Tartu Ülikool

Arvutiteaduse Instituut

JAVA LEKSIKON

Tarkvarasüsteemide seminari töö

Leksikoni artiklite autoriteks on üliõpilased:

Jelena Fjodorova [JF]
Dmitri Brodski [DB]
Kaido Hallik [KH]
Gunnar Kudrjavets [GK]
Urmet Liin [UL]
Tambet Matiisen [TM]
Alar Pandis [AP]
Aleksei Podlipski [APo]
Margit Reinkubjas [MR]
Jaanis Sulg [JS]
Mark Tehver [MT]
Vallo Trell [VT]
Andres Tsengov [AT]
Toomas Viiraslu [TV]

Seminari juhendaja: **J.Kiho**

Tartu 1997

Saateks

Käesolev leksikon on valminud *Tarkvarasüsteemide* seminari tööna 1996/97 õppeaasta kevadsemestril. Seminarist osavõtjatel tuli esineda ettekandega ettenähtud teemal, tuvastada antud teemas käsitletud peamised mõisted ja koostada neid kirjeldavad artiklid. Tulemus on järgnevas esitatud peaaegu muutmata kujul; minu poolt on tehtud vaid pisut hädavajalikku "kosmeetikat".

Koostatud leksikon on ebatäielik (mõned teemad jäid hoopis katmata vastavate seminaristide loobumise tõttu) ning artiklite kvaliteedilt ebaühtlane, peegeldades ka erinevate üliõpilaste võimete- ja isikupäraseid vaateid *JAVA*-le. Siiski on tehtud mahukas töö, mida saab edaspidi võtta aluseks täiuslikuma eestikeelse leksikoni loomisel.

Kuna eestikeelne terminoloogia on veel välja kujunemata, siis rakendasime paindlikku süsteemi, mille kohaselt on põhitekstis eestikeelsete terminte asemel kasutatud (inglisekeelseid) makrosid. Mingi termini asendamisel uuega tuleb lihtsalt teha muudatus makrode tabelis ja kompileerida uuesti põhitekst.

Seminari töö aluseks oli õpik:
P.Niemeyer, J.Peck. *Exploring JAVA*. O'Reilly, 1996, 407 lk.

Seminari juhendaja

absoluutne positsioon, *absolute position* – objekti asukoht on seotud ekraani koordinaattelgedega, paigutus ei sõltu objekti esiletoova rakendi asukohast. [MR]

abstraktne klass, *abstract class* – klass, mis sisaldab ainult meetodite deklaratsioonid ilma kirjeldusteta. [APo]

abstraktne meetod, *abstract method* – meetod ilma kehata, s.t. koosneb ainult deklaratsioonist. [APo]

abstraktsed aknavahendid, *Abstract Windowing Toolkit* – graafilise kasutajaliidese nn. Java variant. Sisaldab kõike, mis ekraanil on nähtav. Üks Java põhielemente. AWT abil saab luua aknaid, lõuendeid, tegeleda piltidega, samuti kasutada selliseid aknakomponente nagu nupud, kerimisribad jms.; klassid, mis hõlmavad AWT, on osa java.awt paketist. Vt. ka *graafiline kasutajaliides.* [AP]

alamklass, *subclass* – alamklass on mingi klassi sees defineertud klass. alamklassil on samad avalikud meetodid ja muutujad nagu klassil, kus ta defineeritud on. Muutujaid ja meetodeid on võimalik üle defineerida. Vt. ka *päritavus, ülemklass, abstraktne klass, klass, üledefineerimine, ülekatmine.* [AT]

alamliides, *subinterface* – liides, mis sisaldub teises liideses. Vt. ka *liides.* [APo]

alammenüü, *submenu* – menüüvalik, mis on ise menüü klassi kuuluv objekt. [MR]

alamtüüp, *supertype* – tüüp, mis sisaldab teise tüübi. [APo]

algtüüp, *primitive type* – alg- või keeletüüp, defineeritakse keele sees. [DB]

arenduskeskkond, *development environment* – programmide kompleks, mis võimaldab Java programme kompileerida, käivitada ja siluda. [MT]

arendusvahendite komplekt, *development kit* – vahendite komplekt programmide kirjutamiseks, käivitamiseks ja silumiseks. Näiteks JDK. [MT]

avalik muutuja, *public variable* – muutuja, mis omab kõige nõrgemat piiratud juurdepääsu. Vt. ka *piiratud juurdepääs.*

avardamine, *extention* – ülemklassi omaduste, muutujate ja meetodite pärimine ja avardamine almklassis. Vt. ka *päritavus.* [DB]

baastüüp, *base type* – antud massiivi elementide tüüp. Java toetab arvulisi, loogilisi ja viidatüüpi massiive. Vt. ka *massiiv.* [JF]

baitkood, *byte-code* – arvuti arhitektuurist sõltumatu, kompileeritud Java kood. baitkoodis olevat programmi täidab Java virtuaalarvuti. Sünon. *J-kood.* Vt. ka *virtuaalarvuti.* [TV]

baitkoodi verifikaator, *byte code verifier* – võimaldab kontrollida Java klassifailide ohutust. Tavaliselt kasutatakse klassifailide puhul, mis loetakse teistest arvutitest võrgu kaudu. Vt. ka *klassifail.* [MT]

datagramm, *datagramm* – andmeühik, mida kasutatakse ühendusteta protokolli juures. Seda kasutab ka näiteks *UDP* protokoll. [GK]

deemon, *daemon* – lõim, mille ainuke eesmärk on teiste lõimede teenindamine. Programmi töö lõpetatakse automaatselt siis, kui töötavateks lõimedeks on ainult deemonlõimed. Vt. ka *lõim.* [VT]

dialoog, *dialog* – Ülataseme aken, mis võib olla modaalne, s.t. kogub kokku kõik sisestatud sündmused seni, kuni saab rahuldatud. [MR]

dünaamiline sidumine, *dynamic binding* – Java dünaamiliselt (käitusajal) otsustab, mille mitmetest ülalkatetud meetoditest kasutada. [APo]

edasilüke, *throw* – kasutatakse meie poolt loodud erindite edasilüketks. [JF]

eeldefineeritud, *predefined* – muutuja või parameeter, mille Java väärtustab programmeerija eest ise. [APo]

eeljoonistamine, *off-screen drawing* – ekraanile kuvatava pildi (või selle osa) joonistamine →eeljoonistuspuhvrisse. On kasutusel esituse sujuvamaks ning kiiremaks muutmisel. [UL]

ekasemplari meetod, *instance method* – meetod, mida võib kutsuda välja ainult klassi eksemplariga koos.Vt. ka *staatiline meetod.* [DB]

erind, *exception* – inditseerib ebatavalise või vigatingimuse ja on esitatud Java objektina (erindiklassi isendina), mis luuakse koodina punktis, kus ilmus vigane tingimus. Vt. ka *erindiklass.* [JF]

erindiklass, *exception class* – Java API erindiklass esitab erindi. Erinevad erindi alamklassid säilitavad infot mitmete probleemide kohta. [JF]

fondimeetrika, *font metrics* – on kasutusel fondi suuruse- ning muude mõõtmetega opereerimiseks. Siin hoitakse infot selle kohta, millised on antud fondi tähtede mõõtmed ning ridadevaheline kaugus. Samuti ka selle kohta, palju mingi fondi tähed üleval- ning allpool baasjoont ruumi võtavad. Vt. ka *font.* [UL]

font, *font* – määrab ära kirjastiili, selle suuruse ning muud atribuudid. Samas on samanimelises klassis veel rida vahendeid erineva kirjastiile puudutava informatsiooni töötlemiseks. Sünon. *kirjastiil, kiri.* [UL]

fookus, *focus* – fookuses olev - hetkel esitamiseks või sisestamiseks määratud aken. Sündmuste edastamiseks peab komponent olema aktiivne (fookuses). Aktiivne komponent on käesolev nö. sisendkomponent (ongi teiste sõnadega "aktiivne komponent"). Temale edastatakse kõik klaviatuuri poolt tektitatud sündmused senikaua, kui fookus kaob (deaktiveeritakse). Vt. ka *sündmus.* Sünon. *aktiivaken.* [AP]

FTP, *FTP (File Transfer Protocol)* – protokoll, mis võimaldab edastada faile võrgu kaudu ühest arvutist teise. Vt. ka *protokoll.* [TM]

graafikakontekst, *graphics context* – joonistatav pind, milles säilitatakse kontekstuaalset informatsiooni joonistatava ala kohta ja pakutakse meetodeid sellel alal joonistamisoperatsioonide teostamiseks. [UL]

graafiline kasutajaliides, *Graphics User Interface* – vahendite kogum kasutajaliidese loomiseks. Vt. ka *abstraktsed aknavahendid.* [AP]

host, *host* – arvuti, mis võimaldab kasutajail suhelda teiste võrgus olevate arvutitega. Sünon. *võrguarvuti.* [TM]

hävitamine, *terminating* – mingi protsessi või lõime lõpetmine. [DB]

HTML, *HTML (HyperText Mark-up Language)* – markeerimiskeel, mida kasutatakse hüpertekstidokumentide loomisel. HTML koosneb märgistest, mis esitavad hüperteksti erinevaid osi, näiteks kirjastiile ja viiteid. [TM]

HTTP, *HTTP (HyperText Transfer Protocol)* – protokoll, mis võimaldab ligipääsu teises arvutis asuvatele hüpertekstidokumentidele. Vt. ka *protokoll, HTML.* [TM]

indeksi rajaerind, *index out of bound exception* – tekib siis, kui püütakse pääseda elemendile, mis ei kuulu antud massiivi; indeksi rajaerind kuulub käitusaegse erindi hulka.Vt. ka *massiiv, käitusaegne erind.* [JF]

interpretaator, *interpreter* – realiseerib Java virtuaalarvuti ja käitab Java rakendust. [MT]

isend, *instance* – klassi järgi vormitud objekt. Klassi isendil on samad muutujad ja meetodid, mis klassis defineeritud. Ühest klassist võib töö ajal tekitada mitu isendit. Ühe klassi isendid võivad olla erinevates olekutes, kuid neil on alati samad muutujad ja meetodid. Sünon. *eksemplar.* Vt. ka *klass.* [KH]

isendmuutuja, *instance variable* – muutuja, mis on defineeritud klassi sees, kuid mitte ühegi meetodi sees; isendmuutuja kuulub kogu klassile, mitte mingile konkreetsele meetodile. Selle muutuja koopia ilmub igasse antud klassi isendisse ning seda saavad kasutada kõik antud klassi meetodid. Olenevalt selle muutujate piiritlejast võib saada seda kasutada ka väljastpoolt klassi. [KH]

jooksev klass, *current class* – klass, mis sisaldab antud momendil täidetava meetodi. [APo]

joonistusmeetod, *drawing method* – kirjeldab erinevaid pildi kujutamise võimalusi. Ekraanile on võimalik tuua pilte kas tervet joonistatavat ala üle joonistades, seda ala piirates ja ikkagi üle joonistades või siis joonistades eeljoonistamist kasutades. Vt. ka *eeljoonistamine, topeltpuhverdus.* [UL]

juhuslik, *random* – mehhanism juhuslike arvude saamiseks, kas siis mingi kindla lõigu piires või suvalises vahemikus. [JS]

juurdepääsu piiritleja, *access modifier* – reserveeritud sõna Javas, mis määrab meetodi või muutuja skoobi. Vt. ka *skoop* [APo]

J-kood, *J-code* – Sünon. *baitkood.* Vt. ka *baitkood.* [TV]

Java, *Java* – võrguprogrammeerimise jaoks loodud programmeerimiskeel.

Arvestab võrgu iseärasustega, pakub mitmeid võimalusi turvaliste, platvormist mitte sõltuvate rakenduste loomiseks. Esmane arendus toimus firmas Sun Microsystems James Gosling'i ja Bill Joy käe all. [TV]

Javaskript, *Java script* – Netscape'i poolt loodud HTML-keele laiendus, lihtne programmeerimiskeel. Mitteprogrammeerijatele Javast lihtsamini mõistetav ja kasutatav. Javaskriptiloomisel oli aluseks Java, kuid siiski on need kaks keelt teineteisest erinevad. [TV]

kaartpaigutus, *cardlayout* – kujundusjuht, mis asetab objektid konteineri sees "kaartidena kuhja". Korraga näidatakse ainult ühte, pealmist objekti. objektide lisamisel antakse ette objekt ja sellele vastava kaardi nimi. Vt. ka *kujundusjuht* [MR]

kaitstud, *protected* – piiratudjuurdepääsuidentifikaator. Määrab niisugused muutujad (või meetodid), mida võib kasutada ainult klassi sees. Vt. ka *piiratud juurdepääs,avalik ja privaatne.* [DB]

kapseldus, *encapsulation* – andmete esituse ja andmetega tehtavate tegevuste realiseerimist puudutavate detailide peitmine kasutaja eest. Objektorienteeritud programmeerimise üks põhilisi mõisteid. [TV]

katse, *try* – algatab erinditöötlemise protsessi. Siin erind ilmub ja saadetakse järgnevale püünis-direktiivile edaspidiseks käsitsemiseks. Vt. ka *püünis, erind, käsitsemine.* [JF]

kerimisriba, *scrollbar* – Standartne kerimisriba, võib olla nii vertikaalne kui horisontaalne (antakse ette konstruktoris). [MR]

keskkonnamuutuja, *environment variable* – operatsioonisüsteemi poolt kontrollitav string-tüüpi muutuja. [MT]

klass, *class* – grupp andmeelemente koos funktsioonidega, mis teostavad operatsioone nende andmetega. Klassi andmeelemente nimetatakse väljadeks või muutujateks, funktsioone aga meetoditeks. Vt. ka *andmeelement, väli, muutuja, meetod.* [KH]

klassi importimine, *class importing* – klassi sisseehitamine programmi direktiiviga import. Vt. ka *klassi otsimistee* [APo]

klassi eksemplar, *class instance* – väärtus muutujas, mille tüübiks on klass, omab iga klassi muutuja oma eri koopia. [DB]

klassi otsimistee, *class qualifier* – klassi leidmiseks Java pärimishierarhias. Selleks, et kasutada oma programmis mingit klassi, tuleb kirjutada selle klassi otsimistee või teha klassi importimine. Vt. ka *klassi importimine.* [APo]

klassi struktuur, *class structure* – abstraktne tüüp, mis defineerib klassi üldise kuju. Vt. ka *klass ja klassi eksemplar.* [DB]

klassifail, *classfile* – Java baitkoodi sisaldav fail, mis vastab ühele klassile. [MT]

klassitee, *class path* – määrab lokaalse arvuti kataloogid ja arhiivifailid, kust Java käitusaegne süsteem klassifaile otsib. Tavaliselt määratakse see keskkonnamuutuja abil. Vt. ka *käitusaegne süsteem, klassifail, keskkonnamuutuja.* [MT]

klient, *client* – kui on tegu võrgurakendusega, siis räägitakse klientidest ja serveritest. Tavaliselt mõistetakse kliendi all seda poolt, kes "vestlust" alustab ja serveri all seda poolt, kes "vestlusnõuet" aktsepteerib. Iga klient võib samal ajal olla ka server ja vastupidi. [GK]

klippimine, *clipping* – joonistatava ala piiramine graafikatöö muutmisel kiiremaks ning sujuvamaks. [UL]

kompilaator, *compiler* – Java kompilaator transleerib Java programmid klassifailideks. Vt. ka *klassifail.* [MT]

kompileerimisühik, *compilation unit* – fail laiendusega *.java*, mis sisaldab ühe (või mitme) Java-klassi lähtekoodi. [APo]

komponent, *component* – abstraktsete aknavahendite ekraanielement. Põhiline kasutajaliidese objekt Javas. Kõik, mis me näeme ekraanil Java rakendustes, on abstraktsete aknavahendite komponent. See sisaldab aknaid, lõuendeid, nuppe, nimekirjasid, tekstiväljasid. Komponendi kasutamiseks paigutatakse ta konteinerisse. Konteinerid grupeerivad komponente, paigutavad nad ekraanil ning vahetevahel seovad neid erinevate ekraaniseadmetega. Kõik Java komponendid on pärit abstraktse klassist nimega java.awt.Component. Vt. ka *abstraktsed aknavahendid, nupp, lõuend, konteiner, nimekiri, aken.* [AP]

kompositsioon, *composition* – juba olemasolevate klasside kasutamine uue klassi sees. [KH]

konstruktor, *constructor* – spetsiaalne meetod, mis kutsutakse välja uue klassi eksemplari loomisel viimase muutujate initsialiseerimiseks. konstruktoril on klassiga sama nimi. Näide - theButton = new Button(”Start”); Ühel klassil võib olla mitu konstruktorit - neil tehakse vahet argumentide järgi. Vt. ka *meetodi ülelaadimine, uus.* [AT]

konteiner, *container* – abstraktsete aknavahendite element, mis võimaldab komponente paigutada ja neid juhtida. Konteiner on ka ise komponent. Vt. ka *abstraktsed aknavahendid, lõuend, komponent, paneel.* Sünon. *sisaldaja.* [AP]

kontrollruudugrupp, *checkbox group* – kontrollruutudest moodustatud grupp, kus korraga on võimalik valida ainult ühe nupu. Vt. ka *kontrollruut* [MR]

kontrollruut, *checkbox* – Nimega valikunupp. Nupud võivad olla üksteist välistavad, kui kuuluvad kontrollruudugruppi. Vt. ka *kontrollruudugrupp* [MR]

korrapäratus, *transgression,irregularity* – mitmesuguseid töö, programmi käivitamise või reeglite korrapäratused. [DB]

kujundusjuht, *layout manager* – AWT kasutab kujundusjuhti komponentide paigutamiseks konteinerites ning nende suuruse ja positsiooni kontrolliks. Vt. ka *komponent, konteiner* [AP]

käitusaegne süsteem, *run-time system* – süsteem, mis suudab lugeda ja käitada Java klassifaile. [MT]

käitusliides, *runnable interface* – sisaldab meetodit run, mis →laetakse üle kasutava klassi poolt. Selle liidese kaasamine poolt muudab klassi lõimeklassiks, võimaldades luua antud klassi isendist →lõime. Alternatiivselt võiks lõimeklassi luua ka Java standardklassi Thread alamklassina. Vt. ka *lõimeklass.* [VT]

käsurea argument, *command line argument* – antakse ette rakendusele rakendust käsurealt käivitades. Vt. ka *rakendus.* [MT]

laiendatavus, *scalability* – programmeerimiskeele (ka programmi või algoritmi) omadus lähteülesande mahu kasvades sama lähenemisviisi kasutada. [TV]

lekseem, *token* – tavaliselt programmeerides võib meil vaja minna, et me tahame analüüsida mingit sõna.Siis seda saab teha lekseemite abil. Nad on ühesõnaga "eraldajad", mille abil me sealt näiteks alamsõne otsida. [JS]

loetelu, *enumeration* – mõeldud loetelude saamiseks mingi hulga elementidest, kusjuures seda saab rakendada ükskord korraga. Ei garanteeri meile mingis kindlas järjestuses loetelusid, vaid suvalisi, nii et selle eest tuleb programmeerijal ise hoolt kanda. [JS]

lokaalmuutuja, *local variable* – muutuja, mis on defineeritud mingi kindla meetodi sees. Seda muutujat näeb ainult meetodi seest ning ta elab ainult meetodi välja kutsumiseajal. [KH]

loomulik kood, *native code* – konkreetse arvutitüübi masinakood. Sünon. *omakood*. [TV]

lõim, *thread* – Java vahend mitmetegumilisuse realiseerimiseks. Iga lõim on vastava operatsioonisüsteemi üksikprotsess. Lõime tegevuse määrab lõimeklassi, mille isend ta on run meetodi sisu ja tema tegevust juhitakse vastava lõime kontrollivahenditega. Vt. ka *lõimeklass, lõime juhtimine.* [VT]

lõime juhtimine, *thread control* – lõimede töö juhtimiseks kasutatakse Java klassi Thread staatilisi meetodeid ja lõime konstruktorit, mille parameetriks antakse mingi →lõimeklass. Loomise järel on lõim alustatud, kuid käivitamata. lõime töö alustatakse kutsudes välja vastava klassi start meetod. Seejärel vastav lõim töötab, kuni toimub üks järgnevatest sündmustest

1. Kutsutakse välja lõimeklassi stop meetod. Selle järel vastav lõim lõpetab töö. Samasse seisu jõuab lõim ka siis, kui ta jõuab oma run meetodi lõppu.
2. Kutsutakse välja lõimeklassi sleep meetod. Lõime töö jätkub sellisel juhul parameetris antud aja möödumise järel.
3. Kutsutakse välja lõimeklassi suspend meetod. Tööd jätkatakse pärast sama objekti resume meetodi kutset.
4. Kutsutakse välja lõimeklassi wait meetod. Tööd jätkatakse siis, kui

vastav →monitor-objekt kutsub selle objekti notify meetodit, või teatab kõigile ootavatele lõimedele võimalusest tööd jätkata meetodi NotifyAll kaudu. Vt. ka *lõim* [VT]

lõime prioriteet, *thread priority* – lõimeolulisus täidetavate lõimede hulgas. Java lõimede prioriteedid antakse täisarvudena vahemikust 10 – 1 (Thread.MIN_PRIORITY – Thread.MAX_PRIORITY). Lõime loomisel antakse prioriteediks 5 (Thread.NORM_PRIORITY). Prioriteeti saab muuta lõimeklassi meetodiga setPriority, mille argumendiks on vastava lõime uus prioriteet. Vt. ka *lõim* [VT]

lõime sñkroniseerimine, *thread syncronization* – protsess, mis tagab lõimede sellise töö, et ükski üksiklõim ei sega ühegi teise tööd. Sünkroniseerimise otstarbeks on Javas olemas sünkroniseeritud meetodid ja monitor-objektid. Vt. ka *monitor-objekt.* [VT]

lõimeklass, *thread class* – klass, mille run meetodit saab käivitada uue lõimena. Selliseks klassiks on Java standardobjekti Thread →alamklass id või klassid, mis kaasavad käitusliidese. Lõimeklassi iga →isendit on võimalik käsitleda eraldiseisva lõimena ja kasutada nende töö juhtimiseks klassi Thread meetodeid. Vt. ka *lõim, käitusliides.* [VT]

lõimerühm, *thread group* – konstruktsioon lõimede haldamise lihtsustamiseks. Samasse lõimegruppi kuuluvatele lõimedele on võimalik juhtimismeetodeid rakendada korraga rakendades neid meetodeid vastavale lõimerühmale. Vt. ka *lõim, lõime juhtimine.* [VT]

lõpetamine, *finalization* – protsess, mis toimub suvalise meetodi lõpus; taastab ruumi ja teised ressursid. Vt. ka *taastumine.* [DB]

lõpuks-lause, *finally clause* – pannakse ühe või rohkem püünis direktiivi järele ja kasutatakse lõpetamiseks. [JF]

lõuend, *canvas* – joonistuste esituse ja joonistustega tehtavate tegevuste realiseerimise komponent abstraktsete aknavahendite koosseisus. Objektid lõuend ja paneel on puhtalt rahuldamaks joonistamise vajadusi. Iseenesest ei oma lõuend välist ilmutust; ta võtab ruumi ning omab tagapõhja värvi, muus osas on ta tühi. Lõuendi alamklassimisel ja enda koodi lisamisel võib luua keerukaid graafilise esitusi. Lõuend on samuti konteiner. Vt. ka *abstraktsed aknavahendid, komponent, konteiner, raam, pilt, paneel.* [AP]

massiiv, *array* – Spetsiaalne objektitüüp, kus saab säilitada sorteeritud elementide kogu; massiivi elementide tüüpi nimetatakse →baastüübiks, elementide arvu nimetatakse massiivipikkuseks. [JF]

massiivi kopeerimine, *array copy* – Java meetod, mis võimaldab ühe massiivi elemendid kopeerida teise massiivi. Vt. ka *massiiv.* [JF]

meetodi otsimise süsteem, *method resolution system* – Java-süsteem ülelaaditud või ülekatetud meetodite valimiseks. [APo]

meetodi signatuur, *method signature* – koosneb meetodi nimest, →nähtavuse piiritlejast, meetodi tüübist (klassi meetod või ekasemplari meetod), argumentide tüüpidest ja väljundtüübist. [MT]

menüü, *menu* – Standartne allatõmmatav menüü, millel on fikseeritud nimi. Hoiab endas menüüvaliku objekte. Võib sisaldada alammenüüsid. menüüd saab lisada ainult menüüribale. Vt. ka *menüüvalik, menüüriba* [MR]

menüüriba, *menubar* – Riba, millele asetatakse menüü objekt. menüüriba saab lsiada ainult raamile või teisele menüüle (kui soovitakse luua alammenüüd). Vt. ka *menüü, alammenüü* [MR]

menüüvalik, *menu item* – Üks menüüs pakutavatest valikutest. [MR]

mitmemõõtmeline massiiv, *multidimensional array* – massiiv, mis koosneb massiivi tüüpi objektiidest. Vt. ka *massiiv, objekt.* [JF]

mitmene päritavus, *multiple inheritance* – päritavuse viis, millal tuletatud klass on moodustatud mitmetest klassidest päritavuse teel. [APo]

monitor-objekt, *monitor object* – klass, mida samaaegselt kasutab mitu erinevat lõime. monitor-objekti (sünkroniseeritud) meetod jälgib lõimede tegevust ja peatab või taastab lõimede tegevuse vastavalt vajadusele. Kasutatakse lõimede töö sünkroniseerimiseks. Vt. ka *lõime snkroniseerimine.* [VT]

muutuja, *variable* – klassi andmeelement; muutuja võib sisaldada standardset tüüpi andmeid (*integer, float, boolean* jne.) või olla osuti objektile. [KH]

muutumatu väärtus, *constant value* – väärtus, mis ei või muutuda programmi täitmise käigus. [DB]

mähisklass, *wrapper class* – klass, mille ülesanne on ühte tüüpi klassi objekte teha teist tüüpi klassi objektideks. maähisklassi näidis kapseldab väärtuse talle vastavaks tüübiks. mähisklassmäärab meetodi *valueOf()*. Näiteks, kui me tahame string objektist teha numbrit, siis pole see enam String klassi ülesanne vaid seda teeb mähisklass. [JS]

mälukoristus, *garbage collection* – mälukoristaja vabastab mälu nende objektide alt, millele ükski muutuja enam ei viita. See toimib automaatselt, ilma programmeerija sekkumata. [AT]

MIME tüüp, *MIME type* – kirjas sisalduvate andmete tüübi kirjeldus. MIME tüüp koosneb kahest osast: kategooria ja formaat, näiteks image/gif. Vt. ka *MIME*. [TM]

MIME, *MIME (Multipurpose Internet Mail Extensions)* – Interneti elektronposti laiendus, mis annab võimaluse edastada mittetekstilisi andmeid, näiteks graafikat, heli ja faksi. Vt. ka *MIME tüüp*. [TM]

nähtavuse piiritleja, *visibility modifier* – määrab, millised klassid tohivad kasutada antud meetodit või muutujat - kas public, protected või private. Vt. ka *avalik, kaitstud, privaatne*. [MT]

objekt, *object* – üldine mõiste, mida kasutatakse mitmes tähenduses. Kõige sagedamini samastatakse objekti isendga. Vt. ka *isend*. [KH]

objekti hävitamine, *object destruction* – tegevus, mis toimub ilma kasutajata ja mille tulemuseks on ruumi ning ressursside taastamine.Vt. ka *taastumine, objekti loomine.*[DB]

objekti loomine, *object creation* – dünaamiline objekti loomine operaatoriga "new". Vt. ka *objekti hävitamine.*[DB]

ohutu, *safe* – programm on ohutu, kui ta ei tee kasutaja ega arvuti suhtes midagi kurja, ei kahjusta arvuti ega kasutaja heaolu ja tal puuduvad negatiivsed kõrvalefektid. Sünon. *turvaline* [TV]

omanduslik, *proprietary* – mingile ainult konkreetsele arvutisüsteemile või arvutile omane omadus, mis muudab ta teistega ühildamatuks ja vähendab sellega teisaldatavust. Vt. ka *Portability.* [TV]

osuti, *reference* – kui on tegu muutujaga, mis hoiab kompleksset tüüpi objekti, siis see muutuja mitte ei sisalda seda objekti, vaid on osuti sellele. Sünon. *viit* . [KH]

paigutus, *allocation* – kasutatakse seoses mälu hõivamisega. Vt. ka *taastumine.* [DB]

paigutusskeem, *layoutscheme* – Määrang, mille järgi kujundusjuht asetab objektid konteinerisse. Vt. ka *kujundusjuht* [MR]

paisktabel, *HashTableI* – kood paisktabeli väljas oleva elemendi identifitseerimiseks. [JS]

paisktabel, *HashTableI* – dünaamiline massiiv, milles iga element on ette nähtud ühe kirje salvestamiseks. [JS]

pakett, *package* – grupp Java klasse, mis on seotud mingi eesmärgi või rakenduse järgi. Sama paketi klassidel on spetsiaalsed juurdepääsu privileegid teineteise suhtes ja nad võivad olla projekteeritud koos töötama. [KH]

paneel, *panel* – peamine konteinerelement, mida kasutatakse komponentide grupeerimiseks raamides. Vt. ka *raam, konteiner.* [AP]

piiratud juurdepääs, *restrict access* – kasutatakse selleks, et piirata välis mõju meie klassi omadusele.Vt. ka *avalik,kaitstud ja privaatne.*

piiritleja, *modifier* – võtmesõna enne klasse, muutujaid ja meetodeid. Mõjub juurdepääsule, käitumisele või semantikale. Sünon. *modifikaator.* [KH]

pildivaatleja, *image observer* – kontrollib ja juhib pildi esitamist. Eeskätt mõeldud juhtudeks, kus pildi toimetamiseks kliendi masinasse kulub palju aega. Ta sisaldab pildi staatuse kohta käivat informatsiooni ning toimib sellele vastavalt kutsudes vajadusel välja värskendamisemeetodi. Vt. ka *värskendamine.* [UL]

pilt, *image* – *java.awt.image* klassina sisaldab →pilditöötlusevahendeid ning vastavaks tööks vajalike andmeid. Neid saab esitada nii staatiliselt (mingi varem salvestatud) või dünaamiliselt (kohapeal genereeritud). [UL]

pistikuliides, *socket* – madalatasemeline programmeerimisliides, mida kasutatakse arvutivõrgus kommunikeerumiseks. Ta saadab andmevooge rakenduste vahel, mis võivad olla erinevates arvutites. Samuti on pistikuliidesel olemas enamus failile iseloomulikke operatsioone (avamine, sulgemine, lugemine, kirjutamine jne.). [GK]

privaatne, *private* – selle modifikaatoriga varustatud meetodid ja muutujad ei ole väljaspool vastavat klassi nähtavad. Vt. ka *kapseldus, avalik.* [AT]

protokoll, *protocol* – sõnumivormingute ja kahele arvutile nende sõnumite vahetuseks kohustuslike reeglite formaalne kirjeldus. Protokollid võivad kirjeldada masinatevaheliste liideste alataseme üksikasju (nt bittide ja baitide liinile saatmise järjestust) või ülatasemel andmevahetust paigutusprogrammide vahel (nt seda, kuidas kaks programmi kannavad faili Interneti kaudu üle). Vt. ka *HTTP, FTP.* [TM]

protokollikäsiti, *protocol handler* – Java klassid, mis realiseerivad erinevaid protokolle URLi sisu edastamiseks. Nende ülesandeks on teha kindlaks URLi MIME tüüp ja filtreerida ühendusvoost kõik protokollisspetsiifiline, nii et sisukäsitini jõuaksid andmed puhtal kujul. Iga protokollikäsiti tuletatakse klassidest java.net.URLConnection ja java.net.URLStreamHandler. Sünon. *protokolliohjur.* Vt. ka *protokoll, MIME tüüp, sisukäsiti.* [TM]

pärimishierarhia, *inheritance hierarchy* – Java-klasside päritavusjärjekord. Vt. ka *pärimispuu.* [APo]

pärimispuu, *inheritance tree* – Java-klasside päritavuspuu. Vt. ka *pärimishierarhia.* [APo]

päring, *query* – dialoogi klassi kuuluv alamklass, kus kasutajalt oodatakse mingile küsimusele vastust. Vt. ka *dialoog* [MR]

päritavus, *inheritance* – võimalus defineerida klasside hierarhiat, kus alamklass pärib ülemklassilt (baasklassilt) tema omadused – muutujad ja meetodid. Vt. ka *mitmene päritavus* [TV]

püünis, *catch* – määrab erindikäsitsemise bloki, mis järgneb katse-direktiivile; püünis-direktiiv näeb välja nagu meetod: ta määrab argumendina selle erindittüübi, mille käsitsemist ta teeb. Kui püünis-direktiiv alustab tööd, siis erindi objekt antakse talle argumentina. Vt. ka *katse, erind, käsitsemine.* [JF]

raam, *frame* – kõige ülemise tasandi aken ekraanil, mis saadud objektist aken. Vt. ka *konteiner, raam, paneel.* [AP]

rakend, *applet* – Java rakendus www-lehekülje tarvis. Sünon. *applett, rakend, rake.* [KH]

rakendiviit, *applet tag* – sektsioon HTML dokumendis. Kirjeldab rakendi asukohta dokumendis ja rakendi ekraaniala suurust. Lisaks võimaldab rakendile parameetreid ette anda. [MT]

rakendus, *application* – rakendus, programm. Peab sisaldama meetodi $main()$, mille Java interpretaator programmi käivitamisel välja kutsub. [KH]

samasus, *peer* – andmete väljastamine ekraanisüsteemile ja sündmuste vastuvõtmine sisendseadmetelt sisaldab endas piiri ületamist Java ja kohaliku töökeskkonna vahel. Mingil tasemel peavad abstraktsete aknavahendite komponendid hakkama suhtlema objektidega, mis sisaldavad nn. sünnipäraseid meetodeid, et tagada koostöö kohaliku operatsioonisüsteemiga. Selle koostöö hoidmiseks nii puhtana ja nii hästi defineeritavana kui võimalik, kasutab Java peer-liidese kogumit. Peer-liides lubab Java keele komponentidel segamatult suhelda vastavate reaalsete komponentidega kasutaja töökeskkonnas. Vt. ka *loomulik, komponent, sündmus.* [AP]

server, *server* – $\rightarrow$klient [GK]

sidumine, *binding* – meetodi väljakutse seostamine meetodi definitsiooniga. Seostamine võib toimuda kas programmi kompileerimise või täitmise ajal. Javas on meetodite sidumine vaikimisi täitmisaegne. Seostamise saab seada kompileerimisaegseks piiritlejaga final. Vt. ka *lõplik.* Sünon. *seostamine.* [TV]

sisukäsiti, *content handler* – Java klass, mis realiseerib URLi käsitlemise vastavalt selle MIME tüübile. Tema ülesandeks on lugeda protokollikäsitilt saadud voost andmed ja konstrueerida nende põhjal mingi tuntud tüüpi objekt, näiteks String või Image. Iga sisukäsiti tuletatakse klassist java.net.ContentHandler. Sünon. *sisuohjur.* Vt. ka *MIME tüüp, pro-*

tokollikäsiti. [TM]

staatiline meetod, *static method* – meetod, mis sisaldub vahetult klassis ja eksisteerib sõltumatult tema eksemplaridest. staatiline meetod saab kasutada ainult staatilisi muutujaid. staatiline meetod võib olla ülekatetud ainult teise staatilise meetodiga. [APo]

staatiline, *static* – muutujad ja meetodid, mis on ühised kogu klassile, on staatilised. Vt. ka *staatiline meetod, staatiline muutuja.* [AT]

sõne, *string* – suvalisest hulgast sümbolitest moodustatud järjend. [JS]

sõnede ühisala, *shared string pool* – Java püüab kindlustada identseid stringe ja stringi literaale samas klassis, hoides neid sõnede ühisalas. [JS]

sõnemeetod, *string method* – Meetod(id), mida saab rakendada sõneobjekti(de)le. [JS]

sündmus, *event* – sündmused luuakse põhikeskkonna poolt ja edastatakse komponendile tema sündmuskäsitlusmeetodite kaudu. Vt. ka *tegevus, sündmuskäsitlus.* [AP]

sündmuskäsitlus, *event-handling* – sündmusele vastav käsitlusmeetod. Kui kasutaja teeb liigutuse (n. hiirevajutuse) komponendi ekraanipiirkonnas, siis abstraktsete aknavahendite lõim edastab selle sündmuse komponendile kutsudes esile sündmuskäsitlusmeetodi. Vt. ka *sündmus, komponent.* [AP]

taasjoonistamine, *repainting* – vastav meetod repaint(); komponent ei pöördu kunagi meetodi update() poole otseselt. Selle asemel, kui komponent vajab ümberesitust, planeerib ta pöördumist meetodi update() poole läbi meetodi repaint(). Vt. ka *ülejoonistamine, värskendamine* [AP]

taastumine, *reclaim* – mälu taastamine töö lõpus.Vt. ka *lõpetamine.* [DB]

tagasiväljakutse, *callback* – situatsioon, millal osuti objekti A meetodile antakse objektile B, ja objekt B kasutades selle osuti kutsub välja objekti A meetodi. [APo]

tegevus, *action* – meetodile action() edastatakse kasutajaliidese poolt tekitatud sündmused. Vt. ka *sündmus, sündmuskäsitlus.* [AP]

teisaldatavus, *portability* – programmi omadus, mis võimaldab teda täita erinevat tüüpi arvutitel ilma programmi muutmise või konverteerimiseta. Sünon. *teisalduvus, mobiilsus*. Vt. ka *omanduslik, loomulik kood*. [TV]

tekstiala, *textarea* – Mitmerealine tekstieditor koos vertikaalse ja horisontaalse kerimisribaga. [MR]

tekstikomponent, *text component* – abstraktsete aknavahendite komponendid tekstisisestuseks ning -esituseks. Vt. ka *tekstiala, tekstiväli* [AP]

tekstiväli, *text field* – Üherealine tekstieditor. [MR]

topeltpuhverdus, *double buffering* – on kasutusel joonistamise muutmisel kiiremaks ning sujuvamaks. Seda meetodit läheb vaja →eeljoonistamise puhul. Pilt joonistatakse eeljoonistuspuhvrisse ning kopeeritakse siis ühe joonistusdirektiiviga ekraanile.Vt. ka *joonistusmeetod*. [UL]

tsükliline osuti, *cyclic reference* – juht, mis tekib, kui viitade järjend algab ja lõpeb sama objektiga.Vt. ka *osuti ja osutitüüp. [DB]*

tuletamine, *derivation* – ???[KH]

turvehaldur, *security manager* – Java tarkvarasüsteemi osa, mida kasutatakse põhiliselt selleks, et tagada üle võrgu kohale tõmmatud koodi turvaline toimimine. Tüüpiliselt turvehaldur kontrollib "võõra" rakendi faili-, võrgu- ja mäluoperatsioone. [GK]

tuum, *CoreI* – Java keele tuum, mille peal (mille kaudu) kõik ülejäänu Javas töötab.Siin mõeldud Java API tuumik pakkette. [JS]

töökindel, *robust* – programm on töökindel, kui tal on mõistlik lahendus kõigi võimalike programmi töötamisel ettetulevate olukordade jaoks. töökindel programm ei jää kunagi "rippuma", pole võimalik, et kasutaja satub kogemata mõne tegevuse tulemusena ettenägematusse situatsiooni. Sünon. *lollikindel, robustne* [TV]

tühi muutuja, *uninitialized variable* – muutuja, mis ei ole veel initsialiseeritud. [DB]

tühi tüüp, *void type* – kasutatakse sellise meetodi defineerimiseks, mis ei tagasta mingit väärtust. [DB]

tüübi alamvahetamine, *downcasting* – osutile klassi A isendile omistatakse klassi B isendile osuti väärtuse (kus B on klassi A alamklass). Vt. ka *tüübi vahetamine* [APo]

tüübi vahetamine, *casting* – objekti osuti tüüpi vahetamine; on võimalik ainult samas pärimishierarhias asuvate objektidega. Vt. ka *tüübi alamvahetamine.* [APo]

tüübisüsteem, *type system* – tüübisüsteem on programmeerimiskeele mehhanism andmetüüpide kirjeldamiseks ja andmeobjektide tüüpide määramiseks, samuti avaldiste tüübilise korrektsuse kontrollimiseks. [TV]

uus, *new* – new operaatorit kasutatakse uue klassi eksemplari loomiseks. Java paigutab uue objekti mällu ning kutsub seejärel välja objekti klassi konstruktori. Vt. ka *konstruktor.* [AT]

URL, *URL (Uniform Resourse Locator)* – WWW-s kasutatav lauseformaat teistel arvutitel asuvatele failidele viitamiseks. URL on tavaliselt kujul protokoll://masin.domeen[:port]/kataloog/failinimi. [TM]

vahetu väljakutse, *explicit call* – meetodi väljakutse viis, millal meetodi kutsub välja programmeerija ise. Vt. ka *vaikimisi väljakutse.* [APo]

vaikimisi konstruktor, *default constructor* – ülemklassi konstruktor ilma parameetrideta. Java kutsub seda välja automaatselt, kui alamklassi konstruktori esimeseks lauseks ei ole this() või super(). [APo]

vaikimisi nähtavuspiirkond, *default level of visibility* – nähtavuspiirkond, mille omab iga muutuja või meetod vaikimisi. [APo]

vaikimisi väljakutse, *implicit call* – meetodi väljakutse viis, millal Java iseseisvalt sisestab programmi kehasse meetodi väljakutset. Vt. ka *vahetu väljakutse.* [APo]

valik, *choice* – Välimuselt nupu sarnane objekt, mis võimaldab teha valiku mitmest alternatiivsest võimalusest. Hetkel aktiivne valik on kirjutatud nupu peale, vajutus nupule avab kõik olemasolevad valikud. [MR]

varjamine, *shadowing* – juht, millal ülemklassi muutuja (või meetod) ei ole nähtav meie alamklassis sama nimega muutuja (või meetodi) esinemise tõttu. [DB]

vastuvõetav, *acceptable* – sobiv või mingis mõttes parim juht mingist valikust. [DB]

veaklass, *error class* – Java API defineerib parandamatu veade klassi veaklassi. Vt. ka *viga.* [JF]

veakäsiti, *error handler* – mehhanism vea töötlemiseks. Vt. ka *viga.* [JF]

veatöötlus, *exception handling* – mehhanism vea töötlemiseks, koosneb katse -püünis lausest.Vt. ka *erind, katse, püünis.* [JF]

vektor, *vector* – muutuva pikkusega või dünaamiline →massiiv. [JF]

viga, *error* – on esitatud Java objektina, ja inditseerib seose või virtuaalarvuti vead. Nendele ei pea püünist tegema. Vt. ka *veaklass.* [JF]

virtuaalarvuti, *virtual machine* – arvutitüübist sõltumatu Java töökeskkond. virtuaalarvuti oskab käitada Java programme ja on nagu omaette arvuti, selle erinevusega, et tegelikult võib virtuaalarvuti olla realiseeritud nii riistvaraliselt kui ka tarkvaraliselt. Java virtuaalarvuti on abstraktne spetsifikatsioon, mis on kõigile vabalt kättesaadav ja igaüks võib selle ise realiseerida talle sobivat riistvara- või tarkvaraplatvormi kasutades. Tänu virtuaalarvutile on saavutatud Java teisaldatavus. Vt. ka *teisaldatavus.* [TV]

virtuaalmeetod, *virtual method* – meetod, mille konkreetne tegevus sõltub meetodi väljakutsumise kontekstist. [TV]

voopaigutus, *flowlayout* – kujundusjuht, mis asetab objektid konteinerisse esinemise järjekorras vastavalt objektide ja konteineri suurusele vasakult paremale ja ülalt alla. Arvestab objekti eelistatud suurust. Algselt kasutab keskele joondamist. On algseks kujundusjuhtiks paneel objektidele. Vt. ka *kujundusjuht* [MR]

võrkpaigutus, *gridlayout* – kujundusjuht, mis asetab objektid konteineri fikseeritud arvu ridadesse ja veergudesse. objektide suurusi muudetakse meelevaldselt, et täita ridade ja veergudega saadud alad. Ridade ja veergude arv (võib ette anda ka ainult ühe dimensiooni) antakse kujundusjuhi konstruktoris. Vt. ka *kujundusjuht* [MR]

välja kutsumine, *invoke* – meetodi käivitamine programmi sees.Vt. ka *meetod.* [DB]

värskendamine, *updating* – kui komponent realiseerib oma vajadust ümberjoonistuda, pöördub ta abstraktsete aknavahendite poole planeerimaks ümberesitust läbi meetodi update(). Vt. ka *ülejoonistamine, taasjoonistamine* [AP]

värv, *color* – kirjeldab konkreetseid värve. Olemasolevatele lisaks saab uusi värve vastavaid RGB väärtusi teades ka ise genereerida. [UL]

värvimudel, *color model* – moodus värvi kirjeldamiseks. Jagatakse otseseks ja kaudseks. Otsese puhul määratakse läbipaistvus ja värvid otse (ARGB mudel). Kaudse puhul saadakse värv indekseeritud värvide väärtuste seast. Vt. ka *värv.* [UL]

õrn baasklass, *fragile base class* – probleem pärineb keelest C++. õrna baasklassiprobleem seisneb selles, et kui paljusid järeltulijaid omava klassi definitsiooni muutmisel, sinna uute muutujate või meetodite lisamisel tuleb ümber kompileerida ka kõik selle klassi järeltulijad. Javas on õrna baasklassiprobleem leidnud lahenduse. Vt. ka *päritavus.* [TV]

äärepaigutus, *borderlayout* – kujundusjuht, mis asetab objektid konteineri sees viide geograafilisse asukohta: North, South, East, West ja Center. konteinerisse saab paigutada ainult viis objekti, lisamisel tuleb anda peale objekti nime ka asukoht, kuhu paigutatakse. On algseks kujundusjuhiks akna ja raami jaoks. Vt. ka *kujundusjuht* [MR]

ühelõimeline, *singlethreaded* – mingi rakend on ühelõimeline siis, kui temas üle ühe lõime kunagi paralleelselt ei tööta. [GK]

ühendus, *connection* – virtuaalne andmevoog kahe rakenduse vahel, kus mõlemad pooled teavad teineteise olemasolust ja on valmis omavahel andmeid vahetama. [GK]

ühendustega protokoll, *connection oriented protocol* – antud protokoll on ekvivalentne telefonivestlusega; pärast ühenduse loomist saavad kaks rakendust andmeid edasi ja tagasi saata. Isegi kui mõlemad pooled ajutiselt infot ei vaheta, jääb ühendus alles. Tagatud on ka see, et andmed kaotsi ei lähe ja saabuvad alati õiges järjekorras. [GK]

ühendusteta protokoll, *connectionless protocol* – see protokoll on analoogne postiteenusega. Omavahel suhtlevad rakendused saavad lühikesi sõnumeid vahetada, aga keegi ei garanteeri, et need sõnumid kohale jõuavad ja õiges järjekorras on. [GK]

üledefineerimine, *overloading* – üledefineerimine on see, kui luuakse mitu sama nimega kuid erinevate argumentidega meetodit ja meetodi väljakutsumisel kasutatakse sobivat argumentide arvu ning tüüpe. [AT]

ülejoonistamine, *painting* – terminid painting, repainting ning updating hõlmavad komponentide ekraanile esitamise ja ümberesitamise meetodid paint(), update(), repaint(). Enamasti on neid vaja vaid objektide lõuend ja paneel puhul; meetodit paint() saab kutsuda igal ajal, aga praktikas pöördutakse tema poole, kui objekt tuuakse nähtavale esmakordselt, kui objekt muudab oma olekut või kui mingi õnnetus ekraanisüsteemis midagi untsu keerab. Vt. ka *taasjoonistamine, värskendamine, lõuend, paneel* [AP]

ülekatmine, *overriding* – alamklass saab ülemklassilt päritud meetodi üle katta, kui alamklassis defineeritakse samasuguse nime ja samasuguste argumentidega meetod kui ülemklassis. Sellisel juhul saab ülemklassi meetodit välja kutsuda nii: Superclass.meetod. Vt. ka *ülemklass, alamklass.* [AT]

ülelaaditud konstruktor, *overloaded constructor* – konstruktor, mille parameetrite arv ja/või nende tüüp erineb teistest konstruktoritest. [APo]

ülemklass, *parent class* – terminit kasutatakse pärimises ja tähendab klassi, mille omadused, muutujad ja meetodid (ainult public ja protected) me pärime meie klassis.Vt. ka *avardamine.* [DB]

ülemklass, *superclass* – kui mingi klassi sees on defineeritud alamklass, siis viimase suhtes on esialgu vaadeldav klass ülemklass. alamklass pärib ülemklassilt kõik oma avalikud muutujad ja meetodid. Vt. ka *päritavus, alamklass, abstraktne klass, klass.* [AT]

Inglise-eesti sõnastik

absolute position – absoluutne positsioon
abstract class – abstraktne klass
abstraction – abstraktsioon
abstract method – abstraktne meetod
acceptable – vastuvõetav
access modifier – juurdepääsu piiritleja
action – tegevus
allocation – paigutus
applet – rakend
applet tag – rakendiviit
application – rakendus
arc – sektor
area – ala
arithmetic exception – aritmeetikaerind
array – massiiv
array copy – massiivi kopeerimine
array index out of bound exception – massiivi rajaerind
array store exception – massiivi salvestuserind
attribute – atribuut
audio – audio
Abstract Windowing Toolkit – abstraktsed aknavahendid
base klass – baasklass
base type – baastüüp
basic access modifier – juurdepääsu baaspiiritleja
binding – sidumine
borderlayout – äärepaigutus
browser – lehitseja
button – nupp
byte-code – baitkood
byte code verifier – baitkoodi verifikaator
callback – tagasiväljakutse
canvas – lõuend
cardlayout – kaartpaigutus
casting – tüübi vahetamine
catch – püünis

checkbox – kontrollruut
checkbox group – kontrollruudugrupp
choice – valik
circle – ringjoon
class – klass
classfile – klassifail
class format error – klassi formaadiviga
class importing – klassi importimine
class instance – klassi isend
class loader – klassilaadur
class method – klassi meetod
class path – klassitee
class qualifier – klassi otsimistee
class structure – klassi struktuur
class variable – klassi muutuja
client – klient
clipping – klippimine
color – värv
color model – värvimudel
command line argument – käsurea argument
comment – kommentaar
compilation unit – kompileerimisühik
compiler – kompilaator
component – komponent
composition – kompositsioon
connection – ühendus
connectionless protocol – ühendusteta protokoll
connection oriented protocol – ühendustega protokoll
constant value – muutumatu väärtus
constructor – konstruktor
container – konteiner
content – sisu
content handler – sisukäsiti
context – kontekst
controller – kontroller
coordinates – koordinaadid
copy – koopia

core – tuum
creation – loomine
current class – jooksev klass
cyclic reference – tsükliline osuti
daemon – deemon
datagramm – datagramm
data item – andmeelement
debugging – silumine
default – vaikimisi
default constructor – vaikimisi konstruktor
default level of visibility – vaikimisi nähtavuspiirkond
derivation – tuletamine
derived type – tuletatud tüüp
destruction – hävitamine
development – arendus
development environment – arenduskeskkond
development kit – arendusvahendite komplekt
dialog – dialoog
double buffering – topeltpuhverdus
downcasting – tüübi alamvahetamine
download – sisselaadimine
drawing method – joonistusmeetod
dynamic binding – dünaamiline sidumine
embedded – sard
empty stack exception – tühimagasini erind
encapsulation – kapseldus
enumeration – loetelu
environment – keskkond
environment variable – keskkonnamuutuja
error – viga
error class – veaklass
error handler – veakäsiti
event – sündmus
event-handling – sündmuskäsitlus
exception – erind
exception handling – veatöötlus
exception class – erindiklass

explicit call – vahetu väljakutse
expression – avaldis
extention – avardamine
field – väli
file – fail
file not found exception – faili puudumise erind
file stream – failivoog
final – lõplik
finalization – lõpetamine
finally clause – lõpuks-lause
flag – lipp
flow layout – voogpaigutus
focus – fookus
font – font
font metrics – fondimeetrika
fragile base class – õrn baasklass
frame – raam
framework – raamistik
friendly class – sõbralik klass
FTP (File Transfer Protocol) – FTP
fully qualified class name – täielik klassinimi
functional – funktsionaalne
garbage collection – mälukoristus
general exception – üldering
graphics context – graafikakontekst
graphics object – graafikaobjekt
gridlayout – võrkpaigutus
Graphics User Interface – graafiline kasutajaliides
handler – käsiti
handling – käsitsemine
hashcode – paiskkood
hashtable – paisktabel
header – päis
heap – kuhi
hide – peitmine
host – host
hostname – arvutinimi

HTML (HyperText Mark-up Language) – HTML
HTTP (HyperText Transfer Protocol) – HTTP
illegal argument exception – vale argumendi erind
image – pilt
image data – pildiandmed
image observer – pildivaatleja
imageprocessing – pilditöötlus
implementation – realisatsioon
implicit call – vaikimisi väljakutse
index out of bound exception – indeksi rajaerind
inheritance – päritavus
inheritance hierarchy – pärimishierarhia
inheritance tree – pärimispuu
instance – isend
instance method – ekasemplari meetod
instance variable – isendmuutuja
interface – liides
interpreter – interpretaator
invoke – välja kutsumine
irregularity – korrapäratus
J-code – J-kood
Java – Java
Java script – Javaskript
kit – komplekt
label – märgend
layout – kujundus
layout manager – kujundusjuht
layoutscheme – paigutusskeem
length – pikkus
library – teek
line – joon
list – nimekiri
literal – literaal
local variable – lokaalmuutuja
location – asukoht
lock – lukustus
low-level – esmatasand

manager – juht
menu – menüü
menubar – menüüriba
menu item – menüüvalik
method – meetod
method overloading – meetodi ülelaadimine
method resolution system – meetodi otsimise süsteem
method signature – meetodi signatuur
MIME (Multipurpose Internet Mail Extensions) – MIME
MIME type – MIME tüüp
model – mudel
modifier – piiritleja
monitor object – monitor-objekt
multidimensional array – mitmemõõtmeline massiiv
multiple inheritance – mitmene päritavus
multithreaded – mitmelõimeline
native – loomulik
native code – loomulik kood
negative array size exception – massiivi negatiivse suuruse erind
networked application – võrgurakendus
new – uus
no such element exception – elemendi puudumise erind
null – null
null pointer exception – nullviida erind
object – objekt
object creation – objekti loomine
object destruction – objekti hävitamine
off-screen drawing – eeljoonistamine
off-screen buffer – eeljoonistuspuhver
origin – nullpunkt
oval – ovaal
overloaded constructor – ülelaaditud konstruktor
overloading – üledefineerimine
overriding – ülekatmine
package – pakett
painting – ülejoonistamine
panel – paneel

parent class – ülemklass
path – otsimistee
peer – samasus
pipe – toru
polygon – hulknurk
port – port
portability – teisaldatavus
predefined – eeldefineeritud
primitive type – algtüüp
priority – prioriteet
private – privaatne
private type – privaattüüp
proprietary – omanduslik
protected – kaitstud
protocol – protokoll
protocol handler – protokollikäsiti
protocoll exception – protokollierind
public – avalik
public variable – avalik muutuja
query – päring
random – juhuslik
random accessfile – otsepöördusfail
reclaim – taastumine
rectangle – ristkülik
reference – osuti
reference type – osutitüüp
relationship – seos
repainting – taasjoonistamine
resource – ressurss
restrict access – piiratud juurdepääs
robust – töökindel
runnable interface – käitusliides
run-time – käitusaegne
Run-time environment – käituskeskkond
run-time exception – käitusaegne erind
run-time system – käitusaegne süsteem
safe – ohutu

scalability – laiendatavus
scope – skoop
script – skript
script language – skriptkeel
scrollbar – kerimisriba
secure – turvaline
security – turve
security exception – turveerind
security manager – turvehaldur
server – server
set – hulk
shadowing – varjamine
shape – kujund
shared string pool – sõnede ühisala
singlethreaded – ühelõimeline
socket – pistikuliides
statement – direktiiv
static – staatiline
static method – staatiline meetod
static variable – staatiline muutuja
stream – voog
stream wrapper – voomähkur
string – sõne
string method – sõnemeetod
subclass – alamklass
subinterface – alamliides
submenu – alammenüü
subtype – alamtüüp
superclass – ülemklass
supertype – alamtüüp
system class – süsteemklass
terminal I/O – terminalikäitus
terminating – hävitamine
textarea – tekstiala
text component – tekstikomponent
text encoding – teksti kodeering
text field – tekstiväli

thread – lõim
thread class – lõimeklass
thread control – lõime juhtimine
thread group – lõimerühm
thread priority – lõime prioriteet
thread syncronization – lõime sünkroniseerimine
throw – edasilüke
token – lekseem
tool – töövahend
toolkit – töövahendite komplekt
transgression – korrapäratus
try – katse
type system – tüübisüsteem
uninitialized variable – tühi muutuja
unknown error – tundmatu viga
unknown host exception – tundmatu hostierind
unknown server exception – tundmatu serverierind
updating – värskendamine
URL (Uniform Resourse Locator) – URL
URL connection – URL-ühendus
validate – kehtiv
variable – muutuja
vector – vektor
verifier – verifikaator
verify error – verifitseerimisviga
view – vaade
virtual – virtuaalne
virtual machine – virtuaalarvuti
virtual method – virtuaalmeetod
visibility – nähtavus
visibility modifier – nähtavuse piiritleja
void type – tühi tüüp
windows – aken
wrapper class – mähisklass