

17. Laisvasis įrankis laisvam programuotojui – PHP

PHP yra programavimo kalba. *Apache* modulis įvykdo *PHP* kalba parašytą skriptą ir vartotojui pateikia *HTML* dokumentą. Laisvai platinama daugybė programų *PHP* kalba, todėl informacinių sistemų kūrėjai paprastai nekuria programų nuo pirmos eilutės, bet ieško jau sukurtų panašios paskirties programų ir patobulina jas bei pritaiko jas savo reikmėms.



17.1 pav. *PHP* logotipas

Nedidelė programuotojų dalis teigia, kad „P“ raidė akronime *LAMP* turėtų reikšti *Perl* arba *Python*. Jų nuomone *Python* yra modernesnė programavimo kalba ir taip teigiančių programuotojų gretos nuolat didėja. *Perl* kalba šiame kontekste vis mažiau naudojama.

Anksčiau aptartą *HTML* dokumentą puikiai papildo skriptai *PHP* kalba (17.1 pav.). Šis įrankis puikiai tinka ir yra dažnai naudojamas įvairios apimties projektuose. *PHP* kūrėjų bendruomenė yra puiki terpė įvairių tipų atviriesiems projektams⁴⁶.

PHP programavimo kalba su atvirąja programine įranga yra susijusi dviem aspektais:

1. Visų pirma pats šios kalbos modulis yra atvirasis projektas. Pati kalba yra kuriama atviros bendruomenės. Laisvai prieinami išsamūs *PHP* kalbos aprašai.
2. Šios kalbos techninė ir organizacinė realizacija puikiai tinka atviriesiems projektams, nepatogi nuosavybiniams projektams.

Šiame skyriuje aptarsime antrąjį aspektą – kodėl dauguma *PHP* programų yra atvirojo kodo. Trumpai susipažinsime su pagrindinėmis kalbos konstrukcijomis. Išmoksime, kaip toliau mokytis kalbos: kur rasti kalbos aprašus ir profesionaliai parašytų atvirųjų *PHP* programų.

⁴⁶ <http://php.net/>

17.1. Programos tekstų redagavimo poreikis

Maža universalių programų vartotojų dalis yra peržiūrinęjusi savo naudojamų programų pradinius tekstus. Dar mažesnė dalis yra juos redagavusi. Tokie vartotojai paprastai pasinaudoja nedidele sudėtingos programos funkcijų dalimi. Sudėtingesnėms internetinėms informacinėms sistemoms paprastai naudojama arba specializuota programinė įranga, arba universali programinė įranga, pritaikyta konkrečioms poreikiams. Abiem atvejais sistemą kurti turi specialistas, mokantis programuoti. Žiniatinklio eros pradžioje daug programų tekdavo programuoti nuo pirmos eilutės. Tokių programų kūrėjas turi kelis motyvus viešai paskelbti savo programų tekstus.

- Naudotojai ne tik pastebi programos klaidas, bet ir pasiūlo, kaip jas būtų galima ištaisyti. Aptarėme tai, kad dauguma instaliuojančių *PHP* programas moka programuoti. Tai natūralus bendradarbiavimas, nereikalaujantis jokių specifinių licencinių sutarčių.
- Kiti vartotojai programuotojai greičiausiai patobulins ir papildys programą. Pirminės programos kūrėjas nebūtinai gaus patobulintą versiją. Tačiau patobulintos versijos autoriai turi analogiškus motyvus viešai atskleisti savo programas. Dauguma atvirųjų licencijų reikalauja, kad patobulinta programa būtinai turi išlikti atviroji. Tai šiame motyvui formaliai padeda atvirosios licencijos.
- Šalutinis programos paviešinimo efektas – savęs pateikimas pasaulio programuotojų bendruomenės kontekste. Daugelį *PHP* programų kuria pavieniai programuotojai. Jie turi unikalią galimybę parodyti savo darbą, rasti bendraminčių, kuriančių panašias programas visame pasaulyje.

Tokių programų užsakovai dažnai nėra IT įmonės ir jie neturi komercinio intereso padaryti programą nuosavybinę. Pradiniai kodo fragmentai dažnai platinami su licencijomis, neleidžiančiomis uždaryti išvestinio kūrinio.

Praėjusiame skyriuje aptarėme, kaip veikia žiniatinklio serveris, pateikiantis nekintančius iš anksto sukurtus dokumentus. Dažnai to neužtenka. Interaktyviose sistemose, pavyzdžiui, el. parduotuvėse, vartotojo matomas vaizdas priklauso nuo jo atliktų veiksmų, todėl turi būti kuriamas programos. Net ir statinėse sistemose, pavyzdžiui, el. laikraščiuose, yra patogiau sukurti kuo paprastesnę sąsają straipsniams į duomenų bazę pateikti. Tada programa kuria vaizdą ekrane prie straipsnio pridėdama kitus komponentus. Bet kokios programos išvedamus duomenis būtų galima perduoti *Apache* serveriui, kuris juos pateiktų vartotojui. Tuo tikslu buvo sukurta *CGI-BIN* sąsaja. Žiniatinklio užduotims spręsti paplito *Perl* kalba, turinti patogių eilučių apdorojimo įrankių, kurie dažnai praverčia formuojant *HTML*.

17.2. Kodėl nuo *Perl* pereita prie *PHP*

Vėliau buvo sukurtas *Apache PHP* modulis. Sukurtos *PHP* kalbos sintaksė mažai skyrėsi nuo *Perl*. *PHP* tapo integruotas *Apache* komponentas. Maždaug 2000 m. ne mažai el. sistemų buvo perrašyta iš *Perl* į *PHP*. Tai suteikė ne tik techninių privalumų, bet ir lėmė patogesnę programavimą, nes pasitaikius klaidai *PHP* modulis galėjo pateikti išsamią informaciją, o „pakibus“ *Perl* programai *Apache* serveris nežinodavo problemos priežasčių, todėl ir vartotojui negalėdavo daug pasakyti.

17.3. Kaip veikia *PHP* modulis

Apache serverio aplanke gali būti įrašomi ne tik *HTML* failai, kurie pateikiami vartotojui tokie, kokie yra, bet ir *PHP* failai, kurie perduodami *PHP* moduliui. Tokiu atveju *PHP* modulis analizuoja tekstą ir apdoroja tai, kas yra tarp ženklų „<?“ ir „?>“. Likusi dalis perduodama vartotojui nepakeista. Tekstą nuo „<?“ iki „?>“ *PHP* modulis interpretuoja kaip programą *PHP* kalba. Jei failu yra keli tokie intarpai, jie interpretuojami kaip viena programa, išsaugomos kintamųjų reikšmės. Pateikiamo dokumento *PHP* programos vietoje išvedamas programos pateikiamas tekstas.

Sudėtingesniuose projektuose failas dažnai prasideda ženklų „<?“ ir baigiasi „?>“, visą *HTML* tekstą kuria *PHP* programa. Į statinius puslapius kartais įterpiamos atskiros *PHP* komandos.

17.4. Programavimo kalbų *Perl* ir *PHP* sintaksė

Matomiausias sintaksės skirtumas tarp čia aptariamų ir senesnių kalbų yra kintamųjų pavadinimai. *C* ir kitų klasikinių kalbų kintamųjų pavadinimai būdavo sudaromi iš raidžių ir skaičių. *Perl* ir *PHP* kalbose kintamojo pavadinimo pradžioje rašomas ženklas „\$“. Dėl šio skirtumo programos tekstas *C* ir *PHP* kalbomis atrodo labai skirtingas, tačiau *Perl* ir *PHP* perėmė daugelį *C* kalbos konstrukcijų.

Priskyrimo sakiny:

```
$a = $b = 500;
```

Čia sujungti du priskyrimai. Šis sakiny iliustruoja iš *C* kalbos perimtą konstrukciją, kai priskyrimo sakiny turi reikšmę, analogišką priskirtajai, todėl ši reikšmė gali būti dar kartą priskiriama.

Aritmetiniai veiksmai:

```
$n = 2 * $a + $b / 300;
```

Rezultato išvedimas:

```
echo $n;
```

Perl ir *PHP* turi sudėtingus eilučių apdorojimo mechanizmus. Taip pat šiomis kalbomis patogiu atlikti paprasčiausius veiksmus – kurti naują eilutę įtraukiant kintamųjų reikšmes.

```
echo "Kintamųjų reikšmės yra: $a, $b ir $n";
```

17.5. Elektroninių projektų specifika

Programinė įranga įkeliamą į serverį. Serverį prižiūrinčios įmonės darbuotojai gali peržiūrėti serveryje esančių failų turinį. Mažesniuose projektuose programinė įranga ir serveris užsakomi iš skirtingų asmenų. Todėl programos kūrėjas sunkiai galėtų apsaugoti programos pradinį tekstą.



17.2 pav. Bendruomenės kuriamas *PHP* vadovas⁴⁷

Kai kurie programuotojai šią problemą sprendžia apsunkindami trečiųjų šalių galimybes pasinaudoti jų programa. Tačiau dažniau pasirenkamas kitas būdas. Dauguma užsakovų keliamų

⁴⁷ http://en.wikibooks.org/wiki/PHP_Programming

uždavinių nėra unikalūs. Ne vienas programuotojas jau yra kūręs ką nors panašaus į užsakomą programą, todėl prieš pradėdant programuoti šioje srityje būtina paieškoti, kokios programos jau yra sukurtos.

WWW sprendimams reikia specializuotų programų. Ypač daug programuotojų visame pasaulyje kuria *PHP* programas. Internete laisvai prieinami ne tik programų tekstai, bet ir knygos apie *PHP* (17.2 pav.).

17.6. Klausimai ir uždaviniai

1. Kas yra *PHP*?
2. Sukurkite dokumentą su *PHP* skriptu, kurio antraštėje ir dokumento tekste būtų nurodyta, kiek milisekundžių praėjo nuo vidurnakčio iki tol, kol dokumentas buvo užklaustas. Abu šie skaičiai turėtų sutapti.