



Увод в програмирането

с Java



Оператори и изрази

Съдържание

- Оператори
 - Оператор за присвояване
 - Аритметични оператори
 - Оператори на една променлива
 - Оператори за сравнение
 - Логически оператори
- Изрази
- Домашно

ОПЕРАТОРИ

ОПЕРАТОРИ

Операторите са специални символи, които извършват действие върху един, два или три *операнда* и връщат резултат.

Операнд е променливата, чиято стойност се използва/променя от оператора.

Пример: $a + b < c$

“+” и “<” са *оператори* а, “b” и “c” са *операнди*

ТИПОВЕ ОПЕРАТОРИ

Операторите могат да се делят по различен начин:

- според броя променливи /операнди/, върху които се прилага
- според разположението на операнда спрямо оператора
- според резултата, който се получава от прилагането на оператора

Оператори според броя операнди

Брой операнда (аргументи)	Тип оператор	Пример
1	Унарен (unary)	a++
2	Бинарен (binary)	a + б
3	Тернарен (ternary)	a ? б : c

Оператори според разположението им спрямо операнда

Разположение	Тип оператор	Пример
Преди операнда	Префиксен	--expr
След операнда	Постфиксен	expr++

Унарни оператори

Оператори, които имат един операнд, се наричат унарни. Резултатът от изпълнението им е число, освен при + където може да е текст и при ! където може да е истина или лъжа.

Унарен Оператор	Описание
+	Унарен оператор плюс; индикация за положителна стойност (числата са позитивни и без да се пише +)
-	Унарен оператор минус; прави израз негативен
++	Оператор за инкрементиране; увеличава стойността на операнда с 1
--	Оператор за декрементиране; намалява стойността на операнда с 1
!	Логическо отрицание; сменя стойността на булев израз или променлива

Пример

```
int a = 5;
```

```
a++;
```

```
int b = 4;
```

```
b--;
```

```
System.out.println("a=" + a);
```

```
System.out.println("b=" + b);
```

++A или A++

Има разлика дали операторът ще е пред или след операнда, когато двете са част от по-голям израз.

Пример:

```
int a = 5;
```

```
System.out.print(a++); // 5
```

```
System.out.print(a); // 6
```

```
System.out.print(++a); // 7
```

Аритметични оператори

Аритметични оператори

Оператори за извършване на аритметични операции. Резултатът от изпълнението им е число.

Аритметичен оператор	Описание
+	Оператор за събиране (използва се и за конкатенация на низове – ще го видите в следващите лекции)
-	Оператор за изваждане
*	Оператор за умножение
/	Оператор за деление
%	Деление по модул

Аритметични оператори

Когато се използва операторът за деление / с целочислен тип (integer), върнатият резултат е отново целочислен (без закръгляне). За да се вземе остатъкът от делене на цели числа се използва оператора %.

Пример

```
int a = 7;
```

```
int b = 4;
```

```
int div = a / b;
```

```
int mod = a % b;
```

```
float fDiv = a / b;
```

```
System.out.println("div = " + div); //1
```

```
System.out.println("mod = " + mod); //3
```

```
System.out.println("fDiv = " + fDiv); //1,0
```

Пример

```
float a = 7;
```

```
float b = 4;
```

```
float div = a / b;
```

```
System.out.println("div = " + div); // 1.75
```


Пример

Колко ще се изведе на конзолата?

- `System.out.println(7 / 2);`
- `System.out.println(7.4 / 2);`

Оператори за сравнение

Оператори за сравнение

Дават възможност за сравнение на два операнда. Резултатът от изпълнението им е истина или лъжа.

Оператор за сравнение	Описание
==	Равно
!=	Различно
>	По-голямо
>=	По-голямо или равно
<	По-малко
<=	По-малко или равно

Пример

```
int a = 5;
```

```
int b = 6;
```

```
boolean greater = a > b;
```

```
boolean smaller = a < b;
```

```
boolean diff = a != b;
```

```
System.out.println("a > b ->" + greater);
```

```
System.out.println("a < b ->" + smaller);
```

```
System.out.println("a != b ->" + diff);
```

Логически оператори

Логически оператори

Оператори за работа с булеви данни и булеви изрази.

Логически оператор	Описание
&&	Логическо "И"
	Логическо "ИЛИ"
!	Логическо отрицание; сменя стойността на булев израз или променлива

Таблица на истинността (Truth table)

При логическото И (&&) всички операнди трябва да са истина за да бъде целия израз истина, при логическото ИЛИ (||) е достатъчно един операнд да е истина за да бъде целия израз истина.

x	y	!x	!y	x && y	x y
true	true	false	false	true	true
true	false	false	true	false	true
false	true	true	false	false	true
false	false	true	true	false	false

Short-Circuiting

При логическите оператори се изпълнява т.нар. “short-circuiting”, т.е. стойността на втория операнд се оценява само, ако е необходимо. При оператор „И“, ако първият аргумент е false, вторият не се оценява и стойността на израза е false. При оператор „ИЛИ“, ако първият аргумент е true, вторият не се оценява и стойността на израза е true.

Пример

```
boolean a = true;
```

```
boolean b = false;
```

```
System.out.println(a || b); // true
```

```
System.out.println(a && b); // false
```

Оператори за присвояване

Оператори за присвояване

- Прост оператор за присвояване. Задава стойност на променлива. Вече сте го учили в предишната лекция.
- Комбинирани оператори за присвояване. Позволяват съкратен запис на две операции.

Пример

```
String myName = „Lilly“;
```

```
int a = 5;
```

```
a += 10; // същото като a = a + 10;
```

```
a -= 7; // същото като a = a - 7;
```

```
a *= 4; // същото като a = a * 4;
```

```
a /= 2; // същото като a = a / 2;
```

```
a %= 3; // същото като a = a % 3;
```

```
System.out.println("a = " + a);
```

Побитови оператори и предимство на операторите

Побитови оператори

Оператори, които възприемат променливите като 1 и 0 и извършват действия върху тях:

- изместване наляво/надясно
- промяна на 0 в 1 и обратно
- други

Няма да ги разглеждаме засега.

Предимство на операторите

Някои оператори имат приоритет над други. Операторите с по-висок приоритет се изчисляват преди тези с по-нисък. Операторът `()` служи за промяна на приоритета на операторите и се изчислява пръв, също както в математиката.

В таблицата са показани приоритетите на операторите в Java:

Оператор	Оператор	Предимство
Постфиксни	postfix	expr++ expr--
Унарни	unary	++expr --expr +expr - expr ~ !
За умножение	multiplicative	* / %
За събиране	additive	+ -
Побитово отместване	shift	<< >> >>>
За сравнение	relational	< > <= >= instanceof
За равенство	equality	== !=
Побитово „И“	bitwise AND	&
Побитово изключващо „ИЛИ“	bitwise exclusive OR	^
Побитово „ИЛИ“	bitwise inclusive OR	
Логическо „И“	logical AND	&&
Логическо „ИЛИ“	logical OR	
Тернарен	ternary	? :
За присвояване	assignment	=

* Операторите с по-малък и светъл шрифт няма да бъдат разглеждани в текущата лекция

Задача

1. Напишете израз, който по въведена температура в градуси по Фаренхайт, извежда температурата по Целзий.

$$T_C = \frac{T_F - 32}{1.80}$$

Решение

```
float temperatureF = 70;
```

```
float temperatureC = (temperatureF - 32)/1.80F;
```

```
System.out.println(temperatureC);
```

Задача

2. Напишете израз, който извежда последната цифра на въведено цяло число.

3. Напишете израз, който извежда предпоследната цифра на въведено трицифрено цяло число.

Решение

```
int a = 847;
```

```
int lastDigit = a % 10; // последна цифра
```

```
int beforeLastDigit = (a % 100) / 10; // предпоследна цифра
```

```
System.out.println("Last Digit: " + lastDigit);
```

```
System.out.println("Before Last Digit: " + beforeLastDigit);
```

Условен оператор

Връща различен резултат в зависимост от изпълнението на дадено условие.

Условие ? Стойност1 : Стойност2

Нарича се тернарен оператор, т.к. има 3 аргумента.

Пример

Дадени са 2 цели числа. Да се напише израз, който връща стойността на по-голямото число, умножена по 2.

Решение:

```
int a = 5;
```

```
int b = 7;
```

```
int doubledGreater = (a > b) ? 2 * a : 2 * b;
```

```
System.out.println(doubledGreater);
```

Задача

Дадено е цяло число. Напишете програма, която умножава числото по две, ако е четно и го умножава по три, ако е нечетно.

Решение

```
int a = 7;
```

```
int result = (a % 2 == 0) ? a * 2 : a * 3;
```

```
System.out.println(result);
```


Четене от конзолата

Четене от конзолата

Може да го използвате за решаване на задачите.

```
Scanner input = new Scanner(System.in);
```

```
System.out.print("Enter number:");
```

```
int number = input.nextInt();
```


Изрази

Израз е последователност от оператори, литерали и променливи, която връща някаква стойност.

Изразите имат тип (int, double, boolean ...) и стойност.

Пример:

```
// Изчисляване на лицето на кръг
```

```
double surface = Math.PI * r * r;
```

Пример

```
int x = (10 + 5)/2; // Израз от тип Int
```

```
int y = (x + 2) * (x - 4) + (2 * x - 6)/2; // Израз от тип Int
```

```
boolean areOddNumbers = (x % 2 != 0) && (y % 2 == 1); // Израз от тип boolean
```

Задача

Дадена е следната информация за ученик: - Дали е първи в класа - Резултат от тест

Условието ученикът да завърши успешно е да е бил първи в класа или резултатът от теста му да е по-голям от 60.

Напишете израз, който определя дали ученикът ще се дипломира.

Решение

```
boolean isFirst = false;
```

```
int result = 70;
```

```
boolean graduate = isFirst || result >= 60;
```

```
System.out.println(graduate);
```

Задачи

Задача

1. Напишете израз, който да проверява дали дадено цяло число е четно или нечетно.

Решение

```
int a = 13;
```

```
boolean isEven = a % 2 == 0;
```

```
System.out.println(isEven);
```

Задача

2. Напишете булев израз, който да проверява дали дадено цяло число се дели на 5 и на 7 без остатък.

Решение

```
int a = 77;
```

```
boolean dividesBy5and7 = (a % 5 == 0) && (a % 7 == 0);
```

```
System.out.println(dividesBy5and7);
```

Задача

3. Напишете програма, която по дадени дължина и височина на правоъгълник, изкарват на конзолата неговият периметър и лице.

$$P = 2 * (a + b)$$

$$S = a.b$$

Решение

```
int a = 10;  
int b = 12;  
int perimeter = 2 * (a + b);  
int area = a * b;  
  
System.out.println("Perimeter = " + perimeter);  
System.out.println("Area = " + area);
```

Задача

4. Напишете програма, която приема за вход трицифрено число във формат abc и след това извършва следните действия върху него:

- Пресмята сбора от цифрите на числото.
- Разпечатва на конзолата цифрите в обратен ред: cba.
- Поставя последната цифра на първо място: cab.

Решение

```
int number = 123;

int a = number / 100;    // first digit

int b = (number % 100) / 10; // second digit

int c = number % 10;    // third digit

System.out.println(a + b + c);

System.out.println("" + c + b + a);

System.out.println("" + c + a + b);
```


Задача на лист

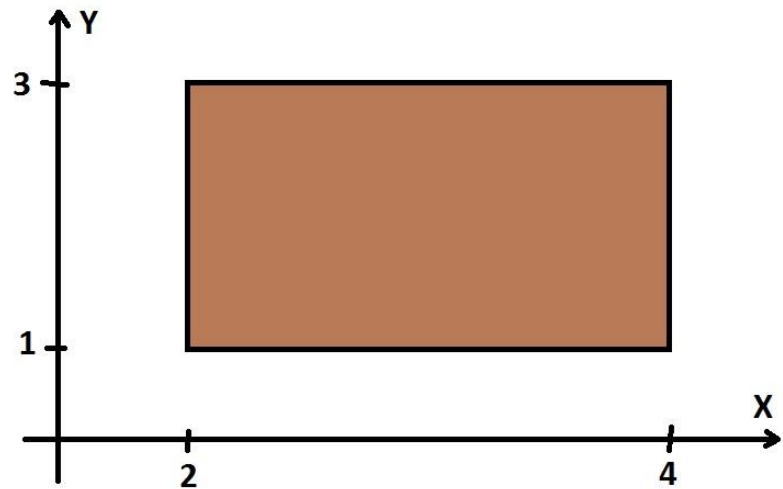
1. Напишете типовете променливи, които знаете. Кога и за какво се използват?

2. Задача

//координати на точка `int x, y;`

Напишете израз, който ще изведе

дали точката е вътре в правоъгълника.



Решение

```
int x = 3;
```

```
int y = 2;
```

```
boolean isSquareInside = (x >= 2 && x <= 4)&&(y >= 1 && y <= 3);
```

```
System.out.println(isSquareInside);
```

Домашно

Задача: 1

За даден имот имаме следните характеристики:

- Площ в кв.м. (в променливата `area`)
- Дали е с добра локация (в променливата `isGoodLocation`)

Един имот е престижен, ако е с добра локация или площта му е по-голяма от 100 кв.м.

Напишете израз, който определя дали имот с дадени характеристики е престижен.

Задача: 2

За даден автомобил са дадени следните характеристики:

- Възраст в години (age)
- Цена в хил.лв. (price)

Един автомобил е от висок клас, ако е по-стар от 5 години и струва над 20 хил.лв. или е на 5 или помалко години и струва над 40 хил.лв.

Напишете израз, който определя дали автомобил с дадени характеристики е от висок клас.

Задача: 3

Напишете израз, който намира и извежда в конзолата абсолютната стойност на дадено цяло число.

Абсолютната стойност на дадено число X се означава като $|x|$ и се изчислява по следния начин:

$$|x| = x \text{ при } x > 0$$

и

$$|x| = -x \text{ при } x < 0$$

Задача: 4

//Дадени са 3 числа.

```
int a, b, c;
```

Напишете програма, която проверява дали съществува триъгълник с дължини на страните, равни на дадените числа.

(За да съществува триъгълник с дадени дължини на страните, трябва тези страни да са положителни числа и всяка страна да е по-малка от сбора на другите 2 страни.)

Задача

Прочетете 3 глава от книгата: „Оператори и изрази”

<http://www.introprogramming.info/intro-java-book/read-online/glava3-operatori-i-izrazi/>