Java-ның интегралданған орталары. Мәліметтердің тұрғызылған типтері,оларға қолданылатын операциялар. Басқарушы операторлар. Java-дағы қарапайым қосымшалар. Мәліметтерді енгізу мен шығару.Файлға мәліметтерді енгізу мен шығару.Массивтер. Массивтерді өңдеу. Кластар. Кластарды қолдану.

- Java тілінің негізгі жетістіктерінің бірі болып, тек қана статикалық мәтіндермен және графикамен ғана емес, сондай-ақ әртүрлі динамикалық объектілермен де жұмыс істеуі болып табылады. Java тілінде программалаудың ең маңызды ерекшелігі қолданушы компьютерінің типіне тәуелді еместігі болып табылады.

- Java тілінде жазылған программа бірден қандай да бір нақты процессор командаларына емес, Java виртуалды машинасының командалары деп аталатын машиналық командаларға жинақталады. *Java виртуалды машинасы* дегеніміз – бұл жүйемен бірге командалардың орындалу кодтарының жиынтығы.

Бірінші сатыда Java тілінде жазылған программа компилятор арқылы байт-кодтарға аударылады.Бұл компиляция қандай да бір нақты процессордың және қандай да бір нақты компьютердің архитектурасының типінен тәуелді болмайды. Ел программа жазылып болғаннан кейін бірден бір рет қана орындалады.

- Байт-кодтар бір немесе бірнеше файлдарда жазылады,сыртқы жадыда сақталуы мүмкін немесе желі арқылы беріледі. Бұл әсіресе шағын көлемді байт-кодтары бар файлдар үшін қолайлы. Одан кейін компиляция нәтижесінде алынған байт-кодтардыJVM-ді іске асыратын жүйесі бар кез келген компьютерде орындауға болады.Бұл кезде процессор типі де компьютер архитектурасы да маңызды емес.

- Қарапайым программаға мысал келтірейік.
- Экранға мынадай хабарлама шығару: «Java программалау тілін үйренеміз»

- class HelloWorld{

- public static void main(String[] args){

- System.out.println (“Java программалау тілін үйренеміз”);

- }

- }

- 1. Барлық программа бір немесе бірнеше кластардан тұрады,берілген қарапайым мысалда тек бір ғана *класс* (class) бар.
- Кластың басы class қызметші сөзімен белгіленеді, одан кейін кластың аты беріледі, ол еркін түрде таңдалады, берілген жағдайда Helloworld. Кластың құрамы түгелдей ирек жақша ішіне жазылады және *кластың денесін* құрайды.

- Барлық әрекеттер ақпараттарды өңдеу әдістерінің көмегімен орындалады,қысқаша жәй *әдіс* деп айтады. Әдістер аттарымен ерекшеленеді. Әдістердің біреуі міндетті түрде main деп аталуы тиіс, программаның орындалуы осыдан басталады.

- Әдіс нәтижесінде әрқашанда бір ғана мән береді, оның типі міндетті түрде әдіс атының алдында көрсетіледі. Сондай-ақ әдіс біздің жағдайдағыдай процедура рөлін атқарып, ешқандай мәнді қайтармауы да мүмкін.Сонда қайтарылатын мән типінің орнынаvoid сөзі жазылады.

- Әдіс атынан кейін жақшада,үтір арқылы, аргументтер немесе әдіс параметрлері тізбектеліп жазылады. Әр аргумент үшін оның типі көрсетіледі де бос орын арқылы аты жазылады.
- Әдіспен қайтарылатын мәннің типі алдында *модификаторлар* жазылуы мүмкін. Public сөзі - бұл әдісті кез-келген жерден қолдана беруге болатынын білдіреді;static сөзі main ()
- Әдісін программаның ең бастапқы орындалуы кезіндешақыруғамүмкіндік. береді
- Модификаторлар жалпы міндетті емес, бірақ main () әдісі үшін олар қажет.
- Берілген аттың жәй айнымалының емес, әдіс аты екендігін ерекшелеп көрсету үшін әдіс
- атынан кейін жақша қойылады.

- 7. *Әдіс денесі,* оның барлық құрамы ирек жақшаға алынып жазылады.
- Мысалда main () әдісінің орындайтын жалғыз әрекеті System.out.println күрделі аты бар басқа әдісті шақырумен және оған өңдеу үшін бір аргументті, «Java программалау тілін үйренеміз!» мәтіндік тұрақтыны берумен шектеледі.
- Мәтіндік тұрақтылар тек қана шектеушілер болып табылатын және мәтін құрамына кірмейтін тырнақшаларға алынып жазылады.
-

- System.out.println құрамасты Java API-ға кіретін System класында println() әдісі бар
- PrintStream класының, Java API кластарының бір данасынан тұратын, out атты айнымалымен анықталатынын білдіреді. Println() әдісінің әрекеті әдетте мәтіндік терминалды экранға шығарумен байланысты өзінің аргументін шығатын ағынға шығарумен шектеледі.Java тілі кіші және үлкен әріптерді айырады,main, Main, MAIN аттары Java компиляторының “көзқарасы ” бойынша әртүрлі. Мысалдарда String, System міндетті түрде бас әріппен жазылу керек, ал main кіші әріппен жазылады.

- Біздің мысалда программаны ағымдағы каталогтаWorld.java атты файлда сақтау қажет. Содан соң файлдың атын аргумент түрінде бере отырып, компилятор шақырылады:
- javac HelloWorld.java
- Компилятор байт-кодпен файл құрып,оған Helloworld.class атын береді де, осы файлды ағымдағы каталогқа жазады.
- Одан кейін оған аргумент ретінде класс атын беріп (файлды емес), интерпретаторды шақыру керек:
- Java HelloWorld
- Экранда пайда болады:

- Java тілінде әртүрлі типтегі тұрақтыларды әр түрде жазуға болады:
- бүтін;
- нақты;
- Нақты тұрақтылар тек қана ондық санау жүйесінде екі түрде жазылады:
- бекітілген нүктемен: 37.25, -128.678967, +27.035;
- жылжымалы нүктемен: 2.5e34, -0.345e-25, 37.2E+4; E латын әріпін бас әріппен немесе кішкене әріппен жазуға болады; бос орын мен жақша қолданылмайды.

- Нақты тұрақтының соңындаF немесе f әріпін қоюға болады, онда тұрақты float типті форматта сақталады (төменді қара): 3.5f, -45.67F, 4.7e-5f. D (немесе d) әріпін жазуға да болады: 0.045D, -456.77889d, бұл double типін білдіреді, бірақ бұл артық болады, өйткені нақты тұрақтылардың өзі де double типті форматта сақталады.
- Жеке белгілерді жазу үшін келесі формалар қолданылады: Баспа белгілерді апострофта жазуға болады: 'a', 'N', '?'.

- Басқарушы белгілер апострофта кері еңкейтілген сызықпен жазылады: '\n' — ASCII 10 кодымен newline жолын аудару белгісі;
- '\r' — 13 кодымен CR арбаны (каретки) қайтару белгісі,;
- '\f' — 12 кодымен FF парағын аудару белгісі,;
- '\b' — бір қадамға қайтару белгісі BS, 8 кодымен; '\t' — көлденең табуляция белгісі HT, 9 кодымен; '\\' — кері еңкейтілген сызық;
- '\"' — тырнақша;
- '\'' — апостроф.

- Кез келген 0 ден 255 дейінгі ондықкодировкасы бар белгінің кодын,сегіздік санау
- жүйесіндеүш цифрдан асырмай кері еңкейген сызықтан кейін апостроф ішінде жазуға
- болады:
- '\123' —S әріпі,
- '\346' —Жәріпі CP1251 кодировкасында. Баспа және басқарушы
- белгілер үшін бұл форма қолданылмайды, өйткені алдыңғы пунктте көрсетілгендей,
- компилятор бірден сегіздік сандау жүйесіндегі жазбаны жоғарыда көрсетілген формаға аударады.
- Ең үлкен код '\377' — ондық сан 255.

- Unicode кодировкасындағы кез келген символдың коды латынның әріпімен төрт оналты орынды цифрмен:
- '\u0053' —S әріпі,
- '\u0416' —Ж әріпі және кері еңкейтілген сызықтан кейін апострофта жазылады. Символдар char типті форматта сақталады.
- Символдар қатары тырнақшаға алынады. Басқарушы символдар мен кодтар да дәл осылай кері еңкейген сызықпен, жолдарға жазылады. Жолдар бастапқы кодтың бір қатарында орналасуы мүмкін, ашылған тырнақшаны бір қатарға,ал жабылатын тырнақшаны келесі қатарға жазуға болмайды.

- Жолдық тұрақтылар үшін плюспен белгіленетін жалғасу операциялары анықталған.

  "Жалғасу " + "қатар" нәтижесінде "Жалғасу қатары" қатарын береді.
- Ұзын қатарды бір қатарлы тұрақтылар түрінде жазу үшін, жабу тырнақшасынан кейін бірінші және келесі жолдарға қосу (+) қою керек, сонда компилятор екі (немесе бірнеше) қатарды бір қатарлы тұрақтыға жинайды, мысалы:

  "Бір қатарлы тұрақты, жазылған "+ "бастапқы мәтіннің екі қатарында"

- Java тілінде құрылған, бастапқы мәліметтердің барлық типтері екі топқа бөлінеді:

- *примитивті типтер* және *сілтемелі типтер.*
- Сілтемелі типтер *массивтерге* (arrays), *кластарға* (classes) және *интерфейстер* (interfaces)
- болып бөлінеді.
- Примитивті типтер барлығы сегіз.Оларды *логикалық* тип boolean және *сандық* деп бөлуге болады.
- Сандық типтерге бүтін: byte, short, int, long, char жатады.

- **Бүтін типтілерге қолданылатын операциялар**
- Арифметикалық амалдарға жататындар:
- қосу + (қосу);
- азайту – (сызықша);
- көбейту * (жұлдызша);
- бөлу / (еңкейтілген сызық- слэш);
- бөлгеннен қалған қалдықты алу (модуль бойынша бөлу) % (процент);
- инкремент (бірге арттыру) ++;
- декремент (бірге кеміту) --

- **Нақты типтер** Java-да екеу: float және double.
- Стандартты арифметикалық және тригонометриялық функцияларды орындауJava тілінде арнайы Math класының көмегімен жүзеге асырылады. Ол әр программада автоматты түрде берілетін Java.lang пакетіне жатады. Math класының кейбір әдістерін келтіреміз:
- Final double PI
- double pow (double, double) double sqrtr (double)
- double atan (double)
- double cos|sin|tan (double) // радианмен double random ()
- int round (float) long round (double) value abs (value)
- value max (value, value) value min (value, value)

- Екі кездейсоқ шамалардың арасынан max әдісін қолдана отырып, ең үлкен санды табу туралы мысал:
- Class pr{ Pr()
- {double random1, random2;
  random1=Math.random(); random2=Math.random();
  System.out.println("қосынды"+random1+random2);
- System.ou.println("max"+Math.max(random1,random2));
- }
- public static void main (string[] args) { new pr();
- }
- }

Java тіліндегі негізгі операторлар жинағы:

- айнымалы және басқа да объектілерді сипаттау операторлары;
- өрнектеу операторлары;
- меншіктеу операторлары;
- If шартты операторы;
- while, do-while, for үш циклдік операторлары; switch нұсқаулы операторы;
- break, continue және return өту операторлары; {}-блогы; бос оператор – жәй нүктелі үтір.
- Мұнда Java операторлар жинағы толық келтірілмеген, ол тілді оқып-үйрену барысында толықтырылады.
- Java тілінде goto операторы болмайды.
- Кез келген оператор нүктелі үтірмен аяқталады.

Меншіктеу операциясы

- теңдік белгісімен (=) белгіленеді, оның сол жағында айнымалы тұрса, оң жағында айнымалының типімен бірігетін өрнек тұрады:
- x = 3.5, y = 2 * (x - 0.567) / (x + 2), b = x < y, bb = x >= y && b.
- Блокта тіл ережесі бойынша тек бір ғана оператор жазылатын орындарда, ноль немесе бірнеше операторлар енгізе алады. Мысалы, {x = 5; y = ?;}. Бос блок жазуға да болады, ол үшін қос өрнекті жақша {} аламыз.
- Операторлар блогі айнымалылар әрекеттерінің шектелу облысы үшін және программа мәтінінің оқылуын жақсарту үшін қолданылады.

- Шартты оператор (if-then-else statement) Java тілінде келесі түрде жазылады: if (логӨрнек) then оператор1 else оператор2
- және келесі түрде орындалады.Алдымен логикалық өрнек(логӨрнек)есептеледі. Егер
- шешімі true болса, онда *оператор1* орындалады
- және осымен шартты оператор жұмысын тоқтатады, *оператор2* әрекет
- етпейді, ары қарай if операторынан кейінгі оператор
- орындалады.
- Егер шешімі false
- болса,
- онда
- *оператор2*

- Шартты оператор қысқаша *if (логӨрнек) оператор*
- болады және false жағдайында ешқандай жұмыс атқарылмайды.
- Тіл синтаксисі then және else бұтақтарына бірнеше оператор жазуға тиым салады. Қажет жағдайда ирек жақшалардағы блок операторлары құрылады.
- Шартты операция үш операндадан тұрады. Алдымен, нәтижесінде true немесе false мәндерін қабылдайтын еркін логикалық өрнек жазылады, содан кейін сұрақ белгісі қойылады, содан соң қос нүктемен ажыратылған екі еркін өрнек келтіріледі, мысалы,
-
- x< 0 ? 0 : x
-
- x> y ? x — y : x + y

- Мысал. Квадратты теңдеудің түбірлерін табу class Kwur1 {
-
- public static void main(String[] args) { double a=1,b=2,c=3,d,d1,a1; d=b*b-4*a*c;

- if (d<0) { d1=0.5*Math.sqrt(-d)/a; a1=-0.5*b/a;

- System.out.println("x1="+a1+"+i"+d1+", x2="+a1+"-i"+d1); } else {
- d=0.5*Math.sqrt(d)/a; a=-0.5*b/a;
- System.out.println("x1="+(a+d)+",x2="+(a-d));

- }
- }
- }

- Циклдің негізгі операторы – while операторы келесі түрде болады: while (логӨрнек) оператор
- Алдымен, *логӨрнек* логикалық өрнегі есептелінеді; егер оның мәні true болса, онда циклді құрайтын оператор орындалады.Одан кейін қайтадан*логӨрнек* есептеледі және опеаратор әрекет етеді, яғни ол false мәні шыққанша орындалады. Егер *логӨрнек* басында false-қа тең болса, онда *оператор* бір рет те орындалмайды. Алдын ала тексеру циклдің орындалуының қауіпсіздігін қамтамасыз етеді, толып кетуден, нольге бөлуден және басқа да қолайсыздықтардан сақтайды.

- Сондықтан while операторы негізгі болып табылады, ал кейбір тілдерде ол тіптен жалғыз циклдік оператор болып табылады.
- Циклдегі оператор бос болуы мүмкін, мысалы, келесі код бөлігі: int i = 0;
- double s = 0.0;
- while ((s += 1.0 / ++i) < 10);
- s гармоникалық қосындысы 10 мәніне жететіндей і көбейтіндінің санын есептейді.

- Шексіз цикл ұйымдастыруға да болады:
- while (true) *оператор*
- Егер циклге бірнеше оператор жазу керек болса,онда операторлар блогін {} қолдануға болады.
- Циклдің екінші операторы - do-while операторы келесі түрде болады: do *оператор* while *(логӨрн).* Мұнда ең алдымен*оператор* орындалып, содан кейін логикалық өрнек*логӨрн* есептеледі. Цикл *логӨрн* true-ге тең болғанға дейін орындалады.

- Continue операторы тек цикл операторларында қолданылады.Ол екі түрге ие. Бірінші түрі тек қанаcontinue сөзінен тұрады және жедел түрде келесі цикл итерациясына өтуді жүзеге асырады. Код үзіндісінде continue операторы нольге бөлуден сақтандырады:
-
- for (int i = 0; i < N; i++){ if (i '== j) continue;
- s += 1.0 / (i - j);
- }

- Break операторы цикл операторларында және осы конструкциялардан жылдам шығу үшін нұсқа операторында қолданылады.
- Нұсқа операторы switch бірнеше бағыт бойынша тармақталуды қамтамасыз етеді.Әр тармақ тұрақтымен немесе қандай да бір бүтін типтің(long типінен басқа) тұрақтылық өрнегімен белгіленеді және таңдалады. Барлық конструкция мына түрде болады:
- switch (БүтінӨрн){
- case констӨрн1: оператор1 case констӨрн2: оператор2
-
- . . . . .
-
- case констӨрнN: операторN
-
- default: операторDef
-
- }
-

- Жақшаға алынған *констөрнек* өрнегі byte, short, int, char типтерін қабылдайды, бірақ long типі емес.
- Тұрақтылардан құралған бүтінсанды өрнектер немесе бүтін сандар
- *Конст*
- *Өрнек*
- long
- типті бола алмайды.

- 1. Java тілінде программалаудың ерекшеліктерін атаңыз.
- Java-программаның орындалу реті қандай?
- Шартты оператор формасы қандай?
- Цикл операторының формасы қандай?
- break операторын белгімен нұсқа операторында қолдануға бола ма?