

Podstawy programowania

Laboratorium 3 - operatory oraz instrukcje warunkowe i wyboru

mgr inż. Krzysztof Szwarz

krzysztof@szwarz.net.pl

Sosnowiec, 9 marca 2020

Definicja

Operator jest konstrukcją językową zwracającą pewną wartość. Operacje wykonywane są na operandach, czyli argumentach operatorów. Przykładowo dla operacji odejmowania $7 - 2$ operandami operatora odejmowania - są cyfry **7** oraz **2**.

Operatory arytmetyczne

Znak	Opis	Przykład	Wynik
+	Dodawanie	$2 + 2$	4
-	Odejmowanie	$2 - 2$	0
*	Mnożenie	$2 * 2$	4
/	Dzielenie całkowite	$2 / 2$	1
%	Reszta z dzielenia (modulo)	$2 \% 2$	0
+	Unarny operator plus	+2	2
-	Unarny operator minus	-2	-2

Przykład

```
int numberOne = 5;
int numberTwo = 2;
int result = numberOne - numberTwo; // 3
result = numberOne + numberTwo; // 7
System.out.print(numberOne*numberTwo); //10
char character = 'k';
System.out.print(character); // k
System.out.print(+character); // 107
```

Operatory inkrementacji i dekrementacji

Operator inkrementacji ($++$) zwiększa wartość zmiennej o jeden, a operator dekrementacji ($--$) ją zmniejsza. Oba operatory występują w postaci przedrostkowej ($++x$) oraz przyrostkowej ($x++$). Dla pierwszej formy (nazywanej również preinkrementacją/predekrementacją) zwiększenie wartości następuje przed użyciem zmiennej, a dla drugiej (nazywanej również postinkrementacją/postdekrementacją) po jej wykorzystaniu.

Przykład

```
int numberOne = 5;
int numberTwo;
numberTwo = ++numberOne;
// numberOne = 6; numberTwo = 6;
numberTwo = numberOne++;
// numberOne = 7; numberTwo = 6;
numberTwo = numberOne--;
// numberOne = 6; numberTwo = 7;
numberTwo = --numberOne;
// numberOne = 5; numberTwo = 5;
```

- 1 Napisz program pobierający od użytkownika dwie liczby całkowite, obliczający ich średnią arytmetyczną i wyświetlający wynik.
- 2 Dla napisanego w punkcie 1 programu sprawdź jaki wynik zostanie zwrócony dla wartości 3 oraz 4.
- 3 Zainicjuj zmienną typu byte wartością 128.
- 4 Zainicjuj zmienną typu byte wartością 127, po czym dokonaj postinkrementacji. Wypisz wynik.
- 5 Zainicjuj zmienną o nazwie liczba typu double wartością 1, po czym wypisz wynik dzielenia jej przez zero.

- 6 Zmiennej z zadania 5 przypisz wartość 0, po czym wypisz wynik dzielenia jej przez zero.
- 7 Zainicjuj zmienną o nazwie liczba typu int wartością 1, po czym wypisz wynik dzielenia jej przez zero.
- 8 Napisz konwerter walut przeliczający złotówki na dolary oraz euro.
- 9 Napisz program wyznaczający moment obrotowy przenoszony przez sprzęgło dla mocy N i prędkości obrotowej n_s podanej przez użytkownika (wzór: $M = 9550 \cdot \frac{N}{n_s}$).
- 10 Zainicjuj zmienną typu float wartością 100000001 i odejmij od niej 100000000. Wypisz wynik.

- 11 Pobierz od użytkownika liczbę całkowitą i wypisz jej resztę z dzielenia przez 2 oraz 1. Wpisz liczby od 1 do 5.
- 12 Pobierz od użytkownika liczbę zmiennoprzecinkową i wypisz jej resztę z dzielenia przez 2 oraz 1. Wpisz liczby od 1 do 5 oraz 3,5.

Operatory porównawcze i logiczne

Znak	Opis	Przykład	Wynik
==	Równy	2 == 2	true
!=	Różny	2 != 2	false
>=	Większy bądź równy	2 >= 2	true
<=	Mniejszy bądź równy	2 <= 2	true
<,>	Mniejszy, większy	2 < 2	false
&&	Koniunkcja	true && false	false
	Alternatywa	true false	true
!	Negacja	!true	false

Przykład

```
int numberOne = 2;
int numberTwo = 1;
boolean result;
result = numberOne == numberTwo;
// false
result = numberOne != numberTwo; // true
result = numberOne >= numberTwo; // true
result = numberOne <= numberTwo;
// false
result = numberOne < numberTwo; // false
result = numberOne > numberTwo; // true
result = true && false; // false
result = true || false; // true
result = !true; // false
```

Porównywanie liczb zmiennoprzecinkowych

```
double number = 0.1;
number+=0.1;
// 0.2
number+=0.1;
// 0.30000000000000004
System.out.print(0.3==number);
// false
```

Porównywanie liczb zmiennoprzecinkowych - rozwiązanie

```
double number = 0.1;
number+=0.1;
number+=0.1;
double epsilon = 0.000001;
double absolute = Math.abs(number-0.3);
System.out.print(absolute<=epsilon);
// true
```

- 1 Napisz program pobierający od użytkownika dwie liczby całkowite i wyświetlający informację czy pierwsza liczba jest większa od drugiej.
- 2 Napisz program pobierający od użytkownika trzy liczby całkowite i wyświetlający informację czy pierwsza liczba jest większa od drugiej oraz trzeciej.

Operatory bitowe

Znak	Opis	Przykład	Wynik
&	Koniunkcja binarna	3 & 2	2
	Alternatywa binarna	3 2	3
~	Negacja binarna	~2	-3
^	Alternatywa wykluczająca (XOR)	3 ^ 2	1
<<	Przesunięcie arytmetyczne w lewo	2 << 1	4
>>	Przesunięcie arytmetyczne w prawo	2 >> 1	1
>>>	Przesunięcie arytmetyczne w prawo z wypełnieniem zerami	-2 >>> 1	2147483647

- 1 Napisz program z wykorzystaniem operatora przesunięcia arytmetycznego w prawo z wypełnieniem zerami przesuwaną liczbę 2 o 1 miejsce.

Operatory przypisania

Założenie: zmienna typu int o nazwie x ma wartość 1.

Znak	Opis	Przykład	Wynik
=	Przypisanie	x = 2	2
+=	Zwiększenie wartości	x += 2	3
-=	Zmniejszenie wartości	x -= 2	-1
*=	Pomnożenie wartości	x *= 2	2
/=	Podzielenie całkowite wart.	x /= 2	0
%=	Reszta z dzielenia wart.	x %= 2	1
&=	Koniunkcja wartości	x &= 2	0
=	Alternatywa wartości	x = 2	3
^=	XOR wartości	x ^= 2	3
<<=	Przesunięcie wart. w lewo	x <<= 2	4
>>=	Przesunięcie wart. w prawo	x >>= 2	0
>>>=	Przesunięcie i wyp. zerami	x >>>= 2	0

Opis

Instrukcja warunkowa w Javie ma następującą postać:

```
if (condition)
    instruction;
```

Jedynie w momencie, gdy warunek ma wartość `true` następuje wykonanie instrukcji.

Wykonanie wielu instrukcji

Aby wykonać wiele instrukcji dla jednego warunku należy dodać nawiasy klamrowe zgodnie z poniższym przykładem:

```
if (condition) {  
    instruction1;  
    instruction2;  
  
    :  
  
    instructionN;  
}
```

Słowo kluczowe else

Jeżeli chcemy wykonać dodatkowe instrukcje w momencie, gdy warunek nie jest spełniony należy wykorzystać słowo kluczowe else zgodnie z poniższym przykładem:

```
if (condition) {  
    instruction1;  
    instruction2;  
}  
else {  
    instruction3;  
    instruction4;  
}
```

Instrukcja warunkowa - przykład 1

```
boolean isDividedByTwo = 7%2==0;
if (isDividedByTwo)
    System.out.println("Liczba jest
        podzielna przez dwa");
else
    System.out.println("Liczba nie jest
        podzielna przez dwa");
```

Instrukcja warunkowa - przykład 2

```
if (7%2==0)
    System.out.println("Liczba jest
        podzielna przez dwa");
else
    System.out.println("Liczba nie jest
        podzielna przez dwa");
```

Instrukcja warunkowa - przykład 3

```
if (7%2==0) {  
    System.out.println("Liczba jest  
        podzielna przez dwa");  
}  
else {  
    System.out.println("Liczba nie jest  
        podzielna przez dwa");  
}
```

Słowo kluczowe else if

Jeżeli chcemy sprawdzić dodatkowe warunki należy wykorzystać słowo kluczowe else if zgodnie z poniższym przykładem:

```
if (condition)
    instruction1;
else if (condition2)
    instruction2;
else if (condition3)
    instruction3;
else
    instruction4;
```

Warunki są sprawdzane do momentu spełnienia pierwszego z nich.

Instrukcja warunkowa - przykład 4

```
boolean isDividedByTwo = 7%2==0;
boolean isDividedByThree = 7%3==0;
if (isDividedByTwo)
    System.out.println("Liczba jest
        podzielna przez dwa");
else if (isDividedByThree)
    System.out.println("Liczba jest
        podzielna przez trzy");
else
    System.out.println("Liczba nie jest
        podzielna przez dwa, ani przez
        trzy");
```

- 1 Napisz program pobierający od użytkownika liczbę zmiennoprzecinkową i wypisujący wiadomość „Wpisana liczba jest mniejsza od 50”, jeżeli jest ona mniejsza od 50 i „Wpisana liczba jest większa bądź równa 50” w przeciwnym razie.
- 2 Napisz program pobierający od użytkownika trzy liczby całkowite i wyświetlający informację która z liczb jest największa - „Liczba x jest największa”, gdzie x jest jej lp.
- 3 Napisz program pobierający od użytkownika liczbę całkowitą i wyświetlający jej wartość bezwzględną.

Opis

Instrukcja warunkowa switch stosowana jest w celu zdefiniowania instrukcji dla różnych wyników wyrażenia.

```
switch (variable) {  
    case value1:  
        instruction1;  
        break;  
    case value2:  
        instruction2;  
        break;  
    default:  
        System.out.print("Inne od 1 i 2");  
}
```

Instrukcja wyboru - przykład

```
int variable = 1;
switch (variable) {
    case 1:
        System.out.print("Zmienna ma
            wartosc 1");
        break;
    case 2:
        System.out.print("Zmienna ma
            wartosc 2");
        break;
    default:
        System.out.print("Zmienna ma
            wartosc inna niz 1/2");
}
```

- 1 Napisz program korzystający z instrukcji switch, który pobiera od użytkownika liczbę całkowitą i wypisującą tekst „Zmienna ma wartość 1”, jeżeli ma ona taką wartość. Domyślnie niech zostanie wypisana informacja „Zmienna ma inną wartość”. Usuń instrukcję `break`, uruchom program i wpisz wartość 1.
- 2 Napisz program pobierający od użytkownika znak (skorzystaj z metod `next().charAt(0)`) i wypisujący dowolny tekst w razie `W` (skorzystaj ze switcha). Sprawdź jak zachowa się program dla tekstu użytkownika `w`, `W` oraz `Więc`.

Trójargumentowy operator warunkowy

Opis

Trójargumentowy operator warunkowy pełni rolę skróconej wersji instrukcji warunkowej. Jego budowa została przedstawiona poniżej:

```
condition ? value1 : value2;
```

Jeżeli warunek jest prawdziwy zostanie zwrócona value1, w przeciwnym razie value2.

Operator warunkowy - przykład

```
int variable = 1;
boolean logical = true;
int result = logical ? 2 : variable;
// result = 2
```

- 1 Napisz program korzystający z trójargumentowego operatora warunkowego, który pobiera od użytkownika liczbę całkowitą i przypisuje nowej zmiennej wartość 0 dla wprowadzonej liczby mniejszej od 50 oraz 100 w przeciwnym razie.

Opis

W celu sprawdzenia poprawności typu danych wprowadzonych przez użytkownika można skorzystać z metod hasNext... klasy Scanner (hasNextInt(), hasNextDouble(), ...). Przykład:

```
int number;
Scanner reader = new Scanner(System.in);
if (reader.hasNextInt())
    number = reader.nextInt();
else
    System.out.print("Niepoprawna
wartosc");
```

- 1 Napisz program, który pobiera od użytkownika liczbę zmiennoprzecinkową i przypisuje jej wartość do zmiennej. Niech aplikacja sprawdza poprawność wprowadzonych danych.

Dziękuję za uwagę