

Шартты операция

Бұл операцияның үш операнды бар. Бірінші кез-келген логикалық өрнек, яғни нәтижелері true немесе false болады, сосын сұрақ таңбасы, содан кейін қос нүкте арқылы ажыратылған екі кез-келген өрнек жазылады,

мысалы,

$$x < 0 ? 0 : x$$

$$x > y ? x - y : x + y$$

Шартты операция былай орындалады. Бірінші логикалық өрнек есептелінеді. Егер true нәтижесі берілсе, онда сұрақ белгісінен кейінгі бірінші өрнек орындалады және оның мәні барлық операция нәтижесі болады. Соңғы өрнек орындалмайды. Егер false нәтижесі берілсе, онда соңғы өрнек орындалады және оның мәні барлық операция нәтижесі болады.

Бұл былай жазуға мүмкіндік береді $n = 0 ? i \text{ә} : m / n$ нөлге бөлінуден қорықпаймыз.

Өрнек

Тұрақтылар, айнымалылар, операциялар, әдістерді шақыру және жақшалар өрнек (expressions) деп аталады. Өрнектерді есептеу кезінде 4 ереже қолданылады:

1. Бір түрдегі операциялар солдан оңға қарай орындалады: $x + y + z$ былай есептелінеді $(x + y) + z$. Ерекшелік: меншіктеу операциясы оңнан солға қарай орындалады: $x = y = z$ есептелінеді $x = (y = z)$.
2. Сол жақтағы операнд оңжақтығы операндтан бұрын орындалады.
3. Операндтар операция алдында толық есептелінеді.
4. Құрама операцияны орындау кезінде сол жақтағы меншіктеу мәні оң жақтағы мән үшін сақталады.

Мысалы

```
int a = 3, b = 5;
```

Өрнек нәтижесі $b + (b = 3)$ мәні 8 тең болады; бірақ $(b = 3) + a$ мәні 6 тең болады. Өрнек $b += (b = 3)$ мәні 8 тең болады, себебі жоғарығыдағылардың біріншісі орындалады.

Операциялардың орындалу реті

Операциялар реті кемімелі түрде берілген. Бір жолдағы операциялар бірдей болады.

- 1. Постфикстік операция ++ және --.
- 2. Префикстік операция ++ және --, толықтыру ~ және теріске шығару !.
- 3. Түрлерді келтіру.
- 4. Көбейту *, бөлу / және қалдықты табу %.
- 5. Қосу + және азайту -.
- 6. Ығыстырулар <<, >>, >>>.
- 7. Салыстыру >, <, >=, <=.

- 8. Салыстыру $==$, $!=$.
- 9. Биттік конъюнкция $\&$.
- 10. Биттік болдырмау НЕМЕСЕ \wedge .
- 11. Биттік дизъюнкция $|$.
- 12. Конъюнкция $\&\&$.
- 13. Дизъюнкция $||$.
- 14. Шартты операция $?:$.
- 15. Меншіктеу $=$, $+=$, $-=$, $*=$, $/=$, $\%=$, $\&=$, $\wedge=$, $|=$, \ll , \gg , \ggg .

Операторлар

Кез келген программалау тілі алгоритмдерден тұрады.

Олар тіл *операторлары* (statements) деп аталады.

Операторлар сызықтық есептеулерден, шартты операторлардан және циклдік операторлардан тұрады.

Java тілінің операторлары:

- айнымалылар мен басқа объектілерді жазатын операторлар;
- өрнектер операторлары;
- меншіктеу операторы;
- шартты оператор if;
- үш цикл операторы while, do-while, for;
- таңдау операторы switch;
- ауыстыру операторы break, continue және return;
- өріс (блок) {};
- бос оператор — нүктелі үтір.

Ескету

Java тілінде goto операторы жоқ.

Кез-келген оператор нүктелі үтірмен аяқталады.

Нүктелі үтірді кез-келген өрнек соңынан қойсақ ол оператор (expression statement) болады. Бірақ оның мәні тек меншіктеу операциясында, инкрементте, декрементте және әдістерді шақыруда ғана анықталады. Басқа жағдайларда мәні болмайды, себебі есептеу мәні жоғалады.

Өріс

Өріс нөлден немесе бір оператор ретінде орындалатын бірнеше операторлардан тұрады. Мысалы, $\{x = 5; y = ?;\}$. Бос өрісті де жазуға болады $\{\}$.

Операторлар өрісі айнымалылардың қолдану аймаған шектеу және программа мәтінін оқуды жақсарту үшін қолданылады.

Меншіктеу операторлары

Кез келген операция соңындағы нүктелі үтір меншіктеу операторы болады. Меншіктеу оператордың негізгі бөлігі болады.

Операция мен меншіктеу операторының айырымы тек теориялық мінездемесінде. Меншіктеу көбінесе операция оператор ретінде қолданылады.

Тармақталу операторы

Шартты оператор (if-then-else statement) Java тілінде былай жазылады:

```
if (логӨрнек) оператор1 else оператор2
```

Бірінші логикалық өрнек орындалады. Егер нәтиже true болса, онда *оператор1* орындалып шартты оператор аяқталады, *оператор2* орындалмайды, содан кейін келесі if операторы орындалады. Егер нәтиже false болса, онда *оператор2* орындалады да оператор1 орындалмайды.

Шартты оператор қысқартылған болуы мүмкін (if-then statement):

```
if (логӨрнек) оператор1
```

Бұл жағдайда өрнек нәтижесі false болса ештеме орындалмайды.

- Мысал 1:
- `if (a < b) {`
- `x = a + b; }` `else {`
- `x = a — b;`
- `}`
- Бұтақтар фигуралық жақшаға алынып жазылады бұл әрбір бұтаққа операторларды қосуға мүмкіндік жасайды.

Мысал 2:

```
if (n == 0){  
    sign = 0;  
} else if (n < 0){  
    sign = -1;  
} else {  
    sign = 1;  
}
```

Мынандай жағдай болуы да мүмкін
("dangling else"):

```
int ind = 5, x = 100;
```

```
if (ind >= 10) if (ind <= 20) x = 0; else x = 1;
```

Бұл ережені блок арқылы жазып бөлуге
болады:

```
if (ind > 10) {if (ind < 20) x = 0; else x = 1;}
```

Бұл шарттарды оңайлатып жазу үшін
логикалық операцияларды қолданған
тиімді болады, мысалы:

```
if (ind >= 10 && ind <= 20) x = 0; else x = 1;
```

Листинг 1.4. Квадрат теңдеуді шешу

```
class QuadraticEquation{
public static void main(String[] args){
double a = 0.5, b = -2.7, c = 3.5, d, eps=1e-8;
if (Math.abs(a) < eps)
if (Math.abs(b) < eps)
if (Math.abs(c) < eps) // Барлық коэффициенттер
    нөлге тең
System.out.println("Шешуі — кез-келген сан");
else
System.out.println("Шешуі жоқ");
else
System.out.println("x1 = x2 = " +(-c / b) );
```

```
else { // Коэффициенттер нөлге тең емес
if((d = b**b — 4*a*c)< 0.0){ // Түбірлер
    комплексті
d = 0.5 * Math.sqrt(-d) / a;
a = -0.5 * b/ a;
System.out.println("x1 = " +a+ " +i " +d+
",x2 = " +a+ " -i " +d);
} else {
// Нақты түбірлер
d =0.5 * Math.sqrt(d) / a;
```

```
a = -0.5 * b / a;
```

```
System.out.println("x1 = " + (a + d) + ", x2 = " + (a -  
d));
```

```
}
```

```
}
```

```
)
```

```
}
```

Бұл программада `abs` және квадрат түбір `sqrt` модулдерін есептеу әдісі қолданылды. Онда Java API классының `Math` қолданылды.

Барлық нақты мәндер жуық есептелінеді.

Теңдеудің коэффициентті нөлге тең болады, егер оның модулі мына саннан кіші болса - 0,00000001.

