Дәріс №4
Таңдау операторы

Таңдау операторы switch тармақталудың бірнеше бағытын көрсетеді. Әрбір тармақ бүтін түрдегі тұрақтылардан немесе тұрақты өрнектерден тұрады (long түрі емес) және мән тұрақтымен сәйкес келсе таңдайды. Мына түрде жазылады.

switch (бүтін Өрнек){

case тұрақтыӨрнек1: оператор1

case тұрақтыӨрнек2: оператор2

. . . . .

case тұрақтыӨрнекN: операторN

default: операторDef

}

Жақша ішіндегі (бүтінӨрнек) byte, short, int, char түрлерінің біреуінде болады тек long түрінде емес.

Таңдау операторында барлық тұрақты өрнектер алдын ала есептелінеді. Сосын сол тұрақтыға сәйкес келетін өрнек есептелінеді.

Егер бір операторды әртүрлі бұтақта орындағымыз келсе case операторынан бірнеше метканы қолдану керек. Қарапайым мысал.

```
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.printlnt"Unknown day");
}
```

Қайталану операторлары

Негізгі цикл операторы while — былай жазылады:

while (логӨрнек) оператор

Бірінші логикалық өрнек орындалады. Егер нәтиже true болса, онда оператор орындалады. Содан кейін тағы да *логӨрнек* тексеріліп оператор орындалады, бұл қашан false мәні шыққанша жалғаса береді. Егер *логӨрнек* басында false мәніне тең болса, онда *оператор* бірде бір рет орындалмайды.

Цикл операторы бос болуы да мүмкін, мысалы, код фрагменті:

```
int i = 0;
double s = 0.0;
while ((s += 1.0 / ++i) < 10);
```

мұнда, қашан s қосындысы 10 тең болғанша i рет қосу орындалады.

Ақырсыз цикл құруға да болады:

while (true) *оператор*

Бірақ бұндай циклден шығу жолын қарастыру керек, мысалы, break операторын қолдануға болады. Басқа жағдайда программа орындала береді және оны тоқтату үшін MS Windows 95/98/ME пернелер "комбинациясы" <Ctrl>+<Alt>+<Del>, UNIX пернелер <Ctrl>+<C>, Windows NT/2000 команда Task Manager орындау керек. Егер циклге бірнеше операторларды енгізу керек болса, онда оларды {} блогына алып жазу керек.

Екінші цикл операторы do-while — былай жазылады:

do оператор while *(логӨрнек)*

Бұнда бірінші оператор орындалады содан кейін логикалық өрнек тексеріледі. Цикл қашан логӨрнек true тең болмағанша орындала береді.

do-while оператор кем дегенде бір рет орындалады.

Листинг 1.5. Бисекция әдісі бойынша сызықтық емес теңдеудің түбірін табу.

```
class Bisection{
static double f(double x){
return x*x*x — 3*x*x +3; // Немесе басқа өрнек
}
public static void main(String!] args){
double a = 0.0, b = 1,5, c, y, eps = le-8;
```

```
do{
c = 0.5 *(a + b); y = f(c);
if (Math.abs(y) < eps) break;
// Түбір табылды. Циклден шығамыз.
// Егер [a; c] кесінді соңында
// функция әртүрлі таңбаны қабылдаса:
if (f (a) * y < 0.0) b = c;
// Онда түбір осы жерде. b нүктесін c
   нүктесіне ауыстырамыз
```

```
//Басқа жағдайда:
else a * c;
// a  нүктесін c нүктесіне ауыстырамыз
// Қашан [a; Ь] кесіндісі кішірейгенше
   жалғастыра береміз
} while (Math, abs (b-a) >= eps);
System.out.println("x = " +c+ ", f(" +c+ ") = "
   +y) ;
}
}
```

Bisection класы қиын болады, себебі онда main () әдісінен басқа f(x) функциясын есептеу әдісі де бар. Бұл әдіс көпмүше мәнін есептеп функция мәні ретінде қайтарады және бұл бір оператор арқылы орындалады:

return өрнек

main о әдісіндегі жаңа оператор break циклдің орындалуын қажет болса тоқтатады.

Циклдің үшінші операторы — оператор for — былай жазылады:

for (*Өрнектер тізім 1i*; логӨрнек; *Өрнектер тізімі* 2) *оператор*

Цикл орындалмас *өрнектер тізімі 1* орындалады. Олар солдан оңға қарай орындалады.

Сосын логикалық өрнек тексеріледі. Егер ол ақиқат, true, болса оператор орындалады, сосын *Өрнектер тізімі 2.* Тағы логикалық өрнек тексеріледі. Егер ол ақиқат, true, болса оператор орындалады, сосын *Өрнектер тізімі 2* осылай жалғаса береді. Логикалық өрнек жалған, яғни false болса циклдің орындалуы аяқталады.

Тізім өрнек1; while *(логӨрнек){*

оператор

тізім Өрнек 2; }

for операторының кез келген бөлігі болмауы мүмкін: цикл бос болады, бірақ нүктелі үтір сақталады. Ақырсыз циклді беруге болады:

for (;;) оператор

Бұл жағдайда цикл денесінен шығу жолын қарастыру керек.

Мысал, бағдарлама бөлігі

```
int s=0;
for (int k = 1; k <= N; k++) s += k * k;
// k айнымалысы белгісіз
```

Бірінші N натурал санның квадратының қосындысын есептейді.

continue операторы және белгілер (меткалар)

continue операторы тек цикл операторларында ғана қолданылады. Ол екі түрлі тұлғасы (формасы) бар. Бірінші тек continue сөзінен тұрады және циклдің келесі итерациясына өткізеді. Мына жағдайда continue операторы нөлге бөлуден өткізіп жібереді:

```
for (int i = 0; i < N; i++){
if (i== j) continue;
s += 1.0 / (i - j);
}
```

Екінші тұлғасы белгіден тұрады:

continue таңба

таңба басқа идентификаторлар сияқты жазылады. Таңба оператор алдына қойылады немесе ашылған фигуралық жақшамен қос нүкте арқылы бөлінеді.

break операторы

break операторы цикл операторларынан, таңдау операторларынан және белгіленген блоктардан шығу үшін қолданылады.

Жазылуы: break таңба

Схема:

MI: { // Сыртқы блок

M2: { // Енгізілген блок — екінші деңгей

M3: { // Енгізудің үшінші деңгейі...

if *(бір өрнек болды)* break M2;

// Егер true болса, онда ештеме орындалмайды

}

// Бұнда да ештеме орындалмайды

}

// Бұнда басқару беріледі

}