

## I V A D A S

*Java* – šiuo metu, matyt, pati populiariausia programavimo kalba. Ja galima kurti tiek savarankiškas autonomiškas programas, tiek tinklu persiunčiamas ir interneto naršyklių vykdomas programas – klientines programas (angl. *applets*). Pagrindinis *Java* pranašumas, palyginti su kitomis programavimo kalbomis, yra jos nepriklausomumas nuo kompiuterių tipo. Nepriklausomumas pasiekiamas apdorojant *Java* programą dviem etapais: pirmiausia ją kompiliuojant į nuo kompiuterio tipo nepriklausomą vadinamąjį baitinį kodą virtualiai *Java* mašinai, o paskui šį kodą interpretuojant. Tai lemia ir *Java* trūkumus – šios programos našumas pasiekia tik kelis procentus analogiškos C++ programos našumo.

*Java* laikoma revoliuciniu programavimo žingsniu, pirmiausia dėl *Java* programavimo galimybių internete, kalbant tiek apie klientinį, tiek apie serverinį programavimą. Pagrindinė klientinių-serverinių sistemų idėja – centralizuotas informacijos bankas yra serveryje, į kurį gali kreiptis grupė klientinių kompiuterių. Serveryje saugomos informacijos pokyčiai greitai prieinami klientiniams kompiuteriams. Programinę įrangą kompiuteryje-serveryje, leidžiančią prie jo prisijungti kompiuteriams-klientams, vadinsime serveriu, o analogišką programinę įrangą nutolusiuose kompiuteriuose – klientu. Serverio realizacija labai keblė, nes jis turi apdoroti daugelį vienu metu ateinančių klientų užklausų, taip pat leisti klientams įrašyti informaciją į serverio saugyklą, apsaugoti informaciją užrašymo metu. Be to, kompiuteriai sistemoje gali būti skirtingų architektūrų ir valdomi skirtingų operacinių sistemų (OS). Interneto dalį – tinklą *www* taip pat galima laikyti klientine-serverine sistema, tik dar sudėtingesne, nes čia visi kompiuteriai gali atlikti tiek kliento, tiek serverio funkcijas (pavyzdžiui, užsakant naršykle prekes internetinėje parduotuvėje).

*Klientinis programavimas.* Pirmosios sistemos serveris – naršyklė funkcionavo maždaug taip: serveris generuoja statinius *www* puslapius naršyklei, o ši gali juos parodyti. Puslapių kūrimo kalba HTML (angl. *Hyper-Text Markup Language*) palaiko paprasčiausius informacijos surinkimo mechanizmus (tekstinius laukus, sąrašus, vėliavėles ir pan.), kurie leidžia atlikti du veiksmus: ištrinti formos duomenis (*reset*), ir pasiųsti surinktus duomenis serveriui apdoroti (*submit*). Serveryje šią užklausą apdoroja visų *www* serverių palaikoma sąsaja CGI (*Common Gateway Interface*; faktiškai tai yra serverio programinės įrangos dalis). Užklausos tekstas

nurodo CGI, ką daryti su atsiųstais duomenimis. Juos gali apdoroti pati CGI programa, kuri dažniausiai rašoma *Perl* kalba, tiesiogiai skirta tekstui apdoroti arba peradresuoti kitai kompetentingai programai.

Daug *www* serverių ir šiuo metu funkcionuoja CGI pagrindu. Jų pagrindinis trūkumas – našumo problemos. Šios problemos sprendimo būdas – programavimas kliento mašinai, pavedant naršyklei visą įmanomą darbą kliento pusėje. Vienas iš klientinio programavimo būdų – plėtinių modulių naršyklėms (*plug-ins*) rašymas. Plėtinys suteikia naršyklei papildomas funkcijas nuo to momento, kai jis yra įdiegiamas. Faktiškai plėtiniai atvėrė kelią kurti naujas klientinio programavimo kalbas – vadinamąsias scenarijų kalbas. Tokia klientinė programa tiesiog įterpiama į HTML puslapį, o ją apdorojantis modulis automatiškai suaktyvinamas peržiūrint puslapį. Labiausiai paplitusi scenarijų kalba, matyt, yra *JavaScript*, palaikoma tiek *Internet Explorer*, tiek *Netscape Navigator* naršyklių. Toks sprendimas yra greitas ir dažniausiai naudojamas grafinėms vartotojo sąsajoms (GUI, *Graphical User Interface*) patobulinti. Laikoma (B. Eckelis), kad taip išsprendžiama 80 % klientinio programavimo uždavinių.

Likusiems uždaviniams dabar populiariausias sprendimo įrankis yra *Java*, turinti visas scenarijų galimybes ir esanti visavertė daugiasrautė programavimo kalba su išplėtotomis bazinio tinklinio programavimo sąsajomis, išskirstyto skaičiavimo bei ryšio su duomenų bazėmis sąsajomis. Klientinis programavimas *Java* – tai vadinamųjų klientinių programų (*applets*) rašymas. Klientinė programa – paprastai nedidelė programa, vykdoma naršyklės viduje. Programa įkeliamą kaip *www* puslapio dalis. Ji suaktyvinus, vykdoma programa. Šių programų patraukli savybė – nepriklausomumas nuo kompiuterio platformos. Programos apdorojamos dviem etapais: pirmiausia kompiliuojamos į baitkodą (būtent baitkodas ir nepriklauso nuo kompiuterio architektūros; būtent jis siunčiamas tinklu naršyklei), o jį interpretuoja virtuali *Java* mašina (JVM, *Java Virtual Machine*), kurią palaiko dauguma naršyklių. Jei klientinę programą sudaro keli moduliai (klasės), tai jo siuntimui tinklu pagreitinti visi moduliai suglaudunami į vieną vadinamąją JAR (*Java ARchive*) rinkmeną, ir ji siunčiama. Saugumo sumetimais klientinės programos turi griežtus apribojimus darbui su klientinio kompiuterio ištekliais: pavyzdžiui, tokia programa neturi prieties teisių prie klientinio kompiuterio kietojo disko nei skaityti, nei įrašyti.

Kitas klientinio programavimo įrankis yra *Microsoft ActiveX* technologija, leidžianti prijungti programinius modulius prie *www* puslapio ir juos vykdyti.

Technologiją palaiko *Internet Explorer* ir *Netscape Navigator* naršyklės. *ActiveX* – savarankiški programiniai komponentai, kuriami kompanijos *Microsoft (Ms)* pasiūlyta technologija *COM (Common Object Model)*, toliau vystančia objektinio programavimo idėjas. Komponentai gali būti vykdomi tik kitų aplikacijų-konteinerių viduje ir praturtina jų funkcines galimybes. Tokių konteinerių pavyzdžiai yra *Ms Visual Basic*, *Ms Access*, *Ms Internet Explorer*. Iš komponentų kaip iš statybinių blokų gali būti montuojamos ir kitos aplikacijos. Tarkime, katalogų medžiui atvaizduoti kurioje nors grafinėje aplikacijoje galima rasti ir panaudoti daugelio firmų platinamus *ActiveX* komponentus, taip sutaupant programuotojo laiką.

Į HTML puslapį tokie komponentai įvedami sakinyje *<OBJECT>*, turinčiame parametrus *CLSID* unikaliam komponento identifikavimo numeriui įdėti ir *<CODEBASE>* – komponento rinkmenos URL (*Uniform Resource Locator*, unikalus rinkmenos adresas tinkle) adresui.

Pagrindinė problema, naudojant *ActiveX* komponentus HTML dokumentuose, yra kliento kompiuterio saugumo užtikrinimas. Jei klientinės programos dėl savo prigimties negali padaryti žalos klientui, tai *ActiveX* komponentai turi prieigą prie kliento kompiuterio atminties. Saugumą stengiamasi užtikrinti naudojant komponentams skaitmeninius parašus. Įdiegiant naują komponentą į kompiuterį, naršyklė dialogo lange užklausia vartotojo, ar komponentą reikia įkelti, ar ne. Lange nurodoma, kas sukūrė komponentą, kas platina, kas nustatė komponento autentiškumą (viena tokių organizacijų, išduodančių autentiškumo sertifikatus, yra *VeriSign*, [www.verisign.com](http://www.verisign.com)). Matyt, kiekvienam yra tekę matyti tokius langus įdiegiant programas Windows genties OS.

*Serverinis programavimas.* Klientų užklaustos serveryje apdorojamos tam tikromis programomis, tradiciškai CGI ir *Perle*. Tačiau *www-serveriai* gali būti rašomi ir *Java* (tokios programos vadinamos serverinėmis programomis (angl. *servlets*), ir naudojant *Java* serverinius puslapius (*JSP, Java Server Pages*). Pagrindinis tokių sprendimų pranašumas yra serverio nepriklausomumas nuo kompiuterių tipo.

*Savarankiškų programų rašymas Java.* *Java* – patogi programuoti kalba. Teigiama, kad *Java* parašyti programą galima daug greičiau nei *C++* kalba. Esminis *Java* trūkumas šioje srityje – našumas. Galimi problemos sprendimo būdai yra tam tikros pagreitinančios technologijos (*hotspot*), galimybė naudoti programoje kitomis kalbomis parašytus modulius; galimybė atsisakyti nepriklausomumo nuo kompiuterių

architektūros ir kompiliuoti programą tam tikriems kompiuterių tipams. Sparčiai evoliucionuojant kalbai, kiekviena aukštesnė *Java* versija tampa vis greitesnė.

## **Trumpa *Java* istorija**

1991 m. *Sun Microsystems* pradėjo projektą *Green*, skirtą buitinėje elektronikoje naudojamiems mikroprocesoriams valdyti. Projekto rezultatas turėjo būti programavimo kalba, nepriklausanti nuo procesoriaus tipo ir tinkanti įvairių gamintojų technikai valdyti. Nepavykus projekto realizuoti C kalba, pradėta kurti nauja programavimo kalba *Oak* (1995 m. pervardyta į *Java*). Kalbos kūrimo grupėje dirbo J. Goslingas (vadovas), P. Naughtonas, Ch. Warthas, E. Frankas ir M.Sheridanas. Pirmoji kalbos versija pasirodė 1992 m. Po kelerių metų projektas *Green* užsibaigė nesėkme, o jame dirbę žmonės įkūrė *JavaSoft* kompaniją, kuri toliau tęsė kalbos tobulinimo darbus.

Kaip tik tuo metu paplito interneto *www* tinklas, įsikūrė *Netscape Communications* kompanija. Kadangi internetas susideda iš daugybės skirtingos architektūros kompiuterių su skirtingais procesoriais ir valdomų skirtingų operacinių sistemų, viena aktualiausių problemų tampa internetui skirtų programų mobilumas. Ta pati programa turi būti įvykdoma ir *Intel*, *Macintosh*, UNIX kompiuterių tipams. Šios priežastys lėmė, kad *Java*, iš pradžių sukurta panašioms problemoms spręsti buitinėje elektronikoje, pasuko į tokių pat problemų sprendimą internete.

Autoriai suteikė *Java* daugelį C ir C++ kalbų bruožų, kad daugelis šiomis kalbomis programuojančių žmonių nesunkiai galėtų įsisavinti naują kalbą. Tačiau kai kurie giluminiai kalbos mechanizmai visiškai skirtingi, todėl klaidinga yra nuomonė, kad *Java* – tai C++ internetinė versija. Be to, *Java* ir nėra skirta konkuruoti su C++ ir galiausiai ją pakeisti; tai visų pirma nuo kompiuterių architektūros nepriklausoma programų internetui kūrimo kalba.

Dabar kelios pastabos dėl *Java* ir *JavaScript* kalbų, kad panašiai skambantys pavadinimai nesudarytų išpūdžio, jog tai giminingos kalbos. *Netscape Communications* sukūrė nesudėtingą scenarijų kalbą *LiveScript*, leidžiančią įterpti nesudėtingas programas tiesiog į HTML dokumento tekstą ir išplėsti standartines HTML galimybes. Taigi tokia kalba yra glaudžiai susijusi su HTML kalba ir gali būti

vykdoma tik su naršykle. Panašios yra *Visual Basic Script*, *AppleScript* scenarijų kalbos. Paplitus *Java*, *Sun Microsystems* sutikus, *Netscape Communications* pervardijo savo kalbą į *JavaScript*. Taigi *Java* ir *JavaScript* genezė yra visiškai skirtinga.

## Pagrindiniai kalbos bruožai

1. *Java* – objektiškai orientuota kalba. Visi *Java* programos duomenys yra objektai, išskyrus tik primityvius aritmetinius, simbolinius ir loginius duomenis. Tačiau ir tokiems duomenims yra klasės – kevalai, kad prireikus būtų galima jais operuoti kaip objektais.
2. Griežtai tipizuota kalba. Kiekvienas objektas turi griežtai nustatytą tipą.
3. Vienkamienis paveldimumas. Daugybinio paveldimumo, kurį palaiko C++, *Java* kalboje nėra. Tam tikrą panašumą su analogiška C++ galimybe užtikrina tik sąsajų klasių, pačių abstrakčiausių klasių, mechanizmas. Visos *Java* klasės paveldi klasės *Object* savybes.
4. *Java* pati skiria dinaminę atmintį objektams ir pati ją išlaisvina, kai objektas nebeturi nė vienos į save nukreiptos rodyklės (kitai – nuorodos). Šiuos veiksmus atlieka mechanizmas „šiukšlių rinktuvas“ (*garbage collector*), tam tikrais momentais peržiūrėdamas programos užimamą atmintį. Pačios rodyklės programuotojui nėra prieinamos. Objektai dinamiškai saugomi atminties srityje, vadinamoje „krūva“ (*heap*). Toks atminties mechanizmas – viena iš *Javos* lėtumo priežasčių.
5. *Java* turi patogų objektiškai orientuoto stiliaus mechanizmą programos vykdymo klaidoms apdoroti. Visos tokios klaidos gali būti apdorojamos pačios programos.
6. *Java* – daugiasrautė (kitai – daugiagijė, *multi-threaded*) kalba, leidžianti rašyti programas, vienu metu vykdančias kelis srautus toje pat programos adresinėje erdvėje. Yra srautų sinchronizavimo ir tarpsrautinio bendravimo galimybės.
7. Nepriklausomumas nuo kompiuterių platformos. *Java* programos apdorojamos dviem etapais: kompiliuojamos į tarpinį kodą – baitkodą

(baitkodas nepriklauso nuo kompiuterio platformos), o šis interpretuojamas virtualios *Java* mašinos JVM.

8. Saugumas. *Java* klientinės programos neturi prieities prie klientinio kompiuterio atminties (sakoma, kad jie vykdomi naršyklės „smėlio dėžėje“ – yra tam net specialus angliškas terminas *sandboxing*). Yra ir „patikimų“ skaitmeniniu parašu aprūpintų klientinių programų, neturinčių tokių ribojimų. Panašiai nieko kenksmingo serverio mašinai negali padaryti ir „nepatikimos“ serverinės programos.

## ***Java* programavimo instrumentai**

Kompiliatorių, interpretatorių, *Java* bazinės klases, klientinių programų peržiūrėjimo be naršyklės pagalbos įrankį, programos dokumentacijos HTML formate įrankį ir kitas priemones galima nemokamai parsisiųsti iš [www.javasoft.com](http://www.javasoft.com) tinklapio. Šiuo metu *Windows* aplinkai platinamas paketas J2SE SDK 5.0 (*Java 2 Standard Edition, Software Development Kit*), maždaug 40 MB apimties. To paties komplekto su integruota programavimo terpe *NetBeans* apimtis – jau per 90 MB. Norint kurti ir derinti serverines programas, teks parsisiųsdinti gerokai didesnę paketą J2EE 1.4 SDK (*Java 2 Enterprise Edition*) – per 110 MB apimties. Šiame pakete yra *Sun* serveris – konteineris serverinėms programoms, įrankis joms derinti ir daugybė kitų priemonių.

Vieno ar kito paketo įdiegimo instrukcija pateikiama minėtame tinklapyje. Patikrinti įdiegimą galima leidžiant tame pačiame pakete, *DEMO* aplanke, esančias demonstracines programas. Jei kompiuteryje nepavyks paleisti kompiliatoriaus, interpretatoriaus ar klientinių programų peržiūros įrankio, tinkamai pagal instrukcijas papildykite savo kompiuterio OS aplinkos kintamąjį *PATH* bei *CLASSPATH*.

Pagrindiniai įrankiai, be kurių neįmanomas šis paskaitų kursas, yra šie:

- *javac* – kompiliatorius į baitkodą,
- *java* – interpretatorius, JVM - *Java* virtuali mašina;
- *appletviewer* – klientinių programų peržiūros be naršyklės pagalbos įrankis,
- *servletrunner* – serverinių programų paleidimo įrankis.

Šių įrankių naudojimas aprašytas tekste.

Kurse naudosime *Java* klases iš šių klasių paketų:

- *java.lang* – bazinių kalbos klasių paketas;
- *java.io* – *Javos* įvesties ir išvesties klasių paketas;
- *java.util* – pagalbinių *Java* klasių paketas;
- *java.awt*, *java.awt.event* – *Java* įvykių apdorojimo klasių paketai;
- *java.net* – klasių darbui su tinklu realizuoti paketas;
- *javax.swing* – paketas grafinėms vartotojo sąsajoms kurti;
- *javax.servlet*, *javax.servlet.http* – paketai serverinių programų kūrimo klasėms.