reszta z dzielenia 14 przez 3, a zatem $b = 2$.
10. Wypisujemy 2.

Wynikiem zastosowania algorytmu będzie wypisanie liczb:

1	6
2	6
3	2
4	2

Polecenie 1

Zapisz algorytm, wykorzystując wybrany język programowania.

Zadanie 2

Treść zadania:

Sprawdź, jakie wartości przyjmą zmienne a, b, c, d po wykonaniu poniższego pseudokodu.

```
1 a = 0 || true && true
2 b = false
3 c = !a
4 d = true || (true && !c || b && !a)
5 c = !c
6 a = a == b
```

Rozwiązanie:

Przeanalizujemy kolejne etapy wykonywania pseudokodu.

1. Zmiennej a przypisujemy wartość działania logicznego `0 || true && true`. Wartość `0` jest równoważna wyrażeniu `false`, a priorytet operatora `&&` jest wyższy niż operatora `||`. Działanie można zapisać jako `false || (true && true)`. Zmienna a przyjmuje zatem wartość `true`.
2. Zmienna b przyjmuje wartość `false`.
3. Zmiennej c zostaje przypisana wartość `false` (zanegowana wartość zmiennej a).
4. Zmienna d przyjmuje wartość skomplikowanego wyrażenia logicznego. Skorzystamy z faktu, że alternatywa logiczna daje wynik `true`, gdy jedno ze zdań jest prawdziwe. Pierwszy element to właśnie `true`, a zatem wartości wyrażenia w nawiasie nie musimy obliczać. Zmienna d przyjmuje wartość `true`.
5. Następuje **negacja** zmiennej c. Zmienna przyjmuje wartość `true`.
6. Zmiennej a jest przypisywana wartość wyrażenia logicznego, które wymaga sprawdzenia równoważności zmiennych a i b. Zmienna a ma wartość `true`, natomiast zmienna b – `false`, a zatem a nie jest równe b. W rezultacie zmienna a przyjmuje wartość `false`.

Oto rozwiązanie zadania: a = false, b = false, c = true, d = true.

Polecenie 2

Zapisz algorytm przedstawiony za pomocą pseudokodu, wykorzystując wybrany język programowania.

Słownik

algebra Boole'a

algebra wyrażeń logicznych, stosowana m.in. w informatyce oraz elektronice; rachunek zdań, w którym wykorzystuje się wartości 1 (prawda) oraz 0 (fałsz)

alternatywa

(suma logiczna) wyrażenie logiczne złożone z dwóch zdań p i q , które jest prawdziwe wtedy, gdy przynajmniej jedno ze zdań (p lub q) jest prawdziwe

koniunkcja

(iloczyn logiczny) wyrażenie logiczne złożone z dwóch zdań p i q , które jest prawdziwe tylko wtedy, gdy oba zdania (p oraz q) są prawdziwe

negacja

jednoargumentowe działanie wykorzystywane do określenia przeciwnej wartości logicznej podanego argumentu

Schemat interaktywny

Polecenie 1

Dane nam zostało wyrażenie:

```
1 d = (15 + a * 2 == b % c * 15) || (a + c == b) && (a + a % c +
```

Policz wartość zmiennej d . Przyjmij następujące wartości zmiennych:

```
1 a = 23
```

```
2 b = 26
```

```
3 c = 3
```

Specyfikacja problemu:

Dane:

- a – liczba całkowita
- b – liczba całkowita
- c – liczba całkowita

Wynik:

Wartość zmiennej d policzona zgodnie z wyrażeniem zaprezentowanym w poleceniu.

Sprawdź swoje rozwiązanie z prezentacją. Następnie przygotuj podobne zadanie dla kogoś z klasy, a koleżanka lub kolega niech ułożą zadanie dla ciebie.

<https://zpe.gov.pl/b/PUmXIAwbV>

Naszym zadaniem jest obliczenie wartości, jaką przyjmie zmienna d:

```
1 a = 23
2 b = 26
3 c = 3
4 d = (15 + a * 2 == b % c *
      15) || (a + c == b) && (a
      + a % c + 1 == b) || (a +
      c > b) && (a + b > c)
```

Nawiasy ułatwiają grupowanie elementów wyrażeń alternatywy lub koniunkcji.

2

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PUmXIAwbV>

Obliczamy wartość wyrażenia wewnątrz pierwszej pary nawiasów:

```
1 15 + a * 2 == b % c * 15
```

Podstawiamy wartości zmiennych:

```
1 15 + 23 * 2 == 26 % 3 * 15
```

Należy pamiętać, że operator % ma taki sam priorytet jak *, a zatem najpierw obliczamy resztę z dzielenia, a następnie wykonujemy mnożenie:

```
1 15 + 46 == 2 * 15
```

Otrzymujemy zdanie nieprawdziwe. Ta część wyrażenia ma wartość false.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PUmXIAwbV>

Kolejna część wyrażenia to:

```
1 a + c == b
```

Podstawiamy wartości zmiennych:

```
1 23 + 3 == 26
```

Otrzymujemy zdanie prawdziwe. Ta część wyrażenia ma wartość `true`.

4

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PUmXIAwbV>

Przechodzimy do kolejnego wyrażenia:

```
1 a + a % c + 1 == b
```

Podstawiamy wartości zmiennych:

```
1 23 + 23 % 3 + 1 == 26
```

Należy pamiętać, że operator `%` ma wyższy priorytet niż `+`:

```
1 23 + 2 + 1 == 26
```

Otrzymujemy zdanie prawdziwe. Wyrażenie ma wartość `true`.

5

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PUmXIAwbV>

Wewnątrz kolejnej pary nawiasów zapisano nierówność:

```
1 a + c > b
```

Podstawiamy wartości zmiennych:

```
1 23 + 3 > 26
```

Otrzymujemy zdanie nieprawdziwe. Ta część wyrażenia ma wartość `false`.

6

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PUmXIAwbV>

Pozostało ostatnie wyrażenie:

```
1 a + b > c
```

Podstawiamy wartości zmiennych:

```
1 23 + 26 > 3
```

Jest to prawda, więc ta część wyrażenia ma wartość `true`.

7

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PUmXIAwbV>

Po przekształceniach w linii numer 4 otrzymaliśmy następujące wyrażenie logiczne:

```
1 false || true && true ||  
false && true
```

8

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PUmXIAwbV>

Koniunkcja ma wyższy priorytet niż alternatywa.
Po wykonaniu operacji `&&` otrzymujemy:

```
1 false || true || false
```

Całe wyrażenie jest prawdziwe. Zatem zmienna `d` przyjmuje wartość `true`.

9

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PUmXIAwbV>

Teraz przygotuj dla koleżanki lub kolegi skomplikowane wyrażenie logiczne. Niech zawiera ono co najmniej koniunkcje oraz alternatywy.

10

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PUmXIAwbV>

Przypominamy na koniec poznane operatory logiczne:

`!` – negacja

`&&` – koniunkcja

`||` – alternatywa

`==` – równość

Polecenie 2

Zapisz wyrażenie zaprezentowane w prezentacji, wykorzystując schemat blokowy lub wybrany język programowania.

1

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Wskaż, która spośród przedstawionych operacji nie jest operacją arytmetyczną.

☐ równoważność

☐ dodawanie

☐ mnożenie

☐ potęgowanie

Ćwiczenie 2



Połącz nazwy operacji arytmetycznych z odpowiadającymi im operatorami.

dzielenie

+

dodawanie

/

mnożenie

%

reszta z dzielenia

*

Ćwiczenie 3



Połącz nazwy operacji logicznych z odpowiadającymi im operatorami.

przypisanie wartości

&&

koniunkcja

!

równoważność

==

negacja

=

Ćwiczenie 4



Wskaż, które spośród przedstawionych operatorów mają ten sam priorytet. Zaznacz wszystkie poprawne odpowiedzi.

☐ /

☐ +

☐ *

☐ %

Ćwiczenie 5



Wskaż wartość, jaką przyjmuje wyrażenie: $3 + 23 * 2 \% 3$.

☐ 4

☐ 5

☐ 3

Ćwiczenie 6



Wskaż wyrażenia, których wartość wynosi `true`. Zaznacz wszystkie poprawne odpowiedzi.

☐ `false || true`

☐ `true && true`

☐ `true`

☐ `!true`

Ćwiczenie 7



Wskaż, która spośród przedstawionych operacji logicznych ma najwyższy priorytet.

alternatywa

równoważność

koniunkcja

negacja

Ćwiczenie 8



Wskaż, które wyrażenie musi być spełnione, aby można było utworzyć trójkąt z odcinków o długościach a , b , c . Nierówność trójkąta to twierdzenie, które mówi, że długość dowolnego boku trójkąta jest mniejsza niż suma długości dwóch pozostałych boków.

☐ $(a - b > c) \ \&\& \ (b + c > a) \ \&\& \ (c + a > b)$

☐ $(a + b > c) \ || \ (b + c > a) \ || \ (c + a > b)$

☐ $(a + b > c) \ \&\& \ (b + c > a) \ \&\& \ (c + a > b)$

☐ $(a + a > c) \ \&\& \ (b + b > a) \ \&\& \ (c + c > b)$

Dla nauczyciela

Autor: Zespół autorski Contentplus.pl sp. z o.o.

Przedmiot: Informatyka

Temat: Operatory – ćwiczenia

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres podstawowy i rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

- I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.
- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

- I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres podstawowy. Uczeń:

- 1) planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania).
- 3) wyróżnia w problemie podproblemy i charakteryzuje: metodę połowienia, stosuje podejście zachłanne i rekurencję;
- 4) porównuje działanie różnych algorytmów dla wybranego problemu, analizuje algorytmy na podstawie ich gotowych implementacji;
- 5) sprawdza poprawność działania algorytmów dla przykładowych danych.

- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres podstawowy. Uczeń:

1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2);

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

2) stosuje zasady programowania strukturalnego i obiektowego w rozwiązywaniu problemów;

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Przeanalizujesz sposób zapisywania i wykonywania działań w programach.
- Scharakteryzujesz sposób wykonywania operacji arytmetycznych oraz logicznych w zadaniach.
- Zapiszesz działania arytmetyczne oraz wyrażenia logiczne.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- oprogramowanie dla języka C++, w tym kompilator GCC/G++ 4.5 (lub nowszej wersji) i Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio;
- oprogramowanie dla języka Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji);
- oprogramowanie dla języka Python 3 (lub nowszej wersji), w tym PyCharm lub IDLE.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Operatory – ćwiczenia”. Nauczyciel prosi uczniów o zapoznanie się z multimedium w sekcji „Schemat interaktywny”.

Faza wstępna:

1. Nauczyciel prosi wybraną osobę o odczytanie tematu lekcji, a następnie określa cele i kryteria sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel prosi wybranego ucznia lub uczniów o przedstawienie sytuacji problemowej związanej z tematem lekcji oraz w odniesieniu do poznanych języków programowania.

Faza realizacyjna:

1. **Praca z tekstem.** Uczniowie analizują treści z sekcji „Przeczytaj” wyświetlone na tablicy.
2. **Praca z multimedium.** Uczniowie pracują w parach. Analizują treść polecenia nr 1: „Zapoznaj się z prezentacją. Następnie przygotuj podobne zadanie dla kogoś z klasy, a koleżanka lub kolega niech ułoży zadanie dla ciebie.” z sekcji „Schemat interaktywny”. Wybrana grupa omawia rozwiązanie na forum klasy.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują ćwiczenia nr 1-5 z sekcji „Sprawdź się”. Nauczyciel sprawdza poprawność wykonanych zadań, omawiając je wraz z uczniami.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat i cele lekcji zawarte w sekcji „Wprowadzenie”. W kontekście ich realizacji następuje omówienie ewentualnych problemów z rozwiązaniem ćwiczeń z sekcji „Sprawdź się”.

Praca domowa:

1. Uczniowie wykonują ćwiczenia 6-8 z sekcji „Sprawdź się”.
2. Uczniowie powtarzają zaprezentowane rozwiązania zadań 1 i 2 z sekcji „Przeczytaj”.
3. Uczniowie wykonują polecenie 2 z sekcji „Schemat interaktywny”.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla oprogramowania Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio.
- Oficjalna dokumentacja techniczna dla języka C++.
- Oficjalna dokumentacja techniczna dla kompilatora GCC/G++ 4.5 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla języka Java SE 8 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Eclipse 4.4 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla języka Python 3 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania PyCharm lub IDLE.

Wskazówki metodyczne:

- Treści w sekcji „Schemat interaktywny” można wykorzystać jako materiał służący powtórzeniu materiału.