

Zobecnění zkušeností z tvorby knihovny pro databázový systém FoxBASE+ 2.10 na PC

Petr Bílý

1. Pojem knihovna v počítačovém prostředí

Použijeme-li pojem "knihovna" v běžném hovoru, míníme tím zpravidla soubor knih, eventuálně instituci. Speciálního obsahu toto slovo nabývá v počítačovém prostředí. Zde se používá dosti často a většinou předpokládáme intuitivní porozumění tomuto pojmu. Dokážeme jej přesněji vymezit? Pokusíme-li se vyjít z vlastností "klasické" knihovny zjistíme, že nám úvahy o ní poskytnou dobré vodítko pro specifikaci atributů "počítačové" knihovny. Jako vlastnosti klasické knihovny bychom mohli uvést m.j. umístění, dostupnost, počet čtenářů, opakovatelnost použití, účelovost, organizovanost, robustnost, řád, existenci seznamu, funkci správy a další.

Prakticky všechny tyto atributy mají svůj obraz v knihovně počítačové. Zde uvažujeme umístění fyzické (typ počítače, paměťové medium,...) a logické (způsob přístupu ke knihovně). Z hlediska uživatele není místo fyzického uložení příliš podstatné, důležité je, aby se logické umístění knihovny v čase neměnilo. Pochopitelně musí být definovány přístupové cesty ke knihovně a musí existovat strategie přístupu s vymezením práv pro jednotlivé uživatele, eventuálně časový režim využívání knihovny. Samozřejmostí je větší počet prvků a vícenásobnost použití, a to jak ve smyslu vhodnosti pro větší počet uživatelů, tak i opakovatelnosti využití jednoho prvku v různých situacích. U počítačových knihoven lze vždy určit jejich orientaci (na vědní obor, počítačový systém, programovací jazyk apod.). Knihovna musí být funkční v různých podmínkách. Je třeba minimalizovat závislosti na softwarovém i hardwarovém vybavení (např. verzi operačního systému, jazyka, typu výstupního zařízení aj.). Prvky (objekty) v knihovně musí mít jednotný formát, například ve způsobu vyzvednutí prvku z knihovny.

Nenahraditelná je role správce knihovny, osoby zodpovědné za provoz a údržbu knihovny na každé instalaci a dodržování provozního pokynu. Dokumentace - popis knihovny je spolu s návodem na využívání knihovny ne-postradatelným vodítkem pro uživatele.

Knihovnu posuzujeme i podle nároků na paměť. Časová hlediska uvažujeme nejen z pohledu zpracování a ladění (vývoje), ale také údržby knihovny samotné.

Počítačová knihovna se samozřejmě vyvíjí v čase. Provádějí se opravy, doplňky a "vylepšení" formou povýšení verzí knihovny, sleduje se kompatibilita s předchozími verzemi. Hodnotí se otevřenost knihovny ve smyslu možnosti nezávislých úprav jejím uživatelem. Nelze zanedbat ani otázku budoucí likvidace knihovny.

2. Co považovat za počítačovou knihovnu

Ve své úvaze se nemíníme omezit pouze na knihovny profesionálních výrobců software. Problematikou knihoven se musí totiž zabývat, byť ve skromnějším rozsahu, každý programátor usilující o vysokou efektivnost své práce. Za knihovnu v neomezenějším slova smyslu tedy můžeme považovat i "soukromou sbírku" obecněji použitelných částí programového kódu.

Klasifikovat knihovny podle obsahu je pro velkou rozmanitost obtížné, uvedeme jen několik příkladů :

- programové moduly k podpoře programovacího jazyka;
- programové moduly oborově orientované;
- matematické/statistické funkce;
- makropříkazy;
- dodefinované příkazy jazyka operačního systému;
- výkresy;
- dokumentace;
- rešerše z literatury;
- konstrukční tabulky a normy.

Je vidět, že v knihovně může být uloženo prakticky cokoliv obecně zejména kód programu, texty, čísla, obrazová informace, zvukový záznam.

Na základě dosavadních úvah se můžeme pokusit o formulaci definice:

Knihovna je organizovaný soubor datových objektů příbuzného typu, určených k opakovanému využití ve standardních situacích s nutností znalosti funkce objektu, ale nikoli jeho vnitřní stavby.

3. Knihovny na podporu vytváření programového vybavení

V dalším zaměříme pozornost již jen na jeden z možných typů knihoven. Jaké požadavky klademe na knihovnu funkcí a procedur pro určitý programovací jazyk, obecněji programovací systém? :

- náhrada často opakovaných úseků kódu jedním příkazem;
- náhrada obtížně programovatelných úseků;
- doplnění za účelem odstranění nedostatků výchozího systému;

- univerzální řešení typických situací, např. vyčíslení algoritmu, tisk formuláře,....;
- řešení obtížného nebo daným jazykem neřešitelného problému v nižším nebo vhodnějším jazyce.

Od používání dobré knihovny dále očekáváme:

- zvýšení čitelnosti a opravitelnosti programů;
- podporu stavebnicového budování úloh;
- sjednocení stylu práce programátorského týmu;
- vzorová řešení pro tvorbu vlastních programů;
- příspěvek ke sjednocení názvosloví;
- zkvalitnění dokumentace.

4. Postavení softwarové knihovny v kontextu CASE

Co je to vlastně Computer Aided Software Engineering (CASE)? Doslovný překlad nám dává širokou možnost zahrnout do něj téměř vše, co napomáhá vytvoření programových produktů. Teoreticky laděné úvahy o CASE v četných literárních pramenech nabízejí lákavé představy komplexního aparátu, jehož použitím se náklady na tvorbu a údržbu programového produktu, zpravidla aplikačního programu, dramaticky snižují. Předpokládá se, že CASE zasáhne do všech fází vývojového cyklu produktu, od analýzy až po sledování jeho chování v provozních podmínkách. Skeptické hlasy naopak soudí, že používání úplného CASE prostředí a zejména jeho úplné zvládnutí analytiky a programátory je časově a ekonomicky tak nákladné, že je vzdálenou budoucností. Jiný proud negativních ohlasů zlehčuje CASE jakožto nástroj pro ty, kteří nejsou schopni při programování přemýšlet. Sledujeme-li například reklamy na nadstavby různých databázových systémů, zjišťujeme, že pod označením CASE zde často figuruje jen generátory obrazovek, programů, výstupních sestav a dokumentátory.

Ve světle těchto skutečností lze považovat knihovnu programových modulů nikoli snad za reprezentanta CASE produktů, ale určitě za plnohodnotnou součást větších celků hodných tohoto označení.

5. Úvodní úvahy o knihovně pro FoxBASE 2.10

Správnost obecných úvah o knihovnách jsme ověřovali při praktickém vytváření programového produktu. Zaměřili jsme se na prostředí databázového systému Fox BASE+. Tento databázový systém pro osobní počítače považujeme za jeden z nejlepších ve své třídě, je snadno dostupný a dosti v ČSFR rozšířený. Vybudováním kni-

hovny jsme sledovali zefektivnění vlastní programátorské činnosti a současně jsme cítili potřebu jednotného komunikačního základu pro tuto zájmovou sféru.

Cílem bylo vytvoření souboru knihovných modulů, které by urychlily programování a současně i prosazení určitého stylu v analyticko-programátorské praxi v dané oblasti. Standardní postupy, jednotný styl a systém práce vnucený filosofií knihovny znamená úsporu času nejen ve fázi programování, analýzy, ale i v řízení spolupráce skupiny při řešení rozsáhlejšího projektu.

Na tvorbě knihovny pracoval 4 členný tým splňující předpoklady dobré úrovně znalosti výchozího systému i dostatku zkušeností z tvorby a provozu rozsáhlejších aplikací. Potvrdila se naše domněnka, že k úspěchu vede pouze týmová práce nevelké skupiny, v níž může docházet k vzájemným oponenturám koncepcí a řešení.

Stanovení postupu práce vycházelo z posouzení reálné potřeby zamýšlené knihovny a jejího určení. Před zahájením vlastních programátorských prací jsme určili rámcovou koncepcí obsahu knihovny, sjednotili jsme názvosloví, interface a způsob popisu.

6. Knihovna pro FoxBASE + 2.10 : LibFOX

LibFOX je název knihovny odsouhlasený firmou FOX Software Int. Tento knihovní systém není pouhá kolekce proveditelných knihovných modulů. Přesto však bývá tato část knihovny uživatelem využita a hodnocena v první řadě. I zde tedy uvedeme nejprve stručný přehled oblastí, které procedury a funkce LibFOX podporují.

Především je zde možnost práce s okny, kterou ve FoxBASE 2.10 citelně postrádáme. Knihovna LibFOX zavádí koncepcí okna, které vyhoví prakticky všem požadavkům programátora a přitom je "cena" jeho použití výrazně nižší než u dBASEIV, eventuálně FoxPro. Lze definovat okno dané velikosti a zobrazit jej na zadaném místě s rámečkem nebo bez něj. Typ rámečku je možno vybrat z několika standardizovaných typů nebo nadefinovat rámeček vlastní. Okno lze vyplnit zvoleným znakem, vyčistit jej, zobrazit v něm zvolený obsah a tento pak horizontálně posunovat či editovat. Významná je možnost vytvoření dynamického okna, jehož konečné rozměry se vypočtou v závislosti na obsahu, který v něm zobrazujeme.

Procedury a funkce pro práci s barvami slouží zejména k úspoře zápisu kódu při změně a návratu barev. Zásadní je však celková koncepce v práci s barvami, založená na předdefinování barev do proměnných charakterizujících nejběžnější situace při běhu aplikace a jejich integrace do mnoha dalších knihovných modulů.

Knihovna LibFOX věnuje mimořádnou pozornost systému hlášení. Hlášením je zobrazený text ve funkci vysvětlení, varování, označení a popisu chybového stavu,

nápovědy dalšího postupu atd. Úplnost, srozumitelnost a jednoznačnost hlášení, které programy poskytují obstará úlohy, mívají často rozhodující vliv na úspěch celé aplikace. K dispozici jsou knihovní moduly pro výpis hlášení do okna, případně do jedné řádky na obrazovku. Rozlišujeme několik typů hlášení - od normálního textu bez dalších efektů až po zvýraznění barvou, blikáním a zvukovým znamením. Text hlášení může být uveden přímo ve vyvolání knihovní procedury nebo pod svým identifikačním kódem. Ve druhém případě se úplné texty hlášení centrálně soustřeďují do jednoho textového souboru, což umožňuje jejich pohodlné vytváření i údržbu. Současně tím přispějeme ke zpřehlednění programu a odstraníme duplicitu textů hlášení.

Snaha o efektivnost zpracování vedla u mnoha systémových modulů k zavedení sdíleného datového prostoru. Je jím systémové pole - knihovním systémem předdefinované globální pole textových prvků. Využívají je zejména procedury pro zobrazení hlášení. Velká skupina procedur a funkcí ukládá své výstupy právě do systémového pole.

Tvorba hlavního menu a s tím spojeného řídicího programu aplikace je kontextově závislá tvůrčí činnost, kterou těžko zjednodušíme univerzálním knihovním modulem. Přitom se ale základní schema řídicích programů úloh pravidelně v různých modifikacích opakuje. LibFOX generátor menu na základě odpovědí uživatele na dotazy programu vytvoří řídicí program aplikace. V něm bude zajištěno provedení hlavní inicializační procedury, otevření příslušných databázových a indexních souborů a vytvořeno řídicí menu aplikace, ale i hlavní větvení podprogramů úlohy.

Help - uživatelsky definovaná nápověda, je průvodním znakem kvalitních, uživatelsky přátelských systémů. V rodině jazyků typu dBASE se s touto možností setkáváme až u FoxPro. Knihovna LibFOX sice nenabízí firemní profesionální HELP přímo integrovaný v systému, ale na druhé straně některými možnostmi HELP Fox Pro předčí. Nápovědu lze definovat nejen pro jakoukoliv položku čtenou příkazem GET/READ, ale i pro jednotlivé položky pop-up menu. Systém HELP lze snadno začlenit do již hotových programů. Texty jednotlivých nápověd se udržují odděleně ve speciálním textovém souboru. Výhodou tohoto uspořádání je jednoduchá tvorba, snadná údržba a přehlednost.

Ošetřování stisku speciálních kláves v různých kontextech vstupu dat je součástí téměř všech programů. Knihovní funkce pro indikaci stisknutí editačních kláves v kontextu READ program výrazně zpřehlední. Užitečná je funkce pro čtení (GET/READ) s rozpoznáním stisknutí jakékoli speciální či editační klávesy, včetně kláves klíčových (F1 až F10).

Problematika ověření podmínek, za kterých je vstupní hodnota platná, je řešena validačními funkcemi pro nejtýpější použití. Vedle funkcí pro test přítomnosti či absence načítané hodnoty v existujícím poli a ve větě databázového souboru je k dispozici schéma a příklad obecné validační funkce. Použitím další funkce lze zajistit, že se do proměnné načte jedna z předem zadané množiny hodnot.

Výstup sestav podporuje univerzální procedura. Má několik parametrů, jimiž lze zadat nejen typ výstupní sestavy (REPORT, LABEL, ...), co je předmětem výstupu, jaké výstupní zařízení použít, ale i podmínku pro výběr údajů do výstupní sestavy. V proceduře je zabudován mechanismus umožňující přerušit výstup klávesou ESC a jeho případné další pokračování.

Výběry z pole pracují se systémovým polem. Jedna z funkcí vrací jeden vybraný prvek zobrazeného pole. Pomocí jiné funkce můžeme označit (ev. zvolená označení rušit) několik prvků pole a tím vytvářet seznam vybraných údajů. Na tyto funkce lze pohlížet jako na dynamická menu s předem neurčeným počtem a obsahem položek. Funkce pro výběry vět z aktivního databázového souboru zobrazí seznam zvolené délky, kdy každá položka (řádka) tohoto seznamu je odvozena z údajů databázové věty. Jsou k dispozici funkce pro výběr jedné věty i funkce, jimiž lze označit (a tudíž vybrat) i několik vět najednou.

Knihovna LibFOX nabízí dokumentátor - databázi popisů, v níž se dokumentují programové moduly, ale i datové objekty jiných typů. Plnění databáze usnadňuje procedura pro zápis specifikací modulů. Využití databáze popisů je možné pomocí příkazů jazyka FoxBASE pro nejrůznější dotazy, zejména charakteru křížových referencí. Například zjištění, ve kterých modulech je použita která globální proměnná, nebo ze kterých modulů je zkoumaná procedura vyvolávána je cenné zejména při zásazích do programů. Nejběžnější činnost - generace popisů modulů v "čitelné" textové formě ale již opět automatizuje knihovní procedura. Pro srovnání: firemní FoxDoc pracuje až s "hotovým" programem, zatímco LibFOX dokumentátor umožňuje zapisovat specifikace modulu a využívat databázi popisů již ve fázi analýzy.

Zásadním přínosem LibFOX pro ošetření chyb je indikace přerušení, která neumí obhospodářit FoxBASE. To například znamená, že program nemusí zhavarovat, není-li po vyvolání přenosu založena disketa nebo nejsou uzavřena dvířka disketové jednotky, není-li v tiskárně papír apod. Navíc naleznete v knihovně vzor obecné chybové procedury pro ošetření chybových stavů všech programů aplikace.

Knihovna LibFOX respektuje práci s národními verzemi, obsahuje aparát pro řazení textových řetězců dle ČSN. Chceme-li naopak data uložená česky vytisknout na tiskárnu, která česky "neumí", pomůže procedura odstraňující diakritická znaménka.

Funkce pro práci se řetězci umožňují vybírání nebo rušení podřetězců zprava i zleva, náhradu všech výskytů znaku v řetězci podřetězcem, srovnání dvou řetězců s možností uvedení "divokých" znaků v jednom z nich, rozdělení řetězce na podřetězce dle určené skupiny znaků.

Generátor náhodných veličin obsahuje funkce pro generaci náhodného čísla, data, znakového řetězce.

Procedura pro interaktivní vytvoření logické podmínky poskytne možnost pro efektivní práci i programátorsky nezkušené obsluze.

Celá řada procedur a funkcí přispívá ke zjednodušení komunikace programu s uživatelem (např. používání kurzoru, zvukového znamení, stav klávesy insert, stisknutí přípustných kláves, zadání čekacího intervalu a další).

7. Knihovna LibFOX a systematické budování aplikace

Je účelné zapojit knihovní systém do všech fází procesu vytváření aplikační úlohy. Sama knihovna LibFOX je koncipována tak, aby nabídla a snad přímo vnutila systém a styl práce, zejména pro potřeby rozsáhlejších aplikací.

Jádrem knihovny je inicializační procedura, která vytváří prostředí aplikace. Některé prvky takto vytvořeného prostředí jsou nutné pro správnou funkci knihovních modulů. Prostředí aplikace jako celek vytváří základnu pro systematické budování zejména složitých úloh. Obsahuje předdefinované hodnoty stavových proměnných systému, inicializaci obecné chybové procedury, HELP procedury a nasazení celé řady globálních proměnných, které plní v aplikačních programech roli systémových proměnných. Příkladem jsou definice barev pro různé typy objektů, kódy pro řízení tiskáren, popis (souřadnice) standardních prostorů na obrazovce, definice doporučených typů rámečků, systémové pole a definice některých systémových proměnných, které mohou aktivně využívat jak systémové tak i aplikační programy.

Knihovní procedura pro vytvoření prostředí je použitelná univerzálně pro libovolnou aplikaci, je však možné, aby správce aplikace provedl potřebná doplnění nebo opravy podle konkrétní potřeby. Z toho vyplývá, že knihovna je otevřený systém, což podtrhuje i možnost zásahu do zdrojových tvarů knihovních procedur.

8. Vytvoření podmínek pro provozování a šíření knihovny

Donedávna jsme byli u nás zvyklí přijímat univerzálnější programové produkty převážně jen ze zahraničí od "velkých" softwarových firem. Pokud vznikaly u nás produkty aspirující svými vlastnostmi na obecnější použití v širším rozsahu, zpravidla jim k úspěšnému šíření něco chybělo. Často to byla neúplná či zcela chybě-

jící dokumentace. Svě se hrála i neexistující či chybná marketingová politika a spolu s tím většinou nedostatek propagace a možná i nízká stimulace k rozšiřování produktu.

Naše zkušenosti z dokončovacích prací na knihovně LibFOX a zahájení jejího prodeje potvrzují, že mezi "programátorským dokončením" softwarového produktu a okamžikem, kdy lze tento produkt uvést do praktického života, je ještě dlouhá cesta. V další části se proto budeme zabývat popisem prací, které spadají do tohoto období.

Jako výchozí stav vezmeme ten, v němž se mnoho softwarových produktů předává uživateli s tím, že dokumentace se dokončí a nedostatky "vyladí" až za provozu. Víme, že to většinou není důsledkem pokusu něco odbýt, často zde hraje zásadní roli potřeba dodržet termíny.

Při důsledném postupu je základním krokem dokončení referenční části dokumentace spolu s případnými přílohami popisujícími syntaxi prvků produktu. Tímto definitivně uzavíráme specifikaci produktu. V ideálním případě je alespoň tato část dokumentace uložena v počítači (viz LibFOX dokumentátor) a jsou k dispozici prostředky, jak z této formy vygenerovat textový zápis.

V další etapě probíhají paralelně práce na dokončování dalších částí dokumentace (uživatelská příručka) a testování funkcí produktu. Při testování je nezbytné ověření činnosti produktu podle jeho popisu (viz referenční část vytvořená v předchozí etapě). Pouhé testování souladu s představami či poznámkami programátora nedostačuje. V jeho průběhu vznikají drobnější příklady, které použijeme do uživatelské části příručky. Po dokončení (sepsání) uživatelské části příručky znovu testujeme správnost v ní uvedených příkladech.

Pokud se rozhodneme doplnit produkt o demonstrační verzi, pracujeme na ní souběžně s předchozími dvěma etapami. Ověřujeme si tím jednak působení našeho záměru v praktických podmínkách, jednak takto podrobuje produkt dalšímu podrobnému testování. Demonstrační verze knihovny LibFOX slouží nejen k poskytování základních informací o funkčních vlastnostech produktu a k předváděcím účelům. Pro uživatele může být zdrojový tvar programů demo verze zásobárnou dalších příkladů praktického použití knihovnických modulů. Na rozdíl od mnoha jiných čistě reklamních DEMO pracuje demo verze LibFOX s "ostrou" knihovnou, čili zobrazené výstupy nejsou simulované.

Sebedůkladnější testování ovšem nikdy nemůže odhalit všechny chyby. Dvojnásob to platí v případě testů prováděných samotnými autory. Ty jsou sice nezbytné, avšak zařizování určitých postupů, stereotyp nahlížení na problém a určitá "únava z vlastního díla" jsou na závalu. Proto lze doporučit, zejména pro větší projekty, zadat tzv. beta-testování nezávislým odborníkům. Stanovení jejich počtu, jejich získání a zainteresování, určení pravidel a doby vymezené pro beta-testování

by mohly být předmětem samostatného rozboru. Po návratu výsledků beta-testování je třeba opravit chyby a rozhodnout, které připomínky k rozšíření nebo naopak omezení funkcí produktu budete respektovat. Beta-testování knihovny LibFOX bylo poměrně rozsáhlé a přineslo všestranný užitek autorům i testujícím osobám.

V tomto okamžiku by se mohlo zdát, že dílo je již zdárně dokončeno. Musíme se však věnovat ještě problémům marketingové a distribuční politiky. Je nutno vyřešit otázku ceny, případně slev. Určí se způsob propagace a náklady věnované na reklamu. Rozhodneme se, zda produkt bude distribuován v jedné, či několika modifikacích. V případě knihovny LibFOX je to například verze uživatelská (přeložené moduly), vývojářská (obsahuje i moduly ve zdrojovém tvaru) a demonstrační. Ve většině případů je třeba se zabývat i ochranou před nežádoucím způsobem využití či dokonce zneužití té které verze. V oblasti softwarového trhu PC počítačů je téměř samozřejmostí, že produkt je schopen respektovat národní prostředí. Zde se opět musíme rozhodnout pro normu Latin2 nebo "Kamenických", případně obě.

Konečně nelze přejít ani problém přípravy produktu k distribuci (v oblasti PC je to konkrétně volba velikosti a kapacity použitých disket), způsob komprese dat na disketě a rozdělení modulů na jednotlivé diskety. Neopomeneme vytvořit instalační proceduru včetně rozvinutí komprimovaného tvaru.

Závěr

Možnost použití knihovny je základem pro vytváření spolehlivých, efektivních, dobře čitelných a dobře se udržujících aplikací. Knihovna LibFOX doplňuje databázový systém FoxBASE 2.10 o více než 100 nových funkcí, které výrazně rozšiřují její možnosti a se zachováním všech výhod, které FoxBASE má, tj. rychlost a malá spotřeba paměti, ji v mnoha ohledech staví na úroveň FoxPro a v některých funkcích dokonce ještě výš. Způsob řešení a obsah inicializačního souboru knihovny LibFOX může sloužit jako základní model, podle kterého lze vytvářet další nadstavby a doplňky této knihovny.

Knihovna LibFOX byla vyvinuta v letech 1989-1990 kolektivem programátorů vedeným ing. Ladislavem Gočem. Další informace o knihovně a možnostech jejího zakoupení lze získat na adrese SOFTIR, PO BOX 39, 751 51 Přerov nebo na pražském telefonním čísle (02) 42 64 43.

Autor: Ing. Petr Bílý
SOFTIR, P. O. BOX 39, 751 51 Přerov I
tel. 02-426 443, 0641-4267