STŘEDNÍ PRŮMYSLOVÁ ŠKOLA ELEKTROTECHNICKÁ
ROŽNOV POD RADHOŠTĚM

PERSPEKTIVY ELEKTRONIKY
3. Celostátní seminář, 18. března 2003

VÝVOJ PROGRAMOVACÍCH JAZYKŮ

Ing. Pavel Pokorný
UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta technologická
Institut informačních technologií
tel.: +420 57 - 6033216, fax: +420 57 - 603 3333, e-mail:
pokorny@ft.utb.cz

1. Úvod

Když se řekne počítač, někteří lidé si představí jen technické zařízení. Ve skutečnosti je nedílnou součástí počítače také programové vybavení. Toto vybavení (software) představuje veškeré programové aplikace které na počítači běží včetně operačního systému. Vše bylo vytvořeno pomocí některého způsobu programování.

Program si obecně můžeme představit jako posloupnost algoritmů, které definují jeho chování. Cílem tohoto stručného příspěvku bylo porovnání vlastností různých programovacích jazyků. Vzhledem k tomu, že jich ale existuje velké množství nejsou zde obsaženy zcela jistě všechny a s jejich členění nemusí každý souhlasit – některé programovací jazyky mohou být zařazeny ve více skupinách.

2. Trocha historie

Stejně jako mnohé jiné produkty lidské tvořivosti, tak i programování prodělalo od jeho počátku spoustu vln. Po prvních programátorských nadšencích, opojených samotnou možností vysvětlit stroji jak má pracovat a přemýšlet, přišli ke slovu lidé, kteří přemýšleli především nad tím, jak na tvorbě programů co nejvíce vydělat. Myšlenky těchto lidí vedly k výraznému zvýšení produktivity práce i přes odpor skutečných programátorů.

Jednotlivé etapy vývoje programovacích jazyků na sebe navazují a často spolu velice úzce souvisí. Proto aby bylo možné pochopit určitý princip, je potřeba znát i jeho předchůdce (například objektové programování navázalo na strukturované programování).

3. Assembler a modulární programování

Počítač je číslicový stroj pracující s dvojkovou soustavou – nulami a jedničkami. Programy které zpracovává jsou vlastně posloupnosti instrukcí (každý typ procesoru má svoji vlastní instrukční sadu). Programovací jazyk assembler je jazyk na té nejnižší úrovni, jinými slovy psaný program je tvořen výhradně instrukcemi procesoru. Výhoda tohoto způsobu programování je zřejmá. Jde o „nejčistší“ kód, který je nejrychlejší a co do velikosti výsledného programu nejmenší ze všech ostatních jazyků (tzv. „vyšší programovací jazyky“ svůj kód překládají do assemblerů pomocí kompilátorů či interpretů). Nevýhodou je pracnost vytváření těchto programů, což se projevilo zejména s rozšířením možností počítačů, kdy rostla potřeba tvořit větší programy s více funkcemi.

Objevilo se tak modulární programování, které učilo, že se mají velké programy „rozbít“ na menší, částečně samostatné části, které měly podrobně specifikovány svoje rozhraní, pomocí kterého mohly spolu jednotlivé moduly komunikovat.

4. Strukturované a objektové programování

Strukturované programování definovalo několik základních pravidel pro psaní programu. Tato specifikace s sebou přinesla maximální přehled a srozumitelnost zdrojového kódu. Hlavním nositelem činnosti jsou algoritmy (posloupnost příkazů, cyklů a podmínek). Příkladem těchto jazyků jsou například C jazyk nebo Pascal.

Tento způsob programování měl i jednu nevýhodu, která se s rostoucí složitostí programů stále více zvýrazňovala. Tou byla obtížná práce se strukturami (například vytváření a práce s databázemi), z tohoto důvodu vznikl směr objektového programování.

Jeho podstatou je filozofie opačná než u programování strukturovaného. Nositelem aktivity se staly data (vytvořily se objekty, kterým se definovaly vlastnosti a činnosti). Například pokud bychom chtěli otevřít soubor, ve strukturovaném programování se s příslušnými parametry volá funkce která soubor otevře. V programování objektovém se vytvoří objekt soubor, který má metody které definují co vše s ním lze provádět (a jednou z nich je jeho otevření).

Objektové programování s sebou také přineslo nové vlastnosti. Jednalo se zejména o zapouzdření, dědičnost a polymorfismus. To, že se vytvářely objekty (třídy) vedlo ke zpřehlednění zdrojového kódu a tím i snížení chyb v programech, dále pak ke zvýšení stability a flexibility. Začaly tak vznikat objektové verze strukturovaných programovacích jazyků (např. C++).

5. Nové jazyky

Důležitým „členem“ objektových programovacích jazyků se stala Java. Pro svoji jednoduchost se ukázala jako téměř optimální jazyk vhodný pro výuku programování. Mezi její další výhody patří ta skutečnost že je zdarma a běží téměř na všech platformách (dnes existuje několik verzí, mimo jiné i pro tvorbu aplikací pro mobilní telefon). Dále vykazovala vyšší produktivitu programátorské práce oproti konkurenčnímu C++ (popudem vznikla další verze C jazyka označována jako C#).

Další způsob programování, který vznikl bylo využití tzv. komponent, což jsou vlastně moduly s rozšířenými vlastnostmi, z nichž nejdůležitější je schopnost relativně samostatné existence. Zpočátku byly komponenty používány k návrhu grafického uživatelského rozhraní (komponenta je např. tlačítko nebo zaškrtávací políčko v okně), v současné době se komponenty používají i při tvorbě dílčích modulů rozsáhlých distribuovaných systémů. Typickým představitelem komponentového programování je Visual Basic.

V posledních letech se objevil i jazyk XML, který se prosadil zejména jako univerzální „dorozumívací jazyk“ mezi komunikujícími počítači. Na nich je tak založené velké množství komunikačních protokolů a v současné době komunikují pomocí XML nejen systémy vzájemně, ale i sami se sebou. Výhodou tohoto jazyka je stálý přístup ke zdrojovému souboru, což usnadňuje eliminaci chyb.

Dalším z řady programovacích jazyků který stojí za zmínku je grafický UML, který se stal určitým standardem, ve kterém jsou zobrazovány návrhy programů v učebnicích i v praxi. Tento jazyk sjednotil roztříštěné způsoby zápisu vývoje programů, které doplnil velice komplexní metodikou návrhu.

6. Programování pro internet

Programování pro internet je samostatnou kapitolou ve vývoji programovacích jazyků. Vzniklo postupně s rozvojem internetu a lze ho rozdělit do dvou velkých skupin, které se vzájemně odlišují na základě své činnosti. Na straně klienta se lze setkat s jazyky HTML či javascriptem. Oba jsou „pasivní“ - jazyk HTML je velice jednoduchý, obsahuje pouze několik příkazů, ovšem jeho možnosti jsou omezené na zobrazování textu v určitém stylu, zobrazení obrázku tabulky a několik dalších. Javascript přináší oživení v podobě aktivních částí jako je např. zobrazení času nebo data, tedy položek, které se sami o sobě aktualizují.

Na straně serveru jsou velice rozšířeny jazyky PHP a ASP, tyto jazyky komunikují se vzdáleným počítačem (klientem), pro které generují pasivní stránky a naopak např. prostřednictvím formulářů zpracovávají klientovy požadavky. V těchto jazycích mohou běžet i složité aplikace pro práci s databázemi (např. e-mailové servery).

7. Výhledy do budoucna

S výhodnými perspektivami do budoucna se stále více prosazují nejrůznější systémy, které by tvorbu programů automatizovaly. Tuto činnost podporují objektově orientované jazyky, které tak umožňují, aby program sám vytvářel jiný program. V dnešní době se vývojových prostředí stále častěji setkáváme s bohatými sadami „průvodců“, kteří po zadání

několika parametrů vytvoří relativně rozsáhlé části programu, do které potom programátor doplní pouze specifické části řešící speciální úkoly.

Také roste obliba systémů, kterým pouze stačí popis vlastností výsledné aplikace a ony téměř sami danou aplikaci vygenerují a na vývojářích nechají pouze doprogramování některých specialit.

8. Závěr

Každý z výše uvedených způsobů programování má svoje přednosti a nedostatky. Volba programovacího jazyka je součástí tvorby aplikace, a je ovlivněna okolnostmi z nichž mezi nejdůležitější měřítka patří druh programu, k jakému účelu slouží, na jaké platformě (z pohledu hardware i software, zejména operačního systému) běží a v neposlední řadě i na znalostech a schopnostech programátora.

9. Literatura

ISSAACS, S. Dynamické HTML. 2. vyd. Computer Press Praha 2000. ISBN 80-7226-268-8.
MARCHAL, B. XML v příkladech. 1. vyd. Computer Press Praha 2000.ISBN 80-7226-332-3.
NENADÁL, K. Turbo C, popis jazyka. 1.vyd. GRADA,a.s. Praha 1991.ISBN 80-85424-08-8.
POKORNÝ, P. MRÁZEK, P. Objektové programování v C++. Skripta FT VUT Brno, Zlín 2000. (ISBN 80-214-1515-7)
VELECKÝ, P. Quo vadis, programování. Computerworld 10/2003, roč. XIV, s.17-19.
www.builder.cz
www.pcsvet.cz
www.w3.org