

OZOGAN KLONDAIK

Vývojové prostředí pro snadné programování ve Windows

Kurz obsluhy a programování
Popis syntaxe jazyka KLONDAIK
Knihovna vizuálních komponent

Runtime modul pro distribuci aplikací

Novinky a změny v programu
Licení podmínky

Program dodává a další informace poskytně:
OZOGAN, 1. Máje 97, 460 01 Liberec
telefon: (048) 52 28 338, e-mail: info@ozogan.cz.

Nové verze programu je možné získat na internetu:
<http://www.ozogan.cz>

Cena programu, způsob dodání:
Cena programu při instalaci na jeden počítač je 1 990,- Kč.
Program je možné objednat telefonicky, poštou, mailem.
Program se dodává poštou na dobírku, daňový doklad
je vždy součástí zásilky.

KURZ OBSLUHY A PROGRAMOVÁNÍ

1. část - Základy programování

Základy programování, použití knihoven procedur a funkcí

2. část - Základy tvorby aplikací na formuláři

Popis základů programování aplikací skládáním komponent na formuláři včetně použití základních komponent

3. část - Knihovna vizuálních komponent

Obsahuje popis použití vizuální knihovny komponent

4. část - Databáze na formuláři

Použití databází na formuláři.

LICENÈNÍPODMÍNKY

PRÁVAREGISTROVANÉHOUŽIVATELE:

- možnostvyužívání programového vybavení po neomezenou dobu
- aktualizace programu na novějšíverzi za zvýhodnìnoucenu
- záruka a servis pro registrovaného uživatele

JE ZAKÁZÁNO:

- jakékoliv i bezúplatné šíøeníprogramu nebo jeho èásti, nedodrženíbude posuzováno jako porušování autorskýchpráv a uživatelbude povinen nahradit výrobcivzniklou škodu
- pøevádìt získané oprávnìní na jinou osobu, pùjèovat, pronajímat, dekódovatèi jiným nepovolenýmzpùsobem nakládat s programovýmproduktem

REGISTRACE PROGRAMU:

- Plná verze programu je vázána na registraci pro konkrétní poèítaè (sériové èíslo harddisku). Poèet poèítaèù s registrovanýmprogramem je vázán na poètu dodanýchlicencií.
- Dalšípoèítaè je možné zaregistrovat bezplatnì vždypouze po roce používání. Døívjší požadavekna zmìnu poèítaèe podléhá manipulaènímu poplatku 250,- Kč, maximálnì však ètyøikrát za rok. Èastìjšímìna za dvojnásobný poplatek.
- Požadavekna registraci a zmìnu poèítaèe musí býtdodán autoru programu vždypísemnì s uvedením kódupoèítaèe, kterýse zobrazuje pøikaždém spuštìníneregistrované verze programu.
- Pøizakoupení licence na jeden poèítaè je možné zaregistrovat program zdarma pro dva poèítaèe. Napøíkladjeden v zamìstnání, druhýdoma. Program nesmí v tomto pøípadibìžetna obou poèítaèích souèasnì.

ZÁRUKA,REKLAMAÈNÍPODMÍNKY:

- výrobceposkytuje šestimìsíèní záruku na provoz programového produktu dle dokumentace. Odpovìdnosti za vady se výrobcemùžezbavit bezplatnýmzasláním opravené verze programu uživateli
- výrobce neruèí za ztráty a škody zpùsobené chybnýmvyužíváním èi neoprávnìnými zásahy do programového vybavení
- reklamace musí býtpodána písemnì a musí obsahovat pøesnìypopis závady

DALŠÍUJEDNÁNÍ:

- výrobceprohlašuje, žeje držitelemautorskýchpráv k dodanému programovému vybavení a že ponese sám dùsledky všechsporù souvisejícími s tímto právy
- autor programu bude veškeré informace zasílat na adresu uvedenou v registraèní kartì. Její pøípadná zmìna musí býtoznámena písemnì.

Novinky a změny v programu

Verze 1.03 (24/06/1999)

- z důvodu malého využívání byly zrušeny grafy a tabulky Excelu. Výhodou je snadnější instalace programu a menší obsazený prostor na disku. Registrovaní uživatelé, kteří uvedené moduly využívali však mají možnost získat verzi 1.03 včetně grafů a tabulek.
- možnost animace programu (umožňuje kontrolovat průběh programu přes jednotlivé příkazy, volání procedur a funkcí)
- doplněna metoda `DataTable.SetExclusive` pro nastavení a kontrolu exklusivního přístupu
- doplněny metody pro kreslení na plochu objektu `Image`
- doplněn popis komponent `Application`, `DataImage`, `MediaPlayer` a `Screen`
- existuje několik variant programu lišící se pouze datem. Síťově byly pouze po internetu pro testování vybraným registrovaným uživatelům.

Verze 1.02 (10/04/1999)

- kompletní instalace na disketách, včetně kolekce obrázků pro tlačítka a ikony

Verze 1.02 (31/03/1999)

- distribuce programu pouze na internetu
- zpřístupnění `RunTime` modulu pro možnost distribuce aplikací
- v editoru programu možno nastavit velikost fontu
- možnost práce s obecnými binárními soubory
- možnost naetení souboru do proměnné (`FileToVar`), uložení proměnné do souboru (`VarToFile`)
- pro databáze doplněny procedury `DbfSetExact` a `DbfSetSoftSeek`, které definují reakci po neúspěšném hledání funkcí `DataTable.Seek` nebo `DbfSeek`
- doplněna komponenta `Timer` pro spouštění akcí v zadaném časovém intervalu
- doplněny nové komponenty pro práci s databázemi:
 - `DataComboBox` - Seznam s možností zadání nové hodnoty
 - `DataListBox` - Výběr seznamu předdefinovaných hodnot
 - `DataLookUpCombo` - Výběr seznamu z jiné databáze
 - `DataLookUpList` - Podřízený seznam z jiné databáze
 - `DataRadioGroup` - Databázový přepínací box

Verze 1.01 (10/03/1999)

Neveřejná verze zpřístupněná pouze pro vybrané registrované uživatele, obsahuje první variantu `RunTime` modulu pro možnost distribuce aplikací.

Verze 1.00 (12/02/1999)

- kompletní popis vizuální knihovny
- doplnění možnosti nastavení vlastnosti z programu u všech komponent
- zrušení komponent, které by nebylo možné ovládat z programu, případně byly funkčně duplicitní
- spousta dalších drobných změn

Verze 0.07 (28/12/1998)

- v deklaraci proměnných a procedur a funkcí již není nutné uvádět typ deklarované proměnné
- doplněny funkce pro konverzi kódování háčeků a čárek mezi kódováním Kamenický, Latin2 a Windows
- z programu je možné volat mimo tisku reportu i editaci předlohy sestavy

Verze 0.06 (12/12/1998)

- doplněna nápověda k programu a kurz použití programu
- při vyvolání nápovědy z editoru programu stiskem klávesy `Ctrl+F1` se zobrazí nápověda pro slovo, na kterém stojí kurzor

Verze 0.05 (07/12/1998)

- doplnily nové procedury a funkce pro deklaraci systémových proměnných
- možnost předávání parametrů do sestav

Verze 0.04 (25/11/1998)

- při výskytu chyby v programu je nastavena pozice kurzoru na místo nalezení chyby
- doplněn help obsahující kurz programu a popis syntaxe
- možnost vytvoření indexu z databázového okna

Verze 0.03 (17/11/1998)

- doplnily procedury a funkce pro kontrolu databází v síťovém zpracování:
- doplnily procedury a funkce pro práci s memopoložkami databází

Verze 0.02 (29/10/1998)

První veřejná demoverze. Èástečný manuál v textovém souboru.

Verze 0.01 (04/10/1998)

První, ne zcela veřejná verze systému. Uvolněna byla pouze pro registrované uživatele systému OZOGAN KLONDAIK 2. Bez manuálů a helpu.

RunTime modul pro distribuci aplikací

RunTime je EXE program, který je možné použít pro spuštění programů vytvořených z kompilovaných vývojových prostředků systému OZOGAN KLONDAIK. Programy spuštěné pod RunTime nejsou omezeny vlastnostmi demoverze. V programech není možné provádět žádné změny. RunTime modul je zahrnut v dodávce systému KLONDAIK a je možné jej šířit vytvořenými aplikacemi zcela zdarma bez dalších poplatků.

RunTime je možné umístit včetně všech požadovaných souborů v archívním souboru na jednu disketu. Není nutné provádět žádnou instalaci, stačí archívní soubor rozbalit v libovolném adresáři. Neprovádí se žádný zásah do systémových registrů Windows. RunTime je možné provozovat také přímo z CD-ROM. RunTime při své činnosti nezapíše žádné údaje na disk.

Pro správnou a bezchybnou činnost RunTime jsou vyžadovány následující soubory (uvedena velikost souboru a datum vytvoření souboru):

KL3RUN.EXE datum a velikost dle verze
SDE32.DLL 131 072 04/05/1998
SDECDX32.DLL 264 192 27/09/1997
VCL30.DPL 1 287 088 23/10/1997
VCLX30.DPL 219 584 05/08/1997
VCLDB30.DPL 656 384 27/09/1997
VCLDBX30.DPL 79 816 05/08/1997

Programy provozované pod RunTime je možné spustit dvěma způsoby. Buď se program pro spuštění zadá jako parametr při startu RunTime, případně spustí samotný RunTime a zadá se požadovaný program. V obou případech se zadávají soubory s příponou *.KLX.

Pro spuštění programu parametrem je vhodné zadat do operačního systému Windows asociaci pro soubory typu *.KLX. Díky tomu stačí pro spuštění programu kliknout na uvedený soubor v manažeru souborů, čímž dojde ke spuštění asociovaného RunTime s parametrem zadaného souboru. Pokud je zadán soubor v parametru, začne se ihned vykonávat zadaný program. Po jeho ukončení je ukončena současně i činnost RunTime. Po čtením pracovním adresářem je adresář, ve kterém se nachází zadaný program pro spuštění.

Je-li spuštěn samotný modul RunTime bez parametru, zobrazí se systémové okno RunTime s řadou tlačítek s následujícím významem:

- výběr spuštění programu
- zobrazení/ukrytí textového výstupního okna
- zobrazení/ukrytí grafického výstupního okna
- zobrazení/ukrytí databázového okna
- zobrazení informací o RunTime
- ukončení činnosti RunTime

Program se zadává po kliknutí na první tlačítko zleva. Pokud je na počítači instalováno vývojové prostředí systému KLONDAIK, nabídne se adresář, ve kterém se naposledy pracovalo. V opačném případě nabízí adresář, ve kterém je RunTime nainstalován. Po spuštění programu je systémové okno RunTime skryto a zobrazí se až po ukončení programu. Jméno souboru, ze kterého byl program spuštěn se následně zobrazí v seznamu spuštěných programů. Seznam je řazen abecedně a jsou do něj zaznamenávány pouze nové programy. Pokud se požaduje spustit program opakovaně, vybere se v seznamu a po kliknutí na tlačítko vpravo od seznamu bude program znovu spuštěn.

VYTVOŘENÍ *.KLX SOUBORŮ PRO RUNTIME

Soubory typu *.KLX se vytvářejí automaticky ve vývojovém prostředí registrovaného systému OZOGAN KLONDAIK. Soubor je vytvářen až po ukončení programu, pouze pokud neobsahuje žádné syntaktické chyby. V souborech typu *.KLX je uložen formulář i program. (*.KFM plus *.KL3). Je možné kompilovat i program, který nemá formulář. Pokud je v adresáři s programem nalezen soubor typu *.TXT stejného jména jako program, je také přilinkován do souboru typu *.KLX. Uvedený soubor bude v budoucnu sloužit například pro uložení dokumentace, kterou bude možné zobrazit. Soubor je uložen v nečitelném tvaru bez možnosti změny obsahu a může proto sloužit například jako digitální podpis autora programu. V budoucnu se uvažuje o možnosti načítat z uvedeného textového souboru uloženého v *.KLX uložený text, případně pouze zadané řádky k možnému použití přímov programu.

Překládá se pouze první program spuštěný z vývojového prostředí. Pokud se v programu používá skok do dalších formulářů nebo se spouští další samostatné programy, je nutné takové programy a formuláře zkompilovat samostatným spuštěním.

Program a formulář je v souboru typu *.KLX uložen v nečitelném tvaru bez možnosti změny zásahu neoprávněným uživatelem. Struktura souboru nebude nikde zveřejněna a nebudou také podávány jakékoliv informace týkající se způsobu uložení programu. Program přilinkovaný do souboru typu *.KLX není možné obnovit do původní podoby.

Ve vývojovém prostředí je pod spuštěním tlačítkem tlačítko dovolující spustit již zkompilovaný program.

KURZ - ZÁKLADY PROGRAMOVÁNÍ

- Lekce 1.01 - Instalace, spuštění a ukončení programu
- Lekce 1.02 - Základy ovládání programu
- Lekce 1.03 - Práce s příkazovým textovým výstupním oknem
- Lekce 1.04 - Příkaz WriteLn, matematické výpočty
- Lekce 1.05 - Používání proměnných
- Lekce 1.06 - Grafické výstupní okno, grafický editor
- Lekce 1.07 - Ovládání grafického okna z příkazového okna
- Lekce 1.08 - Výstup do grafického okna
- Lekce 1.09 - První program
- Lekce 1.10 - Editace programu
- Lekce 1.11 - Používání proměnných v programu
- Lekce 1.12 - Podmínky v programu, logické výrazy
- Lekce 1.13 - Programové cykly (FOR, REPEAT, UNTIL)
- Lekce 1.14 - Zadání vstupních hodnot, zobrazení výsledku
- Lekce 1.15 - Struktura programu
- Lekce 1.16 - Deklarace a používání procedur
- Lekce 1.17 - Deklarace a používání funkcí
- Lekce 1.18 - Návrh a zápis programu
- Lekce 1.19 - Typ ARRAY - proměnná typu pole
- Lekce 1.20 - Knihovna pro zpracování řetězců
- Lekce 1.21 - Knihovna pro práci se soubory a adresáři
- Lekce 1.22 - Spuštění dalších programů (EXE, KL3)
- Lekce 1.23 - Co je to databáze, z čeho se skládá
- Lekce 1.24 - Ovládání databázového okna
- Lekce 1.25 - Navigace v databázi
- Lekce 1.26 - Zadávání výrazů v jazycích xBase
- Lekce 1.27 - Čtení a zápis dat databáze
- Lekce 1.28 - Nastavení filtru, podmínky xBase
- Lekce 1.29 - Sekvenční prohledávání databáze
- Lekce 1.30 - Třídění databáze - indexy
- Lekce 1.31 - Hledání v databázi dle indexů
- Lekce 1.32 - Rušení záznamů v databázi
- Lekce 1.33 - Databázové oblasti
- Lekce 1.34 - Vytvoření nové databáze
- Lekce 1.35 - Změna struktury databáze
- Lekce 1.36 - Editace memo položek databáze
- Lekce 1.37 - Zpracování memo položek z programu
- Lekce 1.38 - Síťové zpracování databází
- Lekce 1.39 - Tisk sestav z databází (reporty)
- Lekce 1.40 - Ovládání oken z programu
- Lekce 1.41 - Spuštění dalšího programu
- Lekce 1.42 - Animace programu

Úvod:

Pro ty, kteří dosud nikdy neprogramovali a nevíte se už zdát pojem programování záhadný a tajemný. Jedná se však o zcela běžnou věc, kterou vlastní všichni známe již od dítství. Určiti jste si již mnohokrát řekli: "ráno půjdu do školy (práce) , odpoledne na koupališti a veer se budu dívat na televizi". Tím jste si sestavili program dne. Definovali jste posloupnost akcí, které provedete. Můžete si však také stanovit podmínky programu. Například pokud bude odpoledne hezky, půjdete na koupališti, jinak (bude pršet) si budete est nebo se uřit. V některých případech si můžete stanovit opakování některých akcí. Například že se budete uřit tak dlouho, dokud to nepochopíte. Jak vidíte, programovat už umíte. Sestavujete si program dne, výuky, dovolené. Nyní už zbývá pouze nauřit se převést program do podoby, kterou zvládne i počítač.

Programování v systému KLONDAIK je velmi jednoduché. Systému zadáváte své požadavky na vzhled vyvíjené aplikace a úkoly, které má systém provést při výskytu očekávaných událostí. K tomu je možné použít moderní vizuální postupy, které Vám umožní bez jakýchkoli předchozích znalostí objektové programování.

Počítačový program představuje definici posloupnosti akcí, podmínky jejich provedení a opakování. Aby byl počítač schopen požadované činnosti provádět, musíte mu je zadat v podobě, které on rozumí. Pokud budete chtít angličtinou napsat program dne, budete mu jej muset přeložit do angličtiny. Stejně tak musíte přeložit program ze slovní podoby do počítačového jazyka. Počítač potom bude číst jednu řádku programu za druhou a vykonávat to, co jste mu zadali.

Lekce 1.01 - Instalace a spuštění programu

Program se dodává na dvou disketách formátu 3,5" HD. Instalace se spustí z diskety příkazem SETUP.EXE. Po spuštění musíte zadat nejprve adresář pro instalaci programu. Standardně je pro instalaci nastaven adresář C:\OZOKLON3\. Cílový disk i adresář je možné změnit, přesto však doporučujeme zachovat předdefinované jméno adresáře. Při instalaci se nekopírují žádné soubory mimo zadaný adresář. Současně se nainstaluje i odinstalace programu.

Pro rozlišení přesné definice programu OZOGAN KLONDAIK od editovaného a laděného programu bude v následujícím textu u všech lekcí uváděn program OZOGAN KLONDAIK jako systém a pod pojmem program se bude rozumět laděný a editovaný uživatelův program.

Systém se startuje po dokončení instalaci kliknutím na systémové tlačítko Windows Start s následným výběrem programu. Systém KLONDAIK je také možné spustit z adresáře zadaného při instalaci spuštěním souboru KLONDAIK.EXE. Při prvním spuštění systému je integrované uživatelské prostředí nastaveno do standardního výchozího stavu.

Program je možné ukončit buď volbou z menu Soubor/Konec programu, případně ikonou z toolbaru. Pokud je editován program a změny nebyly dosud uloženy, budete dotázáni systémem na uložení změn. Při ukončení programu se provede uložení rozložení ikon integrovaného prostředí do registrů systému Windows. Při dalším spuštění programu bude potom obnoven stav rozložení jednotlivých ikon na pracovní ploše systému.

Pokud je možné systém KLONDAIK výše uvedeným způsobem nainstalovat a spustit, pokračujte studiem následující lekce. Jinak si prostudujte následující text, ve kterém jsou nastíněny možné problémy spolu s možnostmi řešení pro odstranění problémů.

Systém KLONDAIK nevyžaduje žádné speciálně instalované podpůrné knihovny. Pro svou činnost vyžaduje pouze následující soubory, které jsou součástí instalace:

```
-----  
soubor délka datum  
-----  
SDE32.DLL 131 072 04/05/1998  
SDECDX32.DLL 264 192 27/09/1997  
VCL30.DPL 1 287 088 23/10/1997  
VCLX30.DPL 219 584 05/08/1997  
VCLDB30.DPL 656 384 27/09/1997  
VCLDBX30.DPL 79 816 05/08/1997  
-----
```

DLL knihovny se používají pro databázový přístup, soubory *.DPL jsou tzv. runtime knihovny Delphi. Uvedené soubory je možné ponechat v adresáři dle instalace. V případě jakýchkoliv problémů zkontrolujte, zda neexistují uvedené soubory v adresáři \WINDOWS\SYSTEM\. Pokud ano, může docházet ke vzájemným kolizím s jinými programy. V takovém případě pokuste soubory přesunout do adresáře aplikace, která je využívá. Těžko lze však nalézt obecně použitelné řešení. Bylo opakovaně prověřeno, že pokud se systém KLONDAIK instaluje na počítač se správně instalovaným operačním systémem Windows, nenastanou žádné problémy.

Následující lekce

Lekce 1.02 - Základy ovládání programu

Systém OZOGAN KLONDAIK je aplikace Windows a proto je ovládání programu podřízeno tomuto standardu. Systém je ovládán z menu, často používané volby jsou přitom přístupné i z toolbaru pomocí ikon, případně pomocí funkčních kláves. Veškeré akce probíhají v několika oknech s přesně definovaným určením. Celá práce se systémem probíhá v několika režimech dle kterých jsou využívána příslušná okna. Okna lze na ploše obrazovky uspořádat podle aktuální potřeby.

Nejčastěji používané okno je okno s programem, které obsahuje vyvíjený a laděný program. V dalším, příkazovém okně je možné zadávat přímo příkazy jazyka bez nutnosti spuštění programu. Výsledek programu je možné zobrazit buď v textovém nebo grafickém výstupním okně. V dalším okně je možné pracovat s databázemi ve formátu FoxPro. Některá okna mají svou vlastní sadu ikon umístěných v toolbaru, který se zobrazí po aktivaci okna v jeho horní liště.

Okna můžete po pracovní ploše libovolně přepínat, posouvat, zvětšovat je či zmenšovat. Přepínání oken je možné kliknutím na libovolnou viditelnou část okna. Po kliknutí se stane příslušné okno aktivní. Pokud není okno viditelné, použijte menu Okna, kde zvolíte požadované okno. V dalších lekcích se naučíte, že je možné zviditelnit požadované okno i příkazem z programu. Kliknutím na příslušnou ikonu v pravém horním rohu okna je možné okno maximalizovat, případně minimalizovat. Okno se přesouvá nejlépe myší, kdy je uchopíte za horní lištu a přesunete na požadované místo. Změny velikosti okna dosáhnete po uchopení pravého dolního rohu okna a nastavení na požadovanou velikost.

Systém obsahuje kontextovou nápovědu, která je dostupná v několika stupních. Při výběru volby z menu je krátká nápověda k vybrané volbě zobrazována ve stavovém řádku v dolní části systému. Tzv. bublinová nápověda je zobrazována po najetí myší na ikonu v toolbaru. Nejobtížnější nápověda se zobrazí po stisku klávesy F1. Jedná se o hypertextový dokument s provázanými odkazy na zvolená hesla. Zde uváděné příklady programů máte možnostsi do systému z nápovědy přetáhnout pomocí blokového přesunu přes schránku Windows. To provedete tak, že najedete v nápovědi myší na začátek ukázky programu, stisknete levé tlačítko myši, držíte jej stisknuté a přetáhnete myš na konec textu příkladu. Tam tlačítko myši uvolníte. Stiskem kláves Ctrl+C se převede takto označený blok textu do schránky Windows. Nyní se přepnete do okna s programem a stiskem kláves Ctrl+V převedeme obsah schránky.

K ovládání programu je vhodné používat myš. Většina voleb je sice dostupná současně z menu i myší, přesto je však ovládání myší mnohem příjemnější a operativnější. Pokud se v dalším textu používá myš, předpokládá se levé tlačítko.

Následující lekce

Lekce 1.03 - Práce s příkazovým oknem

Nejprve si ukážeme, jak se dají velmi jednoduše zadávat systému příkazy bez zápisu programu. Najděte na pracovní ploše okno nazvané příkazy a udeřte je aktivním - klikněte na něj myší. Pokud není okno viditelné, případně minimalizované, můžete zvolit z menu volbu Okna/Příkazy nebo stiskem klávesy Ctrl+F2. Upravte myší velikost okna tak, aby zabíralo levou polovinu obrazovky. Obdobným způsobem nastavte okno Výstup text do pravé poloviny obrazovky. Přejděte zpět do příkazového okna.

Okno příkazy se podobá textovému editoru. Má však velmi důležitou vlastnost. Pokud napíšete v okně libovolný text a stisknete klávesu ENTER, pokusí se systém ihned napsaný řádek vykonat jako známý příkaz. To znamená, že systém si přečte text řádku, na kterém stál kurzor a zjistí, zda se jedná o jemu známý příkaz. Pokud ano, příkaz provede. Proto se toto okno nazývá příkazové.

Příkazy a příkazy jsou pro systém KLONDAIK instrukce, co má vykonat. Příkazem je většinou označen příkazový příkaz systému. Jako příkazy jsou označovány doplňkové příkazy, které jsou zabudovány jako podpůrné akce. Příkazy a příkazy je vhodné psát malými písmeny s velkými písmeny na počátku slova. Pokud jsou příkazem názvy příkazů uvedeny z více slov, nesmí být mezi nimi uvedeny mezery.

Napište nyní v příkazovém okně slovo BEEP a stisknete klávesu ENTER. Systém zjistí, že příkaz BEEP je příkazem, pro pípnutí. Zajistí proto, že reproduktor přehraje krátce pípnutí. Takto systém vykoná vami zadaný příkaz do podoby srozumitelné pro přehraze.

V příkazovém okně máte možnost vyzkoušet si mnoho příkazů a příkazů systému KLONDAIK, aniž napíšete jedinou řádku programu. Pokud uděláte chybu, systém vás na to upozorní a vy máte možnost ihned chybu opravit bez nutnosti opakovaně kompilace programu. Zapomenete-li například napsat ve výše uvedeném příkazu BEEP jedno 'E', oznámí vám systém, že uvedený příkaz je neznámý. Najedete proto kurzorem na chybné místo, provedete opravu a po stisku klávesy ENTER se již příkaz vykoná.

Jistě jste si již všimli, že klávesa ENTER neukončí na místě kurzoru řádek, jak to bývá u textových editorů. Řádek zůstane celý zachován a kurzor se automaticky přesune na začátek prázdného řádku za předchozí příkaz. Pokud zkusíte zapsat příkaz

```
Writeln("KLONDAIK");
```

vypíše se do textového výstupního okna text KLONDAIK. Možnost příkazu Writeln je vlnována následující lekcí. Nyní bude stačit, pokud si zapamatujete, že příkaz Writeln v uvedeném tvaru vypíše obsah textu uvedeného mezi uvozovkami do textového výstupního okna. Mezi uvozovky můžete zkusit příkazem zadat libovolný jiný text. Zkuste text mezi uvozovkami několikrát změnit a změny vždy odešlete klávesou ENTER. Jistě si všimnete, že původní text před opravou zůstane v příkazovém okně zachován a příkaz novým textem bude doplněn na konec příkazového okna. Díky tomu máte vždy přehled o historii zadávaných příkazů. Neprovedete-li v textu žádnou změnu a příkaz opakovaně odešlete, nebude stejný příkaz zadávaný bez změny za sebou v historii příkazového okna za sebou znovu opakován.

Historii zadávaných příkazů můžete libovolně používat bez ohledu na pořadí původního zadávání. Pokud provedete v některém řádku příkazového okna libovolnou změnu a následně stisknete klávesu ESC, příkaz se neprovede a text upraveného řádku bude uveden do původního stavu. Příkazy vyzkoušené v příkazovém okně máte možnost přeshránu Windows následně převést do programu.

Různými pokusy s příkazovým oknem jste dosáhli toho, že máte ve výstupním textovém okně uvedeny různé texty. S každým použitím příkazu Writeln přibudena konec okna jeden řádek.

To připomíná použití psacího stroje, kdy také není možné se vracet zpět. V dávných dobách prahistorie výpočetní techniky sálových počítačů se podobné psací stroje používaly pro komunikaci s počítačem. Tehdy se jim říkalo konzole - anglicky console. Uvedené anglické označení je proto také používáno pro příkaz přímo související s obsluhou textového výstupního okna. Pokud například zadáte v příkazovém okně příkaz `ConsoleClear`, zjistíte, že zruší obsah textového výstupního okna. Obdobného efektu dosáhnete, pokud se přepnete do textového výstupního okna a stisknete v podřízeném toolbaru uvedeného okna ikonu znázorňující prázdnou stránku. Po kontrolním dotazu bude obsah textového výstupního okna zrušen.

V následující kapitole se seznámíte, jak vypisovat v textovém výstupním okně za použití příkazu `WRITELN` i jiné informace a hodnoty.

Následující lekce

Lekce 1.04 - Příkaz Writeln, matematické výpočty

V předchozí kapitole jste se naučili zadávat z příkazového okna příkazy umožňující vypsát do textového výstupního okna znaky. Nyní si ukážeme, jak vypisovat čísla a aritmetické výpočty.

S příkazem `Writeln` jste již měli možnost se krátce seznámit. Používá se na výpis hodnot. Požadovaná hodnota se přitom uvádí v závorce. Pokud se u příkazu `Writeln` uvedou pouze prázdné závorky, nebude nic vypsáno a další výstup do textového výstupního okna bude prováděn na novou řádku. `Writeln` je pouze příkaz, který uvádí systému, že má něco vypsát. Uvedené něco se přitom uvádí v závorce jako parametr příkazu. Pokud uvedete znaky v uvozovkách nebo apostrofech, převedou se uvedené znaky do textového výstupního okna. Pokud byste chtěli vypsát číslo, musíte zadat příkaz následujícím tvarem:

```
Writeln(10);
```

Stačí tedy napsat jako parametr příkazu do závorek požadované číslo. Číslo se vypíše se zarovnáním od levého kraje okna. Obdobným způsobem můžete zadat i jednoduchý matematický výraz, který se vypočítá a zobrazí se výsledek. Ve výrazu můžete samozřejmě použít závorky pro definici postupu výpočtu:

```
Writeln((17-5)*3);
```

S možností výpisu numerických údajů, tedy znaků jste se již seznámili v předchozí lekci. Seskupení několika znaků se přitom nazývá řetězec také tak, již bude v následujícím textu uváděno. Pokud byste chtěli jedním příkazem `Writeln` vypsát najednou řetězec znaků, můžete tak učinit uvedením několika parametrů oddělených čárkou:

```
Writeln('Výsledek výpočtu je:', 10/3);
```

Stejně tak můžete vypsát samozřejmě i několik čísel nebo řetězců oddělených v příkazu čárkami. To vám dovolí ve spojení s možností uvádění délky čísel vhodnou grafickou úpravu textu:

```
Writeln('Seznam výsledků', 12/3, " ", 45-26, " ", 3*3);
```

Příkaz `Writeln` provede po svém ukončení vždy přechod na novou řádku. Pokud však budete potřebovat, aby další výpis pokračoval na stejné řádce, můžete použít příkaz `Write`. Ten neprovádí ukončení řádku, následující výpis je zahájen od pozice ukončení příkazu `Write`. Možnosti a parametry příkazu `Write` jsou přitom totožné jako pro uváděný popis příkazu `Writeln`.

Výše uvedený popis použití příkazu `Writeln` se vám může zdát na první pohled složitý. Je však nutné jej přesně dodržet. Systém si hlídá důsledně jeho dodržování a v případě chyby odmítne příkaz vykonat. Přesná definice jakéhokoli používaného jazyka se nazývá syntaktická pravidla. Pokud zadá uživatel chybný zápis, dojde k porušení syntaxe a odmítnutí systému k vykonání chybného zadání. Byla proto sestavena pravidla pro použití všech příkazů. Uvádíme zde pro představu zkrácenou definici syntaxe zápisu příkazu `WRITELN` tak, jak byla výše popsána:

```
Writeln( [ výraz ] [, výraz] [...] );
```

V zápise znamená výraz libovolný výraz, například řetězec, číslo nebo výpočet. Údaje hranatých závorek se nemusí uvádět. Proto je možné dle výše uvedeného popisu uvést jeden, nebo několik výrazů. Druhý a další výraz ale musí být od předchozího výrazu oddělen čárkou. Pokud si dobře prohlédnete uvedený syntaktický zápis zjistíte, že mezi závorkami nemusíte nic uvádět. To je dáno levou hranatou závorkou uvedenou ihned za levou kulatou

závorkou. Ukončení této volitelné části je provedeno pravou hranatou závorkou před pravou kulatou závorkou. Pokud proto uvedete příkaz:

```
Writeln();
```

nevypíše se žádný text, dojde ale k vynechání prázdného řádku. Všimněte si, že je na konci každého řádku uveden středník. Středníkem se oddílují jednotlivé příkazy. Pokud vypisujete příkaz do příkazového okna, není jej sice nutné uvádět, ale bude vhodné, pokud si na to zvyknete. Uvedený způsob popisu syntaxe je použit u popisu všech příkazů a knihoven procedur a funkcí uvedených v manuálu i helpu k programu.

Pokud jste se dokonale seznámili s použitím příkazu `Writeln`, můžete ve spolupráci s manuálem k programu vyzkoušet některé funkce matematické knihovny. Dále uvedené příklady uvádí na konci řádku mezi složenými závorkami komentáře, které nemají na výsledek žádný vliv. Ve skutečnosti je proto nemusíte většinou složených závorek uvádět.

```
Writeln(Abs(-55)); {absolutní hodnota čísla}
Writeln(Cos(PI)); {kosinus Ludolfova čísla PI}
Writeln(Max(3,10)); {maximální hodnota zadaných čísel}
Writeln(Min(3,10)); {minimální hodnota zadaných čísel}
Writeln(Random(500)); {náhodné číslo do 500}
Writeln(Round(12.82)); {zaokrouhluje číslo}
Writeln(Sqr(5)); {vrací druhou mocninu čísla}
Writeln(Sqrt(16)); {vrací druhou odmocninu čísla}
Writeln(Trunc(12.82)); {odřízne desetinou část čísla}
```

V některých případech může vadit, že výsledek matematického výpočtu je uveden většinou plného počtu desetinných míst podle typu matematického výpočtu. Pokud by to někdy vadilo, je možné použít funkci RealRound, která zaokrouhluje číslo s uvedením počtu desetinných míst, která mají být zachována. Dále uvedený příklad zaokrouhlí výpočet druhé odmocniny na dvě desetinná místa:

```
Writeln(RealRound(Sqrt(5),2));
```

Pokud by vám při výpisu výsledků matematických výpočtů vadilo, že jsou čísla vypisována bez mezer vlevo a vždy plnou délku desetinných míst, můžete použít funkci RealToStr pro převod čísla na znaky (četítec). Jako parametr funkce se přitom uvádí číslo pro převod, celkový počet znaků (většinou počet mezer a desetinné části) a počet desetinných míst. Vyzkoušejte si proto například následující výpočty a porovnejte zobrazené výsledky:

```
Writeln(RealToStr(10/3,6,0));
Writeln(RealToStr(10/3,6,1));
Writeln(RealToStr(10/3,6,2));
Writeln(RealToStr(10/3,6,3));
```

Jak vidíte, můžete použít příkaz `Writeln` jako docela chytrou kalkulačku. Zatím ale neznáte, jak by bylo možné uložit vypočtené hodnoty do 'paměti' naší kalkulačky k dalšímu použití. S tím se seznámíme až v některé z následujících lekcí. V příští lekci se seznámíte s dalším výstupním oknem, které se používá pro výstup a kreslení grafiky.

Následující lekce

Lekce 1.05 - Používání proměnných

Když jsme si ukazovali v minulé lekci, jak vypsat příkazem `Writeln` číslo do výstupního textového okna, bylo uvedeno, že není možné jednoduchým způsobem uložit výsledek výpočtu z příkazového řádku do paměti. Je to proto, že je k tomu nutné použít nový příkaz `SysVarSet`. Ten umožňuje deklarovat v paměti místo pro uložení hodnoty. Nic jako paměť na kalkulačce. V našem případě musíte na rozdíl od kalkulačky paměťové místo pojmenovat. Protože je možné takto ukládat hodnoty, které je možné v dalších výpočtech zmínit, nazývají se uvedené paměti "proměnné". Proto se musí při zadávání nové proměnné uvádět mimo jejího názvu i počáteční hodnota. Zkuste si nadefinovat proměnnou pro uložení ceny a počtu kusů:

```
SysVarSet('cena', 123.45);  
SysVarSet('pocet', 10);
```

Pro zjištění hodnoty definované systémové proměnné z příkazového okna můžete následně použít buď přímo její název, nebo je možné použít příkaz `SysVarGet`.

```
Writeln(SysVarGet('cena') );  
Writeln(cena*pocet);
```

Je již také možné přiřadit proměnné novou hodnotu:

```
cena := 999;
```

Proměnné deklarované příkazem `SysVarSet` zůstávají v paměti až do ukončení systému KLONDAIK. Je možné je proto používat pro všechny následující výpočty. Pro zobrazení hodnot systémových proměnných je možné použít příkaz `SysVarShow`. Kompletní výmaz všech systémových proměnných se provede příkazem `SysVarClear`.

Na používání proměnných je založená i možnost programování. V některé z dalších lekcí se seznámíte s tím, že proměnné mohou být určitého typu pro uložení různých hodnot. Proměnné bude také možné používat pro předávání hodnot mezi programy. Systémové proměnné budete také moci používat pro uložení výsledků výpočtů i po ukončení programu. Se všemi uvedenými možnostmi se podrobněji seznámíte v dalších lekcích používání programu.

Následující lekce

Lekce 1.06 - Grafické výstupní okno

V předchozích kapitolách jsme se naučili používat příkazové okno a textové výstupní okno. Systém ale obsahuje i další, velmi zajímavé výstupní okno. Je to grafické výstupní okno, které vám umožní své plochy libovolně kreslit.

Uzavřete dříve používané textové výstupní okno. Můžete tak učinit kliknutím na uzavírací ikonu, případně můžete zadat v příkazovém okně povel `ConsoleHide`. Dále aktivujte grafické výstupní okno. To lze provést z menu volbou `Okna/výstup grafika`. Okno je možné také zobrazit povel `ImageShow`. Zobrazené okno upravte opět tak, aby pokrývalo pravou polovinu obrazovky.

Jak vidíte, každé okno je pro účely ovládnutí z programu pojmenováno významově dle anglického názvosloví. Existují příkazy pro zobrazení (anglicky `show`) a ukrytí (anglicky `hide`) okna. Možná se pozastavujete nad tím, že u českého programu se používá cizojazyčných termínů. Je to proto, že systém `OZOGAN KLONDAIK` může posloužit jako nástroj pro výuku programování. Proto je podle nás vhodné si již od počátku zvyknout na terminologii, která se v obličejném programování používá. Názvy procedur, funkcí a konstant přitom vychází z používané terminologie jazyků `Pascal` a získané znalosti jistě dále využijete. Použitá počítačová angličtina je přitom velmi jednoduchá a neměla by nikomu dělat problémy.

Po aktivování grafického výstupního okna se vám v jeho horní liště zobrazí řada ikon pro ovládnutí zabudovaného grafického editoru a ikony pro možnost kreslení na grafické ploše. Ikony jsou umístěny do tří skupin. Levá skupina se používá pro načítání a ukládání obrázků v grafickém formátu `*.BMP`. Střední skupina slouží pro definici kresleného tvaru a v pravé skupině naleznete nastavení typu čáry a výplně ploch.

Nejprve se seznámíme s možností kreslení základních geometrických tvarů. Ve střední skupině ikon jsou seskupena tlačítka pro čáru, obdélník, kružnici a obdélník se zakulacenými hranami. Klikněte nejprve na tlačítko s čarou. Tím jste zadali, že budete chtít kreslit čáry. Uvidíte, že tlačítko zůstalo stisknuto. Přesunutím ukazatele myši na grafickou plochu. Pokud nyní na grafické ploše stisknete tlačítko myši, podržte jej a přesunete na novou pozici, bude se kreslit čára z bodu stisku tlačítka do aktuální pozice ukazatele myši. Po uvolnění tlačítka zůstane čára zachována. Zkuste si nakreslit i vodorovné a svislé čáry. Obdobným způsobem lze na grafickou plochu kreslit obdélníky, kružnice a obdélníky se zakulacenými rohy. Pro zvolení nového tvaru musíte stisknout příslušné tlačítko na liště s ikonami.

Čáry se kreslí černou barvou a tenkou čarou. Pokud budete chtít změnit barvu čáry, nebo její tloušťku, stiskněte ikonu v pravé horní části okna s vyobrazením tužky. Zobrazí se vám další řada ikon pro zadávání barvy, typu a tloušťky čáry. Po opětovném stisku ikony s tužkou se nastavení skryje. Ikona tedy pracuje jako přepínací tlačítko. Klikněte si proto na tlačítko tak, aby jste měli zobrazeny ikony pro nastavení čar. Vlevo je umístěna tabulka barev, uprostřed tlačítko pro výběr typu čáry a vpravo můžete zadat tloušťku čáry. Barvy se vybírají kliknutím na požadovanou barvu. Vybraná barva je označena dvoupísmennou anglickou zkratkou barvy. Typ čáry můžete vybrat plnou čarou, tečkovanou, čárkovanou, čerchovanou nebo můžete posledním tlačítkem úplně zrušit. Tloušťka čáry se udává v bodech. Pro zřetelnou tloušťku můžete zapsat do editačního boxu příslušnou hodnotu, nebo můžete použít pro nastavení šířky. Pokud bude nastavena tloušťka čáry větší než jedna, neuplatní se zadaný typ čáry a čára se bude vykreslovat vždy plná, případně nebude kreslit vůbec. Zkuste si nastavit parametry čáry a prověřte si účinky změny příkreslení základních geometrických tvarů.

Dosud jsme kreslili pouze okraje geometrických tvarů. Standardně je totiž nastaveno, že se plocha kreslených geometrických tvarů nevykresluje. Podobným způsobem, jako se nastavují parametry kreslení čar, máte možnost nastavit parametry vykreslování ploch. Ikony pro nastavení se zobrazí po stisku tlačítka s vyobrazením štětce. Opět máte možnost zadat barvu, tentokrát plochy (výplně). Současně můžete zadat typ výplně. Máte možnost vybrat si buď

kreslení plné plochy zadanou barvou, nevykreslování plochy, nebo z několika druhů vykreslení plochy čarami. Lze vybrat čáry vodorovné, svislé, vodorovné i svislé současně (mřížka) a různé druhy diagonálních čar. Tloušťka čar vyplňuje pětimvždy jeden bod.

Nyní jste již schopni nakreslit na grafické ploše pouze za použití myši jednoduché obrázky. Vyzkoušejte si různé možnosti nastavení čar a plochy. Parametry nastavení čar a plochy je možné změnit samozřejmě také přímo z programu, případně zadat z příkazového okna. To si však ukážeme až v následujících lekcích. Nyní bude pro vás jistě zajímavá možnost uložení nakreslených obrázků na disk do souboru pro pozdější použití a zpětná možnost načtení obrázku z disku do grafického okna pro provedení úprav. Ukážeme si také, jak je možné obrázek vytisknout.

Pokud máte grafickou plochu zaplněnou předchozími pokusy, můžete provést její výmaz pomocí ikony s obrázkem prázdné stránky. Ikona se nachází v levé skupině ikon umístěné v grafickém výstupním okně. Po kontrolním dotazu bude grafická plocha vymazána. Stejněho efektu lze dosáhnout povelem `ImageClear` zadaným v příkazovém okně. Povel se zadává bez parametrů, v tomto případě provede výmaz grafické plochy již bez kontrolního dotazu.

Nakreslené obrázky máte samozřejmě možnost uložit do souboru. Používá se známý a běžný bitmapový soubor typu *.BMP. Pro uložení stiskněte ikonu s obrázkem diskety. V dialogovém okně zadejte adresář a jméno souboru. Obrázek můžete uložit samozřejmě i povelem `ImageSave` z příkazového okna. Například:

```
ImageSave("obrazek.bmp");
```

Obrázky můžete i zpět načíst. Použijte ikonu s obrázkem šipky směřující do stránky. V dialogovém okně vyberte adresář a jméno souboru typu *.BMP. Načíst můžete i obrázky vytvořené v jiných grafických systémech. Po načtení obrázku je velikost grafické plochy nastavena dle načítaného obrázku. Obrázek můžete načíst i povelem z příkazového okna `ImageLoad`. Například:

```
ImageLoad("obrazek.bmp");
```

Pokud byste chtěli vytvořený obrázek vytisknout, můžete tak učinit pomocí ikony zobrazující tiskárnu. Pro tisk obrázku z příkazového okna můžete použít povel `ImagePrint`.

V následující lekci se seznámíte s možnostmi ovládání grafického výstupního okna pomocí povelů z příkazového okna.

Následující lekce

Lekce 1.07 - Ovládání grafického okna

Předchozí lekce vás seznámila krátce se základními možnostmi použití grafického výstupního okna. Naučili jste se kreslit do grafické plochy pomocí myši, ukládat, načítat a vytisknout vytvořený obrázek. Nyní se seznámíte s dalšími možnostmi grafického okna a s parametry okna, které budete potřebovat pro programování grafických povelů. Poznáte, že vše, co bylo možné nastavit v grafickém okně budete mít možnost zadat pomocí povelů z příkazového okna.

Činnost grafického okna je možné si představit jako malířské plátno definovaných rozměrů, na které se kreslí perem (anglicky pen). Větší plochy je možné vybarvit štětcem (anglicky brush). Kreslí se přitom vždy nastavenou barvou. Systém obsahuje povely pro nastavení parametrů pera i štětce. S možnostmi se seznámíte v následujícím textu.

Přikreslení obrázků do grafického okna se kreslí obrázce čárou, jejíž typ je definován povelom ImagePenStyle a tloušťka čáry je definována povelom ImagePenWidth. Plocha nakreslených geometrických obrázků je vyplněna stylem zadaným povelom ImageBrushStyle.

Nejjednodušší je zmínit tloušťku čáry, kdy uvádíte přímo jako parametr povelu tloušťku čáry v bodech. Pro zmínku tloušťky čáry na píť bodů použijetez příkazového okna následující povel:

```
ImagePenWidth(5);
```

Všimněte si, že pokud nastavíte sílu čáry povelom z příkazového okna, použijete nastavená síla čáry i pro následné kreslení pomocí myši přímo v grafickém okně. Obdobná vlastnost je platná i pro nastavení všech parametrů grafického okna. Máte proto možnost libovolně kombinovat zadávání povelů z příkazového okna nebo jejich nastavení pomocí ikon. V budoucnu budete mít samozřejmě možnost uvedené parametry nastavit i přímo z programu.

Styl čáry máte možnost zadávat povelom ImagePenStyle. Předdefinováno je šest stylů. Jako parametr povelu musíte přitom zadat definovaný styl čáry. Parametr můžete uvést buď číselnou hodnotou, nebo jménem konstanty dle následující tabulky:

```
-----  
hodnota konstanta název stylu  
-----
```

```
1 psSolid souvislá čára  
2 psDash přerušovaná čára  
3 psDot tečkovaná čára  
4 psDashDot čerchovaná čára  
5 psDashDotDot čerchovaná čára se dvěma tečkami  
6 psClear neviditelná čára  
-----
```

Příklady použití definice stylu čáry (text mezi složenými závorkami nemusíte psát, jedná se o poznámku) :

```
ImagePenStyle(1); { souvislá plná čára }  
ImagePenStyle(3); { tečkovaná čára }  
ImagePenStyle(6); { neviditelná čára }  
  
ImagePenStyle(psSolid); { souvislá plná čára }  
ImagePenStyle(psDot); { tečkovaná čára }  
ImagePenStyle(psClear); { neviditelná čára }
```

Existuje samozřejmě způsob nastavení barvy čáry. Nejprve se však budeme muset seznámit s možností použít barev v počítači. Zobrazování barev závisí na vlastnostech videokarty ve vašem počítači. Používáte-li barevnou VGA kartu, máte možnost zobrazit minimálně 16 barev. Po příslušném nastavení videoadaptéru je možné běžně zobrazovat 256 barev, výjimečně i více. Vy budete mít možnost nastavit v systému libovolnou barvu z rozsahu 16 milionů barev. Skutečně zobrazená barva ale závisí na vlastnostech technického zařízení, protože se zobrazí vždy barva nejbližší.

Hodnotu barev je možné zadávat dvěma způsoby, které lze v programu libovolně kombinovat. Pokud budete používat pouze základní, šestnáctibarevnou paletu, můžete tak učinit zadáváním předdefinované konstanty udávající anglické jméno barvy. Stejnou barvu máte možnost zadat i pomocí tzv. RGB hodnoty.

Hodnota barev zadávaná definicí RGB znamená, že každá barva je definována jako poměr kombinace barev modré, zelené a červené. Pro každou barvu je možné volit hodnoty v rozsahu 0 až 256. Násobek těchto hodnot (modrá x zelená x červená) udává výslednou barvu. Výhodné je používat tzv. hexadecimálního zápisu, kdy jsou pro každou barvu vyhrazeny dvě pozice čísla s hodnotami od 00 (číslo 0) až do FF (číslo 256). Při použití hexadecimálního čísla je nutné uvést před číslem rozlišovací znak \$ a číslo musí začínat vždy číslicí (ne písmenem). Viz tabulka hodnot barev:

hodnota	konstanta	název barvy
\$000000	clBlack	černá
\$000080	clMaroon	kaštanově červená
\$0000FF	clRed	světle červená
\$008000	clGreen	tmavě zelená
\$008080	clOlive	tmavě žlutá
\$00FF00	clLime	světle zelená
\$00FFFF	clYellow	žlutá
\$800000	clNavy	tmavě modrá
\$800080	clPurple	tmavě fialová
\$808000	clTeal	tmavě modrozelená
\$808080	clDkGray	tmavošedá
\$0C0C0C	clLtGray	světle šedá
\$0FF0000	clBlue	modrá
\$0FF00FF	clFushsia	fialová
\$0FFFF00	clAqua	modrozelená
\$0FFFFFF	clWhite	bílá

Příklady použití definice barvy čáry (text mezi složenými závorkami nemusíte psát, jedná se o poznámku) :

```
ImagePenColor(clWhite); { bílá barva čáry }
ImagePenColor(clRed); { světle červená barva čáry }
ImagePenColor(clBlue); { modrá barva čáry }

ImagePenColor($0FFFFFF); { bílá barva čáry }
ImagePenColor($0000FF); { světle červená barva čáry }
ImagePenColor($0FF0000); { modrá barva čáry }
```

Podobně, jako je možné nastavit parametry kreslené čáry je možné nastavit parametry vykreslovaných ploch geometrických obrazců. Barva plochy se přitom zadává příkazem `ImageBrushColor`. Používá se přitom výše zadaných hodnot a konstant pro definici parametru barvy. Příklady použití definice barvy plochy (text mezi složenými závorkami nemusíte psát, jedná se o poznámku) :

```
ImageBrushColor(clWhite); { bílá barva plochy }
ImageBrushColor(clRed); { světle červená barva plochy }
ImageBrushColor(clBlue); { modrá barva plochy }
```

```
ImageBrushColor($0FFFFFF); { bílá barva plochy }
ImageBrushColor($00000FF); { světle červená barva plochy }
ImageBrushColor($0FF0000); { modrá barva plochy }
```

Pro nastavení stylu vyplňování ploch se používá povel ImageBrushStyle, kde se jako parametr povelu udává buď číslem definovaný styl, nebo jméno konstanty dle následující tabulky:

```
-----
hodnota konstanta název stylu
-----
1 bsSolid vyplní oblast jednou barvou
2 bsClear vyplní oblast barvou pozadí
3 bsHorizontal vyplní oblast vodorovnými čarami
4 bsVertical vyplní oblast svislými čarami
5 bsFDiagonal diagonální čáry \\\\\\\
6 bsBDiagonal diagonální čáry //\\\\\\
7 bsCros vodorovné a svislé čáry
8 bsDiagCross vodorovné a svislé čáry diagonální
-----
```

Příklady použití definice stylu plochy (text mezi složenými závorkami nemusíte psát, jedná se o poznámku) :

```
ImageBrushStyle(1); { plné vybarvení plochy }
ImageBrushStyle(3); { výplň vodorovnými čarami }
ImageBrushStyle(8); { výplň diagonálními čarami }
```

```
ImageBrushStyle(bsSolid); { plné vybarvení plochy }
ImageBrushStyle(bsHorizontal); { výplň vodorovnými čarami }
ImageBrushStyle(bsDiagCros); { výplň diagonálními čarami }
```

Aby bylo možné kreslit v grafickém okně do přesně určených pozic, musí být zadány souřadnice pro kreslení. Souřadnice určují polohu jednotlivých bodů kresby. Souřadnice znamená, že musíte uvést vzdálenost v bodech od levého okraje grafického okna a vzdálenost v bodech od horního okraje grafického okna. Souřadnicový systém je tedy vztažen k levému hornímu rohu, který má souřadnici 0,0. Hodnoty ve směru osy X narůstají směrem doprava, hodnoty ve směru osy Y narůstají směrem dolů. Při zápisu souřadnice se uvádí nejprve osa x, potom osa y. Je přitom možné zadávat příkazy pro kreslení mimo plochu grafického okna, zobrazí se však pouze ta část, která je obsažena maximálními souřadnicemi grafického okna.

Aktuální souřadnice se zobrazují ve stavovém řádku systému vždy, když máte nastavenou myš nad grafickým oknem. Souřadnice je vhodné si vyzkoušet také na povelu Point, který slouží pro zobrazení bodu na zadané souřadnici. Povel nakreslí na zadaných souřadnicích bod o zadané velikosti. Bod se nakreslí aktuální barvou pera, kterou lze nastavit procedurou ImagePenColor. Vyzkoušejte si několik příkladů pro seznámení se se způsobem označování souřadnic grafické polohy:

```
Point( 0, 0, 2); { levý horní roh }
Point( 0, 100, 2); { levý dolní roh }
Point(100, 0, 2); { pravý horní roh }
Point(100, 100, 2); { pravý dolní roh }
Point( 50, 50, 10); { uprostřed, větší bod }
```

Souřadnice grafického okna se použijí i pro kreslení geometrických tvarů pomocí příkazů. Možné je kreslit čáru příkazem Line, obdélník příkazem Rectangle, kružnici nebo elipsu příkazem Ellipse a obdélník se zaoblenými rohy příkazem RoundRect. Příkazem Triangle je možné nakreslit trojúhelník, což není pomocí myši možné. Pomocí příkazů je možné také kreslit příkazem Arc část kružnice a příkazem Pie kruhovou výseň. Vyzkoušejte si kreslení základních geometrických tvarů:

```
Line(20, 20, 50, 100); { nakreslí čáru }  
Rectangle(10, 10, 100, 100); { nakreslí čtverec }  
Ellipse(30, 30, 120, 120); { nakreslí kružnici }  
Ellipse(30, 30, 120, 60); { nakreslí elipsu }  
Triangle(10,100,55,10,100,100); { nakreslí trojúhelník }
```

```
Arc(0,0,100,100, 50,0,0,50); { levý horní čtvrtkruh }  
Pie(0,0,100,100, 50,100,100,50) { pravý dolní čtvrtkruh }
```

Následující lekce

Lekce 1.08 - Výstup do grafického okna

V předchozí lekci jsme se naučili kreslit do grafického výstupního okna z příkazového okna a nastavovat parametry řádků a ploch. Nyní se zaměříme na nastavení grafické plochy a převodě části grafického okna ze zásobníku Windows a zpět. Seznámíme se také s možností zápisu textu do grafického okna.

Až dosud jsme měli velikost grafické plochy nastavenou vždy podle toho, jak velké bylo grafické okno v okamžiku aktivace grafické plochy. Další zmínou velikosti grafického výstupního okna se již velikost grafické plochy neměnila. Pokud proto potřebujeme nastavit velikost grafické plochy na požadované rozměry, můžeme tak učinit příkazem `ImageInit`. Příkaz inicializuje grafické okno a nastaví jeho velikost na rozměry zadané parametry. Nastaví současně bílou barvu plochy, styl štítků pro plně vybarvené plochy, černou barvu pera, sílu čáry na jeden bod. Inicializací se provede výmaz původního obsahu grafického okna. Vyzkoušejte si například následující hodnoty:

```
ImageInit( 50, 200);  
ImageInit(120, 60);
```

Pokud potřebujete znát parametry již inicializované grafické plochy, můžete použít dotazy `GetMaxX` a `GetMaxY`, který vrácí velikost grafické plochy zadané strany. Pokud budete například potřebovat vykreslit bod uprostřed grafické plochy bez ohledu na velikost aktuálně inicializované plochy, můžete tak učinit příkazem:

```
Point(GetMaxX/2, GetMaxY/2, 5);
```

V některých případech by bylo vhodné kreslit čáry do grafické plochy zadáváním v absolutních přírůstkových hodnotách místo přesné definice souřadnic. Proto je v grafickém okně definován tzv. grafický ukazatel, který zaznamenává pozici vykreslení posledního bodu. Poloha grafického ukazatele se při použití některých příkazů automaticky mění. Je možné ji nastavit i z programu příkazem `MoveTo`. Čáry je možné potom zadat příkazem `LineTo` definicí konečného bodu. Využitím uvedených příkazů je možné například nakreslit libovolný mnohostranný mnohoúhelník. Vyzkoušejte si následující příkazy, které by měly v grafické ploše nakreslit čtyřúhelník:

```
MoveTo( 10, 10); { přesun na počáteční bod }  
LineTo(100, 10); { horní hrana }  
LineTo(100, 100); { pravá hrana }  
LineTo(100, 10); { spodní hrana }  
LineTo( 10, 10); { levá strana }
```

Až dosud jsme důsledně dodržovali, že je možné textové informace vypisovat zásadně do textového výstupního okna a grafické informace do grafického výstupního okna. Je však možné provádět výstup textových informací i do grafického okna. Nelze přitom použít příkaz `WriteLn`, se kterým jsme se již dříve seznámili. Zobrazovat lze pouze textové informace. Číslo je nutné předem převést na řetězec. K výstupu textových informací se používá příkaz `TextOut`. Jako parametry příkazu se udávají souřadnice pro zobrazení textu v grafickém okně a řetězec, který se má zobrazit. Příkaz si můžete vyzkoušet například následujícími příklady:

```
TextOut(10, 10, "OZOGAN");  
TextOut(10, 30, "KLONDAIK");
```

Text se při použití příkazu `TextOut` vypisuje předdefinovaným fontem. Jeho změna se provede buď nastavením příkazem `ImageFontColor` (barva fontu), `ImageFontName` (jméno fontu), `ImageFontSize` (velikost fontu) a `ImageFontStyle` (styl fontu).

Nastavení barvy fontu se zadává u povelu `ImageFontColor` jako parametr buď číslo barvy, nebo jméno konstanty udávající barvu. Tabulka hodnot barev je stejná jako pro nastavení barvy čar a ploch. U povelu `ImageFontName`, který nastavuje jméno fontu se uvádí přímo jméno fontu. Pokud není zadáno jméno v systému Windows dostupné, použije se font s podobným jménem. Velikost fontu se uvádí v povelu `ImageFontSize` přímo požadovanou hodnotou. Příklad možných nastavení si prozkoušejte včetně výpisu textu po každé změně parametru fontu:

```
ImageFontColor(clRed); { nastaví červenou barvu fontu }
ImageFontColor($00FFFF); { nastaví žlutou barvu fontu }
ImageFontName("Arial CE"); { nastaví font Arial CE }
ImageFontSize(12); { nastaví velikost fontu 12 bodů }
```

Pro nastavení stylu fontu se v povelu `ImageFontStyle` používá jako parametr předdefinované hodnoty jednotlivých stylů. Uvádí se součet hodnot požadovaného výsledného stylu:

```
-----
hodnota  konstanta  popis stylu
-----
0 fsNormal  normální písmo
1 fsBold    tučné písmo
2 fsItalic  nakloněné písmo
4 fsUnderline  podtržené písmo
8 fsStrikeOut  přeškrtnuté písmo
-----
```

Pro nastavení nakloněného podtrženého písma můžete zadat jednu z následujících možností:

```
ImageFontStyle(2 + 4); { součet hodnot }
ImageFontStyle(6); { součet 2 + 4 }
ImageFontStyle(fsItalic + fsUnderline); { uvedení konstanty }
```

Pokud budete potřebovat převést nakreslený obrázek přes schránku Windows (clipboard) do jiné aplikace, můžete tak provést pomocí povelu `ImageToClip`, kdy uvedete jako parametry souřadnice ohraničené plochy pro převod. Pokud budete chtít například přesunout do schránky obsah celého grafického okna, zadejte příkaz:

```
ImageToClip(0,0,GetMaxX,GetMaxY);
```

Načtení obsahu schránky se provádí повеlem `ImageFromClip`, kdy se jako parametry uvedou souřadnice ohraničené plochy, kam se má obsah schránky převést. Pokud budete chtít převést obsah schránky Windows na celou plochu grafického okna, zadejte příkaz:

```
ImageFromClip(0,0,GetMaxX,GetMaxY);
```