

Дәріс-3. Операциялар, тіл операторлары мен функциялары

Жоспар:

1. Салыстыру операциялары. Биттік операциялар. Меншіктеу операциялары. Шартты операция. Өрнектер
2. Операторлар. Меншіктеу операторы. Шартты оператор. Цикл операторлары. Continue, break операторы. Вариант операторы.
3. Java тілінің функциялары. Трансцендентті және экспоненциалды функциялар.
4. Жиымдар. Жиымдарды жариялау. Көп өлшемді жиымдар

Java тілінің операциялары

1. Салыстыру операциялары

Java тілінде шамалар бойынша бүтін сандарды салыстырудың 6 операциясы бар:

- үлкен > ;
- кіші < ;
- үлкен немесе тең >= ;
- кіші немесе тең <= ;
- тең == ;
- тең емес != .

Қосарланған символдар бос орынсыз жазылады, бірақ олардың орнын ауыстыруға болмайды, мынадай => жазба дұрыс емес, >= болып жазылуы керек. Салыстыру нәтижесі— true немесе false мәні болады, мынадай салыстыру 3 != 5 мәні – true, ал мынадай салыстыру 3 == 5 нәтижесі false болады.

Күрделі салыстыруларды жазу үшін логикалық операцияларды қолдану керек. Мысалы, $a < x < b$ сияқты теңсіздіктерді тексеру кезінде төмендегідей түрде

$(a < x) \&\& (x < b)$ немесе
 $a < x \&\& x < b$ жазу керек.

Меншіктеу операциялары

Қарапайым меншіктеу операциялары (simple assignment operator) = теңдік таңбасы арқылы жазылады, оның сол жағында айнымалы, ал оң жақта осы айнымалыға сәйкес мән тұрады:

$x = 3.5,$
 $y = 2 * (x - 0.567) / (x + 2),$
 $b = x < y,$
 $bb = x >= y \&\&b.$
 $x = 3.5, y = 2 * (x - 0.567) / (x + 2), b = x < y, \quad bb = x >= y \&\&b.$

Шартты операция

Бұл операция үш операндтан тұрады. Алдымен кез-келген логикалық өрнек жазылады, оның нәтижесі true немесе false болуы тиіс, содан кейін сұрақ

белгісі, ал содан кейін қос нүктемен бөлінген екі өрнек жазылады. Мысалы,

$x < 0 ? 0 : x$
 $x > y ? x - y : x + y$

Осыны пайдаланып, былай жаза аламыз: $n == 0 ? m : m / n$;

Шартты операция шағын тармақтарды жазуға өте ыңғайлы болып келеді.

Өрнектер

Өрнектер (expressions) тұрақтылардан, айнымалылардан және соларға қолданылатын операциялардан, жай жақшалардан тұрады. Өрнектерді есептеу кезінде 4 ереже орындалады:

1. Бір приоритетті операциялар солдан оңға қарай есептеледі:

$x + y + z$ өрнегі есептеу былай орындалады $(x + y) + z$.

2. Меншіктеу операциясы оңнан солға қарай есептелінеді :

$x = y = z$ өрнегін есептеу былай орындалады $x = (y = z)$.

2. Сол жақтағы операнд оң жақтағыдан ерте есептеледі.

3. Операндтар операцияға дейін алдын-ала есептеледі.

4. Құрамалы меншіктеу операциясының орындалуы алдында сол жақтың мәні оң жақта қолдану үшін сақталады.

Төмендегі мысалдар бастапқы үш ереженің қолдану ерекшеліктерін көрсетеді:

`int a = 3, b = 5;`

`b + (b = 3)` өрнегінің нәтижесі 8 болады; `(b = 3) + b` өрнегінің нәтижесі 6 болады.

Төртінші ереже арқылы `b += a += b += 7` өрнегінің мәні 20-ға тең болады.

Операторлар

Операторлар (statements) – алгоритмдерді жазу құралы.

Java тілінде **goto** операторы жоқ. Барлық операторлар нүктелі үтірмен аяқталады. **Блок.** Операторлар блогы программаның мәтінінің оқылуын жақсарту үшін және айнымалылардың әсер ету аймағын шектеу үшін жиі қолданылады. Мысалы, `{x = 5; y = ?;}`. Бос блокты да жазуға болады: `{ }`.

Шартты оператор

Шартты оператор (if-then-elsestatement). Java тілінде былай жазылады:
if (логикалық өрнек) оператор1; else оператор2;

Шартты оператор қысқартылған түрде болуы да мүмкін (if-then statement): **if (логикалық өрнек) оператор1;**

Логикалық өрнек false болған жағдайда еш нәрсе орындалмайды.

Циклдік операторлар

while операторы:

while (өрнек) оператор;

Екінші циклдік оператор: do-while

do оператор while(өрнек);

Алдымен оператор орындалады, содан кейін логикалық өрнек тексеріледі. Логикалық өрнек true болғанша цикл қайталана береді.

for операторы:

for(өрнек1; өрнек; өрнек2) оператор;

Ауыстыру операторы бірнеше бағыт бойынша тармақталуды орындайды: **switch (өрнек) {**

case өрн1: оператор1 case өрн1: оператор2

.....

case өрнN: операторNdefault: операторDef }

Мысалы: switch(dayOfWeek) {

case 1: case 2: case 3: case 4:case 5: System.out.println("Week-day"); break;

case 6: case 7: System.out.println("Week-end"); break;

default: System.out.println("Unknown day");

}

Әрбір жолдағы әрекеттерді **break** операторымен аяқтауды ұмытпау керек.

1 Java тілінің функциялары

Math класы математикалық есептеулерді орындайтын статикалық әдістер жиынтығынан тұрады : **Math.PI ($\pi=3.14159$)**

Трансцендентті функциялар

Төменде double (радианмен берілген бұрыш) типті параметрлерді қабылдайтын және double типті параметрлерді қайтаратын үш әдіс берілген: **Math.sin(x) Math.cos(x) Math.tan(x)**

Экспоненциалды функциялар

Math класында экспоненциалды әдістер анықталған.

(double типті аргументтер және нәтиже):

Math.exp(x) $-e^x$ **Math.log(x,y)** $-\ln x_y$

Math.pow(x) $-x^y$ **Math.sqrt(x)** $-\sqrt{x}$

Дөңгелектеу функциялары.

Math класында келесідей дөңгелектеу әдістері анықталған:

Math.abs(x) $-|x|$

Math.max(x,y) - x пен y мәнінің ең үлкенін қайтарады;

Math.min(x,y) - x пен y мәнінің ең үлкенін қайтарады;

Math.ceil(x) - x-ке тең немесе x-тен үлкен бүтін кіші санды қайтарады;

Math.floor(x) - x-ке тең немесе x-тен кіші бүтін үлкен санды қайтарады;

Math rint(x) – x-ке жақын бүтін мәнді қайтарады.

Math.round(x) - жақын бүтін int мәніне дейін дөңгелектелген x-ті қайтарады(аргументі float, нәтижесі int).

Қосымша келесі әдістермен анықталған:

Math.random() – 0 мен 1 аралығындағы кездейсоқ санды қайтарады(нәтижесі double).

Math.toRadians(x) - градусы радианға ауыстырады (аргументі де, нәтижесі де double).

Math.toDegrees(x) - радианды градусқа ауыстырады (аргументі де,нәтижесі де double).

Жиымдар

Жиым — жедел жадыдағы көршілес ұяшықта сақталынатын бір типті айнымалылар жиынтығы. Java тіліндегі жиымдар сілтемелі типке жатады. Сипаттама үш кезеңде жүргізіледі.

Жиымды хабарлау. Бірінші кезең—хабарлау (declaration).Бұл кезеңде тек жиымға сілтеме типіндегі айнымалылар анықталады. Мысалы,

double[] a, b; екі айнымалы анықталған—double типті a мен b жиымдары. Екінші кезең — анықтау (installation). Бұл кезеңде жиым элементтерінің саны көрсетіледі, айнымалы жиым адресін алады.

Мысалы, **a=newdouble[5];**

Элементтері: **a[0],a[1],a[2],a[3],a[4].**

Үшінші кезең — инициалдау (initialization). Бұл кезеңде жиым элементтері бастапқы мәндерді қабылдайды. Мысалы,

```
a[0] = 0.01; a[1] = -3.4; a[2] = 2.89; a[3] = 4.5; a[4] = -6.7; for (int i = 0; i < 100; i++) b[i] = 1.0 /i;
```

```
for (int i = 0; i < 50; i++) t[i] = 2 * i + 1;
```

Жиым элементтері — бұл кез келген бір типтегі айнымалылар, осы типке қатысты әр түрлі операцияларды орындауға болады:

(a[2] + a[4]) / a[0]; т.с.с.

Көп өлшемді жиымдар

Java–да жиым элементтері ретінде жиымның өзі бола алады. Хабарлауға болады: **char[] [] c;**

бұған эквивалентті түрде былай да жазуға болады: **char c[] c[];**

немесе **char c[][];**

Сыртқы жиымды да анықтаймыз:

```
c = new char[3][];
```

c — жиымның үш жиым элементтерінен тұратын жиым. Енді оның жиым-элементтерін анықтай аламыз:

```
c[0] = new char[2];
```

```
c[1] = new char[4]; c[2] = new char[3];
```

Бақылау сұрақтар:

1. Java тілінің қандай операциялары бар?
2. Меншіктеу операциялары қалай жазылады?
3. Шартты операция неше операндтан тұрады?
4. Өрнектер неден тұрады?
5. Операторлар деген не?
6. Шартты операторы Java тілінде қалай жазылады?
7. Қандай циклдік операторлар бар?
8. for операторы?
9. Ауыстыру операторы нені орындайды?
10. Math классы неден тұрады?
11. Жиым деген не?