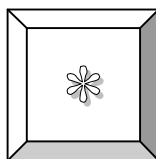
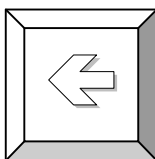


# GRAFICKÉ RUTINY pro PASCAL



**Autor: RNDr. Rudolf Kryl**  
**Autor programu: Stanislav Brabec**  
**Recenzenti: RNDr. Miloslav Feil, CSc.**

**(C) Komenium Praha 1989**

## **GRAFICKÉ RUTINY PRO PASCAL**



Uživatelský manuál a příklady použití

autor programu : Stanislav Brabec

autor příručky : RNDr. Rudolf Kryl



## A. Uživatelský manuál

=====

Tento manuál popisuje grafické prostředky, které jsou k dispozici uživatelům programujícím v Pascalu na počítači IQ 151 pod operačním systémem AMOS.

Implementace Pascalu pod OS AMOS umožňuje (přesto, že překladač i systém podpůrných podprogramů jsou jednou pro vždy pevně vypáleny v pamětech typu EPROM) rozšiřovat dodatečně nahráním programových modulů do paměti RWM repertoár standardních prostředků jazyka (tj. předdekla- rovaných procedur, funkcí a konstant).

Manuál popisuje první takové rozšíření Pascalu o prostředky pro práci s grafikou. Dále budeme Pascal rozšířený o tyto standardní prostředky označovat pro jednoduchost jako G-Pascal.

Ke studiu manuálu je potřebná znalost alespoň základů jazyka Pascal a základních informací o OS AMOS.

### 1. Charakteristika G-Pascalu

-----

G-PASCAL je rozšířením jazyka Pascal o grafické procedury a funkce. Tyto prostředky jsou jednotné jak pro kreslení na obrazovce pomocí modulu GRAFIK, tak pro kreslení na Minigrafu Aritma. Uživatel si tedy může po odladění obrázku na obrazovce vykreslit bez podstatné změny v programu stejný obrázek na Minigrafu. Dalšími výhodami grafických prostředků G-Pascalu oproti grafickým prostředkům Basicu G je jejich podstatně větší výrazová síla, umocněná ještě možností využívat prostředků Pascalu samého, a v ne- poslední řadě větší rychlost provádění programů.

Chceme-li pracovat v G-Pascalu plnohodnotně, musíme mít v počítači zasunuty tyto moduly:

PASCAL a PASCAL1

VIDEO (32 nebo 64)

podle druhu práce i moduly GRAFIK nebo MINIGRAF.

Pro překlad programů v G-Pascalu však není nutná přítomnost modulů GRAFIK ani MiniGRAF, ty musí být v počítači až při spouštění těchto programů. Naopak při spouštění již dříve přeložených programů nemusí být v počítači přítomen modul PASCAL1 obsahující překladač.

Programy v G-Pascalu nemohou pracovat, stejně jako jiné součásti OS AMOS se souřadnicovým zapisovačem XY4131 z n.p. laboratorní přístroje Praha. Důvodem je to, že programová obsluha tohoto zapisovače přepisuje oblast paměti RWM, ve které má OS AMOS své systémové proměnné.

## 2. Zavedení G-Pascalu do počítače

-----

Systémový program instalující rozšíření G-Pascalu se dodává jako object-kód na magnetofonové kazetě. Je na ní uložen v souboru :MG:GRAPH.OBJ. Je to běžný soubor typu object-kód OS AMOS, můžeme jej tedy bez problému obvyklým způsobem kopírovat na libovolné zařízení, nedoporučujeme však měnit při tom jeho jméno (protože tvar jména tohoto souboru je při jeho zavádění do paměti počítače využíván).

Rozšíření Pascalu o grafické prostředky provedeme tak, tento program spustíme příkazem OS AMOS

```
:MG:GRAPH .
```

Tím se do paměti RWM zavede z magnetofonu systémový program obsahující tato rozšíření. Správné zavedení systémového programu poznáme podle toho, že se na obrazovce zobrazí zaváděcí text

```
IQ 151 G -Pascal .
```

Program po zavedení zabírá v paměti cca 19 sektorů a je umístěn v adresovém prostoru pod tabulkou souborů. Při svém zavedení posune tabulku samu, i všechny soubory, které jsou právě v paměti, o příslušný počet sektorů směrem k vyšším adresám (o tom, že se to skutečně stalo, se můžete přesvědčit příkazem CAT).

Po úspěšném nahrání těchto rozšíření můžeme překládat a spouštět programy v G-Pascalu. To znamená, že všechna rozšíření jazyka, popsaná dále v tomto manuálu, se chovají jako standardní funkce a procedury a nemusíme je tedy ve svých programech deklarovat.

Je však třeba dát pozor na to, abychom nedeklarovali vlastní identifikátory shodné s identifikátory nových grafických prostředků, protože bychom je tím (v souladu s pravidly o lokalitě identifikátorů v Pascalu) zastínili.

V případě, že máme při překladu programu zapnut přepínač tisku varování o použití nestandardních rysů implementace (přepínač w), vydává překladač varování při každém použití nových rysů G-Pascalu.

Překládáme-li program používající rozšíření G-Pascalu v situaci, kdy není systémový program realizující rozšíření zaveden v paměti, ohlásí překladač chybu (obvykle to bude chyba 100 - nedeklarovaný identifikátor). V takovém případě jej musíme dříve popsaným způsobem zavést.

Systémový program rozšiřující Pascal o grafické prostředky může být do paměti zaveden kdykoli během práce se systémem AMOS. Může být v paměti současně s dalšími rozšířeními OS, které se do paměti RWM zavádějí (již bylo realizováno rozšíření OS o prostředky práce se soubory s přímým přístupem, připravuje se systém okének, instalace Minigrafu jako uživatelského zařízení, atd.). Uživatel musí však počítat s tím, že se paměťové nároky všech rozšíření v daném okamžiku právě zavedených do paměti sčítají.

### 3. Grafické pixely

-----

Kresba na obrazovce se skládá z elementárních grafických bodů tzv. pixelů. Pod názvem pixel tedy rozumíme nejmenší zobrazitelný útvar na obrazovce.

Modul GRAFIK jich zobrazuje na obrazovku 512 ve vodorovném směru (osa x) a 256 ve směru svislém (osa y). Pixely jsou obdélníkové (vyšší než širší). Každému pixelu odpovídá v grafické paměti RWM (obsažené v modulu GRAFIK) jeden bit, ve kterém je uložena informace o tom, zda má bod při zobrazení svítit nebo ne. Kreslení se provádí vlastně tím, že měníme obsah této grafické paměti.

Obsah grafické paměti se v daném okamžiku může a nemusí zobrazovat na obrazovce. Její zobrazování můžeme zapínat a vypínat jak příkazy z programu, tak i za běhu programů pomocí povelů OS AMOS.

Současný stisk tlačítek

SH+FB+BR      zapíná zobrazování obsahu grafické paměti,

FB+BR          vypíná zobrazování obsahu grafické paměti.

Abychom dosáhli jednotného ovládání Minigrafu a grafiky na obrazovce, představujeme si také kresbu na Minigrafu jako složenou z pixelů.

MINIGRAF má v jednom směru (osa y - směr posuvu písátka) 1500 pixelů, v druhém směru (osa x - směr posuvu papíru) není pro počet pixelů žádné omezení technického charakteru (při inicializaci se nastavuje 2100 pixelů).

Při práci s Minigrafem můžeme do pixelů pochopitelně pouze zapisovat (a ne testovat jejich "obsah"). Pixely Minigrafu jsou čtvercové.



Pozor

Je nutné si uvědomit, že v G-Pascalu jsou při kreslení na Minigrafu oproti BASICu (i standardních procedur Pascalu pro ovládání Minigrafu z Pascalu, které jsou vypáleny v pamětech Eprom a začínají všechny písmenem Q - viz. odstavec 7) navzájem prohozeny osy x a y. Bylo to uděláno proto, abychom při přenosu obrázku z obrazovky, která je širší než delší, na papír efektivně využili jeho velikost.

Programátor pracuje přímo s pixely jen ve zcela výjimečných situacích. Obvykle zadává povely pro kreslení v jemu lépe vyhovujících uživatelských souřadnicích.

#### 4. Skutečné a uživatelské souřadnice

-----

V G-Pascalu může pracovat v zásadě ve dvou různých typech souřadnic:

skutečné souřadnice

Pomocí těchto souřadnic zadáváme skutečné místo na zobrazovacím médiu. Každému pixelu v nich odpovídá dvojice celočíselných souřadnic.

Souřadná soustava májící v levém dolním rohu bod o souřadnicích (0,0). S těmito souřadnicemi se programátor v G-PASCALU setká pouze u funkcí XPOS a YPOS a procedury WINDOW.

uživatelské souřadnice

Programátor má možnost definovat si příkazem SCALE vlastní souřadnou soustavu tak, aby co nejvíce vyhovovala problému, který ve svém programu řeší. V tomto příkaze uživatel vlastně předepisuje, jaká část roviny se bude zobrazovat v okénku definovaném příkazem WINDOW. Na jednotlivé body roviny se pak odkazuje jejich souřadnicemi v této uživatelské souřadné soustavě.

O přepočítání uživatelských souřadnic na skutečné souřadnice pixelů (se kterými pracuje modul Grafik a Minigraf) se automaticky postará systém podpůrných programů pro běh pascalských programů.

## 5. Popis standardních konstant, procedur a funkcí G-Pascalu

---

### 5.1. Standardní konstanty

---

G-Pascal je rozšířen o dvě předdeklarované konstanty.

PI = 3.14159

Udává přibližnou hodnotu Ludolfova čísla  $\pi$ . Je výhodné ji použít např. při zadávání velikostí úhlů.

ORTPRO = .....

Udává poměr šířky pixelu GRAFIKU k výšce. Je větší než 0.5 a menší než 1 (její hodnotu můžeme měnit pomocí programu KOREKCE viz kapitola 8.).

### 5.2. Podprogramy pro inicializaci a přepínání režimů

---

Funkce grafických prostředků G-Pascalu je ovlivňována aktuálním nastavením dvou přepínačů.

První z nich nazýváme módem. Určuje zda se má kreslit na obrazovce nebo na Minigrafu.

Druhý určuje, zda používáme tzv. ortonormálního režimu práce nebo ne. Doporučujeme uživatelům pracovat v ortonormálním režimu, protože v něm jde jednoduchým způsobem nastavit měřítko tak, aby se vzdálenosti ve směru obou os měřily ve stejných jednotkách (viz. popis procedur WINDOW a SCALE v 5.3.). Není-li tato podmínka splněna, dochází vlivem různých měřítek na osách x a y ke zkreslení obrázků.

V popisech parametrů procedur ovlivňujících nastavení těchto dvou přepínačů používáme v tomto odstavci identifikátoru SWITCH ve významu označení typu

```
SWITCH = 0..1      .
```

Tento identifikátor však není součástí G-Pascalu.

Obě zařízení (GRAFIK a MINIGRAF) musíme před první prací s nimi inicializovat pomocí procedury INIT.

G-Pascal může pracovat ve dvou modech

```
0 .. GRAFIK      a      1 .. MINIGRAF      .
```

Po provedení procedury INIT je mód nastaven na právě inicializované zařízení. V programu pak lze mód měnit pomocí procedury MODE.

#### PROCEDURE INIT (MOD:SWITCH);

Procedura INIT inicializuje zařízení udané parametrem a zároveň nastavuje mód práce s tímto zařízením. Před započítím jakékoliv práce se zařízením musí být příslušné zařízení inicializováno !

Procedura INIT provádí tyto akce (jejich význam je popsán dále) :

ORTON(1),

WINDOW nastaví maximální okénko na daném zařízením,

SCALE(-1,1,-1,1), SOLID(0,0), ROT(0),

MOVE(0,0),

u GRAFIKU navíc SHOW, COLOR (0) a CLS,

u MINIGRAFU navíc PNU .

PROCEDURE MODE (MOD:SWITCH)

Přepíná podle specifikované hodnoty parametru MOD mód práce

0 = GRAFIK , 1 = MINIGRAF .

Po provedení procedury MODE doporučujeme vždy znovu nastavit parametry kresby pomocí procedur WINDOW, SCALE MOVE, SOLID.

PROCEDURE ORTON (ORTONORM:SWITCH)

Procedura nastavuje (je-li hodnota parametru 1) nebo ruší (je-li hodnota 0) ortonormální režim práce.

### 5.3. Definování okénka a uživatelských souřadnic

-----

Grafické prostředky G-Pascalu nekreslí na celou kreslicí plochu, ale jen do zvoleného okénka. Toto řešení umožňuje velmi snadno jak zmenšovat resp. zvětšovat nakreslený obrázek, tak skládat obrázek z několika částí. Umístění okénka provádíme pomocí procedury WINDOW.

PROCEDURE WINDOW (XLEFT,XRIGHT,YDOWN,YUP:INTEGER)

Specifikované parametry udávají ve skutečných souřadnicích polohu okénka, do kterého se bude kreslit.

Udáváme po řadě x-ovou skutečnou souřadnici levého a pravého okraje a y-vou skutečnou souřadnici dolního a horního okraje tohoto okénka. Mimo toto okénko se kresba neprovádí.

Není vhodné, nastavovat okénko větší, než může být zobrazeno na daném zařízení. Po provedení příkazu INIT se nastaví největší okénko používané pro zadané zařízení (pro Grafik 511 x 255 , pro Minigraf 2100 x 1500).

PROCEDURE SCALE (XLEFT,XRIGHT,YDOWN,YUP:REAL)

Pomocí této procedury definujeme v okénku určeném příkazem WINDOW soustavu uživatelských souřadnic. Jednotlivé parametry specifikují po řadě:

uživatelskou x-ovou souřadnici levého okraje okénka,  
uživatelskou x-ovou souřadnici pravého okraje okénka,  
uživatelskou y-ovou souřadnici dolního okraje okénka,  
uživatelskou y-ovou souřadnici horního okraje okénka.

Máme-li však nastaven ortonormální režim, mají parametry procedury poněkud jiný význam. V tomto případě neudávají uživatelské souřadnice vrcholů okénka, ale vrcholů maximálního (s okénkem soustředného) čtverce, který jde do okénka vepsat. Jde při tom skutečný čtverec na obrazovce resp. na Minigrafu, nikoli o obdélník, který má v uživatelských nebo skutečných souřadnicích stejnou výšku a šířku.

Důvodem pro tuto změnu významu parametrů byla snaha dát uživateli snadnou možnost jak nezávisle na nastavení okénka jednoduše (tj. bez zbytečných propočtů) definovat uživatelské souřadnice tak, aby se na obou osách užívalo stejného měřítka. Abychom toho dosáhli, stačí v ortonormálním režimu definovat měřítko SCALE(a,a+c,b,b+c) s libovolnými hodnotami a,b,c.

Protože se kreslení provádí pouze v aktuálním okénku (jehož poloha se zadává ve skutečných souřadnicích), je pro programátora často důležité mít možnost jednoduše otestovat, zda bod s danými uživatelskými souřadnicemi v okénku leží nebo ne. K tomu mu slouží funkce INSIDE.

FUNCTION INSIDE (X,Y:REAL):BOOLEAN

Funkce otestuje zda bod o uživatelských souřadnicích (X,Y) leží v aktuálním okénku (a zda by se tedy např. při provedení příkazu PLOT (X,Y) zobrazil). Leží-li bod v okénku, vrátí hodnotu true, jinak hodnotu false.

Uživatelskou souřadnou soustavu můžeme posouvat a otáčet pomocí procedur SOLID a ROT. Parametry oběma těmito procedurám zadáváme v neposunutých a nenatočených uživatelských souřadnicích.

PROCEDURE SOLID (X,Y:REAL);

Parametry určují uživatelské souřadnice nového počátku souřadného systému (okolo kterého pak můžeme např. souřadnou soustavu otáčet příkazem ROT). Souřadnice tohoto nového středu souřadné soustavy se udávají v původní neposunuté a neotočené soustavě souřadné.

PROCEDURE ROT (FI:REAL)

Parametr FI udává úhel otočení souřadné soustavy. Úhel se při všech voláních procedury stále udává vzhledem k původní nerotované souřadné soustavě. Úhel otočení se měří v radiánech, kladná orientace je proti směru hodinových ručiček.

Je třeba mít na zřeteli, že nejsou-li měřítka na obou osách stejná, měří se úhel vlastně v obloukové míře elipsy.

5.4. Procedury specifické pro modul GRAFIK

-----  
(kreslení na obrazovce)  
-----

Procedury a funkce tohoto odstavce mají smysl pouze při nastavení módu 0 tj. práce s GRAFIKem, pro MINIGRAF jsou to prázdné akce.

#### 5.4.1. Zapínání a vypínání zobrazování obsahu grafické paměti

---

##### PROCEDURE SHOW

Procedura zapne zobrazování obsahu grafické paměti v modulu GRAFIK (stejnou funkci má povel OS AMOS SH+FB+BR).

##### PROCEDURE HIDE

Procedura vypne zobrazování obsahu grafické paměti v modulu GRAFIK (stejnou funkci má povel OS AMOS FB+BR).

#### 5.4.2. Kreslení na obrazovce

---

V tomto odstavci jsou uvedeny prostředky specifické pro kreslení na obrazovce. Kromě nich lze samozřejmě pro tento účel používat základních grafických prostředků, které popisuje odstavec 5.7.

##### PROCEDURE CLS

Procedura smaže (tj. vynuluje) celý obsah grafické paměti v modulu GRAFIK.

Na obrazovce je možné kreslit v G-Pascalu několika různými "barvami". Barva kresby se volí procedurou COLOR.

##### PROCEDURE COLOR (COL: Integer)

Parametr COL smí nabývat pouze hodnot 0..3. Hodnota parametru udává způsob kresby (její barvu):

0 .. kreslí se bíle

1 .. kreslí se černě

2,3 .. při kreslení se invertuje barva, která na obrazovce v daném bodě právě je.

Pro proceduru PAINT a funkci POINT jsou barvy 2 a 3 ekvivalentní barvám o hodnotou o 2 menší.

#### PROCEDURE CLEAR

Procedura "vymaže" aktuální okénko tj., vyplní oblast uvnitř okénka barvou obrácenou k nastavena barvě kreslení, případně vnitřek okénka invertuje.

#### FUNCTION POINT (X,Y:REAL):BOOLEAN

Funkce testuje zda bod, který je uvnitř aktuálního okénka, má stejnou barvu jako je barva kresby (nastavená posledním příkazem COLOR) . Má-li stejnou barvu, pak funkce vrátí hodnotu true, má-li jinou, pak vrátí hodnotu false.

#### PROCEDURE PAINT (X,Y:REAL)

Vyplní oblast v níž leží bod (X,Y) a která je ohraničená libovolnou křivkou nakreslenou barvou kresby (tu určil poslední příkaz COLOR).

Pro bod (X,Y) při tom musí platit  $INSIDE(X,Y)=TRUE$  a  $POINT(X,Y)=FALSE$ , tj. tento bod musí ležet uvnitř aktuálního okénka a musí barvu obrácenou k té, která byla naposledy nastavena příkazem COLOR.

### 5.5. Grafický kříž a pisátko

---

Základní procedury G-Pascalu pro kreslení, se kterými se seznámíme v odstavci 5.7. pracují, jak je to obvyklé, s pojmem tzv. grafického kříže.

Grafický kříž je myšlený bod, jehož zavedení nám výrazně zjednoduší psaní příkazů pro kreslení. velmi často totiž kreslíme složitější obrazce skládající se z více na sebe navazující čar. Bylo by tedy velmi nepohodlné, kdybychom museli v příkazu pro kreslení čáry vždy zadávat jak výchozí,



tak i koncový bod čáry. Proto zadáváme u většiny příkazů jen koncový bod a za výchozí bod se bere poloha grafického kříže.

Máme k dispozici proceduru MOVE, která nedělá nic jiného, než že nastaví grafický kříž do požadovaného bodu. Polohu grafického kříže však ovlivňují i další procedury pro kreslení (většinou Jej nastaví do bodu, ve kterém skončily s kresbou).

U Minigrafu většinou odpovídá poloha grafického kříže poloze pisátka.

Polohu grafického kříže ve skutečných souřadnicích jde zjisti pomocí funkcí XPOS a YPOS.

FUNCTION XPOS : INTEGER;

FUNCTION YPOS : INTEGER

Funkce zjišťují polohu grafického kříže ve skutečných souřadnicích.

## 5.6. Procedura specifická pro Minigraf

-----

### PROCEDURE PNU

Procedura zvedne pisátka. Jako vedlejší efekt při tom přesune grafický kříž do aktuální polohy pisátka.

Je-li nastaven mód kreslení na obrazovce (mód 0), je zavolání této procedury prázdná akce.

## 5.7. Základní prostředky pro kreslení

-----

V tomto odstavci jsou soustředěny základní procedury realizující kreslení bodů, úseček a oblouků, které jsou v G-Pascalu společné pro obrazovku i Minigraf.

Je však nutné mít na zřeteli, že pro kreslení na obrazovce můžeme nastavit "barvu" kresby procedurou COLOR. Všechny procedury kreslí tedy na obrazovku v barvě zvolené při posledním volání, této procedury.

#### PROCEDURE FRAME

Nakreslí rámeček okolo aktuálního okénka.

#### PROCEDURE MOVE (X,Y:REAL)

Přesune grafický kříž do bodu o zadaných uživatelských souřadnicích.

U Minigrafu se při provedení tohoto příkazu skutečně pisátka vůbec nepohne, systém si pouze poznamená novou polohu grafického kříže. Pisátka se posune do správné polohy až při prvním příkazu ke kreslení. Tímto způsobem nedochází ani při nešikovném programování ke zbytečným pohybům pisátka.

#### PROCEDURE PLOT (X,Y:Real)

Nakreslí bod v místě o zadaných uživatelských souřadnicích a zvedne pisátka. Zároveň se do tohoto bodu přesune grafický kříž.

#### PROCEDURE DRAW (X,Y:REAL)

Nakreslí čáru ze stávající polohy grafického kříže do bodu o zadaných uživatelských souřadnicích (X,Y) a pak do něj přesune grafický kříž.

Kreslí-li se bod (X,Y) skutečně (tj. leží-li uvnitř aktuálního okénka), pak pisátka Minigrafu zůstává spuštěno.

Na obrazovce je čára kreslena tak, že bod, do kterého čáru táhneme, už kreslen není. Má to význam při kreslení invertování (barvy 2 a 3), při kterém by vadilo, kdyby se bod, ve kterém jedna čára navazuje na druhou, kreslil dvakrát.

PROCEDURE CIRCLE (XDIM,YDIM:REAL)

Nakreslí elipsu se středem v grafickém kříži, s osami ve směrech uživatelské souřadné soustavy a délkami poloos, zadaných hodnotami parametrů (zadávané je v uživatelských souřadnicích). Procedura nemění polohu grafického kříže.

PROCEDURE ARC (XDIM,YDIM,FISTART,FISTOP:REAL)

Nakreslí oblouk elipsy, se středem v grafickém kříži, osami ve směrech uživatelské souřadné soustavy a délkami poloos, zadaných hodnotami prvních dvou parametrů (zadávané je v uživatelských souřadnicích). Počáteční a koncový úhel oblouku je odčítán od uživatelské osy x proti směru hodinových ručiček v obloukové míře měřené na oblouku elipsy CIRCLE (0,0) - v případě ortonormální soustavy souřadnic jde tedy o běžnou obloukovou míru. Procedura nemění polohu grafického kříže.

## 6. Běhové chyby G-Pascalu

Dojde-li při práci procedur napsaných v G-Pascalu k chybě, ohlásí se obvyklým způsobem hlášení o běhové chybě pascalského programu. Jen k chybám uvedeným v chybníku přibývají ještě další s těmito kódy:

- A1 .. špatný parametr grafického příkazu
- A2 .. proceduře PAINT schází paměťový prostor pro zásobník (program je příliš dlouhý nebo kreslený obrazec je příliš složitý)
- A3 .. k vykonání příkazu je nutno mít připojený modul MINIGRAF nebo GRAFIK.

Dále mohou grafické procedury hlásit některou z chyb aritmetiky reálných čísel. Tato chyba vzniká při nevhodně nastavených hodnotách parametrů procedur SCALE a WINDOW nebo při "kreslení" bodů velmi vzdálených od aktuálního okénka.

## 7. Spolupráce se standardními procedurami pro ovládání ----- Minigrafu -----

Implementace Pascalu pod OS AMOS obsahuje již ve své verzi vypálené v pamětech EPROM skupinu standardních funkcí pro ovládání Minigrafu (jejich výčet je v odstavci 7.1).

Tyto prostředky vlastně pouze realizují zavolání podprogramů umístěných v paměti EPROM přípojovacího modulu Minigrafu. Všechny pracují přímo se skutečnými souřadnicemi. Proto není jejich používání zdaleka tak pohodlné jako práce s procedurami G-Pascalu.

V principu však můžeme tyto standardní procedury používat i kombinovaně s grafickými prostředky G-PASCALU, nemá to však velký praktický význam.

Jediným důvodem pro to může být skutečnost, že standardní procedura QWRITE umožňuje psaní textu na Minigraf. Analogická procedura nebyla do G-Pascalu zahrnuta proto, že při práci s ním nemusíme mít (kreslíme-li na obrazovce) v počítači zasunut přípojný modul k Minigrafu, který obsahuje tvary písmen. Chtěli bychom vás však upozornit na to, že chceme-li na Minigrafu pouze nechat vypisovat texty (bez nároků na změny tvaru písmen), pak přirozenějším a pro uživatele pohodlnějším řešením tohoto problému je použít pro to systémový program pro instalaci Minigrafu jako uživatelského výstupního zařízení. Tento program je již k dispozici.

Při kombinovaném používání procedur G-Pascalu s těmito standardními prostředky pro ovládání Minigrafu, je nutné mít na paměti tyto skutečnosti:

- 1) Chceme-li používat prostředku G-Pascalu, musíme provést inicializaci Minigrafu příkazem INIT(1 )

- 2) Při přechodu ze standardních procedur pro ovládání Minigrafu na procedury G-Pascalu je vhodné zavolat proceduru PNU .
- 3) Je třeba si dále uvědomit, že procedury G-Pascalu mají oproti standardním procedurám pro ovládání Minigrafu vzájemně prohozené souřadné osy.

### 7.1. Seznam standardních procedur pro ovládání Minigrafu

---

Názvy všech standardních procedur začínají písmenem Q, jejich funkce je stejná jako funkce odpovídajících prostředků Basicu a je podrobně popsána v manuálu Minigrafu. Proto zde uvádíme pouze tvar a význam jejich parametrů. Pascalský systém ani tyto podprogramy samy však nekontrolují, zda byly procedury zavolány s přípustnými hodnotami skutečných parametrů. To musí hlídat programátor. Procedury pracují se skutečnými souřadnicemi, hodnoty parametrů musí ležet v intervalu 0..1500 pro souřadnici x a v intervalu 0..2100 pro souřadnici y.

procedure QORG (X,Y:integer);

inicializace polohy písátka do bodu (X,Y)

procedure QMOVA. (X,Y:integer);

přesune (bez kresby) písátko do bodu (X,Y), písátko zůstane zvednuto

procedure QMOVE(DX,DY:integer);

přesune (bez kresby) písátko z dosavadní polohy (X0,Y0) do bodu (X0+DX,Y0+DY), písátko zůstane zvednuto

procedure QVECTA (X,Y:integer);

nakreslí úsečku z dosavadní polohy písátka do bodu (X,Y), písátko zůstane spuštěno

procedure QVECTR (X,Y:integer):

nakreslí úsečku z dosavadní polohy písatka (X0,Y0) do bodu (X0+DX,Y0+DY), písatko zůstane spuštěno

procedure QPNTA (X,Y:integer):

přesune (bez kresby) písatko do bodu o souřadnicích (X,Y) a nakreslí tam bod , písatko zůstane spuštěno

procedure QPNTR (X,Y:integer):

přesune (bez kresby) písatko z dosavadní polohy (X0.Y0) do bodu (X0+DX,Y0+DY) a nakreslí tam bod, písatko zůstane spuštěno

procedure QSIZE(XX,XY,YX,YY:integer):

volba tvaru, sklonu a směru tisku

procedure QNARROW:

nastavení úzké mezery mezi znaky

procedure QWIDE:

nastavení široké mezery mezi znaky

procedure QWRITE( znakový řetězec nebo char );

vypíše požadovaný text

procedure QSPEED(S:integer):

nastaví rychlost kreslení (příp. hodnoty 1 . .5)

## 8. Program KOREKCE

-----

Protože ne všechny obrazovky jsou nastaveny stejně, je spolu s programovým modulem G-Pascalu dodáván i program KOREKCE. Pomocí tohoto programu si může uživatel změnit hodnotu konstanty ORTPRO udávající poměr šířky pixelu k jeho výšce tak, aby odpovídala seřízení jeho obrazovky.

Zdrojový tvar programu KOREKCE (je napsán v G-Pascalu) a dodává se na stejné magnetofonové kazetě jako programový modul G-Pascalu. Je nahrán v souboru :MG:KOR.

Po překladu tohoto programu např. příkazem OS AMOS

PAS :MG:KOR

(v té době již musí být zaveden modul G-Pascalu v paměti) lze tento program spustit příkazem

KOR .

Po svém spuštění se program nejprve zeptá, zda-li chceme rovnou zadat hodnotu konstanty ORTPRO.

Chceme-li přímo zadat novou hodnotu, odpovíme na dotaz klávesou "H" a pak novou hodnotu zadáme.

Nechceme-li ji přímo zadat, odpovíme na dotaz stiskem libovolné jiné klávesy.

Program vykreslí několik soustředných kružnic a čeká na stisk klávesy. Pak vypíše dotaz, zda je nastavení správné.

Nejsme-li s tvarem kružnic spokojeni, vykreslí nám na obrazovku obdélník a vyzve nás, abychom změřili jeho skutečnou šířku a výšku. Protože je podstatný pouze poměr těchto dvou velikostí, nezáleží na jednotkách, ve kterých měříme. Při zadání nepřijatelných hodnot se na ně program ptá znovu.

Z těchto údajů program spočte novou hodnotu konstanty ORTPRO a s jejím použitím pro kontrolu opět vykreslí systém soustředných kružnic. Jsme-li již spokojeni, program vypíše novou hodnotu konstanty ORTPRO a skončí.

Hodnota konstanty ORTPRO se dosazuje do object-kódu uživateleova programu při jeho překladu. Je proto třeba všechny programy, které mají pracovat s nově nastavenou hodnotou konstanty ORTPRO, přeložit po korekci znovu.

Pokud potřebujeme zahrnout korekci konstanty ORTPRO do svého vlastního programu, stačí k tomu opsat do něj z programu KOREKCE proceduru EXEC (PROP:REAL). Tato procedura mění hodnotu konstanty ORTPRO na hodnotu udanou jejím parametrem.

## 9. Relativní souřadnice

-----

Do repertoáru základních prostředků G-Pascalu nebyly zařazeny procedury pro práci v tzv. relativních souřadnicích, ve kterých nezadáme souřadnice nové polohy grafického kříže, ale jeho posunutí vůči poloze předchozí. Důvodem pro to bylo, že přidávání dalších procedur by zbytečně zvětšovalo paměťové nároky systémového programu realizujícího rozšíření G-Pascalu. Při tom si tyto procedury může každý uživatel bez problémů sám naprogramovat.

Například tak, že si deklaruje globální proměnné XG a YG udávající uživatelské souřadnice grafického kříže a proceduru GRSTART, která je inicializuje. Pak můžeme deklarovat např. procedury relativního přesunu RMOVE a relativního kreslení úsečky RDRAW takto:

```
var XG,YG :real;          { souřadnice grafického kříže }
procedure GRSTART(X,Y:real) ;
  begin XG:=X; YG:=Y; MOVE(X,Y) end;
:
procedure RMOVE(DX,DY:real) ;
  begin XG:=XG+DX; YG:=YG+DY; MOVE(XG,YG) end;
:
procedure RDRAW(DX, DY : real) ;
  begin XG:=XG+DX; YG:=YG+DY; DRAW(XG,YG) end;
:
atd
```



## 10. Podrobněji o uživatelských souřadnicích

---

Programátor v G-Pascalu pracuje v uživatelských souřadnicích, grafická zařízení (modul GRAFIK a Minigraf) však dokáží pracovat jen ve skutečných souřadnicích, udávajících, o který pixel jde. Proto musí systém G-Pascalu provádět transformaci uživatelských souřadnic na souřadnice skutečné.

K definování uživatelských souřadnic slouží procedury WINDOW a SCALE.

Procedura WINDOW slouží k umístění okénka, do kterého chceme kreslit na obrazovce nebo na papíře, dostává tedy jako parametry skutečné souřadnice.

Procedura SCALE pak určuje, jakou část roviny nám má zvolené okénko představovat. Při jejím provádění se tedy spočítají koeficienty transformační funkce z uživatelských do skutečných souřadnic.

Při provádění příkazu WINDOW však k takovému přepočítávání transformačních koeficientů nedochází. Proto máme dvě možnosti, jak proceduru WINDOW použít.

Pokud při druhém volání procedury WINDOW zachováme poměr stran okénka a použijeme znovu přesně stejné volání procedury SCALE, dostaneme kresbu stejného motivu (případně zvětšeného či zmenšeného) v jiném místě.

Pokud změníme okénko volání procedury WINDOW, aniž bychom volali proceduru SCALE, pak se transformační funkce nemění a dostáváme jiný "výřez" z téhož obrázku.

K posouvání a rotaci uživatelské souřadné soustavy slouží procedury SOLID a ROT. Jejich parametry se udávají v "absolutních souřadnicích" tj. v původní nerotované a neposunuté souřadné soustavě.

Protože pixely na obrazovce nejsou čtvercové, ale jsou vyšší, než je jejich šířka, musí grafické procedury při práci v ortonormálním režimu vlastní vytvářet nové "falešné" čtvercové pixely. Tyto pixely se zobrazují tak, že do některých obrazových řádků se zobrazují dva "falešné" pixely. Nemusí tedy v ortonormálním režimu dávat např. funkce POINT po změně režimu vždy očekávané hodnoty.

Programátor se však nemusí o existenci "falešných" pixelů nemusí zajímat. Je třeba vědět jen to, že při práci v ortonormálním módu musíme mít stále nastaveno ORTON(1).

Skutečné hodnoty úhlů nakreslených prostředky G-Pascalu mohou nabývat pouze 512 hodnot.

## B. Příklady použití

V této části příručky ukážeme několik jednoduchých programů demonstrujících možnosti G-Pascalu.

### 1. Program Elipsy

Program demonstruje otáčení souřadné soustavy. Může pracovat jak na obrazovce, tak na Minigrafu. Zařízení se volí v úvodní konverzaci.

```

program ELIPSY(input,output);
var I,J: integer;
{program nakreslí postupně osm elips
 v otočených souřadných soustavách}
begin
{volba zařízení}
WRITE('ZADEJTE MOD '); READ(J);
INIT(J);           { inicializace }
if J=0 then WRITE(CHR(31)); {smazání obrazovky}
for I:=0 to 7 do
  begin
    ROT(PI/8*I);           {otočení soustavy}
    CIRCLE(0.5,0.25)       {elipsa}
  end;
FRAME                   {rámeček}
end.

```

## 2. Demonstrace invertování na obrazovce.

---

Program pracuje pouze s obrazovkou a využívá možnost nastavit barvu kresby 2, tj. invertování. Demonstruje rychlost grafických prostředků G-Pascalu při kreslení na obrazovku a funkci procedur WINDOW a CLEAR.

```

program Demo(input,output);
{demonstruje rychlost práce grafiky na obrazovce,
 používá příkazu WINDOW a kreslení invertováním}
var I:integer;
begin
  INIT(0);           {inicializace}
  WRITE(CHR(31));   {smazání obrazovky}
  COLOR(2);         {invertování}
  for I:=1 to 127 do
    begin
      WINDOW(0,4*I,0,2*I);  {nastavení okénka}
      CLEAR                 {smazání okénka }
    end;
  for I:=1 to 127 do
    begin
      WINDOW(511-4*I,511,255-2*I,255);
      CLEAR
    end
  end.
end.

```

### 3. Opakování téhož motivu

-----

Procedura MOTIV nakreslí postupně ve čtyřech otočených souřadných soustavách vždy deset elips a ukáže funkci procedury PAINT. Kreslíme-li na obrazovce, zopakuje tentýž motiv ještě jednou menší.

```

program Elipsy2(INPUT,OUTPUT);
var J:INTEGER;
procedure MOTIV;
  var I,J:INTEGER;
  begin {of MOTIV }
    for I:=0 to 3 do
      begin
        ROT(PI*I/4);
        for J:=0 to 9 do CIRCLE(J/10,1)
      end;
    ROT(0); PAINT(-0.9,-0.9)
  end {procedurey MOTIV}
begin {hlavní program}
  WRITE('ZADEJTE MOD '); READ(J); {výběr zařízení}
  INIT(J);
  if J=0 then WRITE(CHR(31)); {mazání obrazovky}
  MOTIV;
  if J=0 then {jde-li o obrazovku}
    begin {nakresli tentýž motiv znovu}
      WINDOW(270,480,140,240);
      SCALE(-1,1,-1,1);
      MOVE(0,0); SOLID(0,0); CLEAR;
      FRAME; MOTIV
    end
    else PNU {zvedni pisátko}
end.

```

#### 4. kreslení n-cípých hvězd

-----

Program nakreslí pravidelnou hvězdu se zadaným počtem cípů a se zadanou relativní hloubkou těchto cípů (vzhledem k průměru hvězdy).

```

program Hvezda(INPUT,OUTPUT);
var C,J:INTEGER;
W:REAL;
begin
    {volba zařízení}
    WRITE('ZADEJTE MOD '); READ(J);
    {zadání požadovaného počtu cípů hvězdy}
    WRITE('ZADEJTE POCET CIPU '); READ(C);
    {zadání relativní hloubky cípů}
    WRITE('ZADEJTE HLOUBKU - REALNE CISLO MEZI 0 a 1');
    READ(W);
    INIT(J);
    if J=0 then WRITE(CHR(31));
    ROT(PI+PI*(2*C-1)/C);      {natočení souřadné soustavy}
    MOVE(0,0.9);              {pro kreslení hvězdy}
    for J:=0 to 2*C do
        begin
            DRAW(0,0.9-0.9*W*(J-2*TRUNC(J/2)));
            ROT(PI+PI*J/C);
        end;
    PAINT(0,0);              {vyplnění vnitřku hvězdy}
    PNU
end.

```

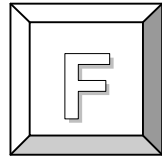
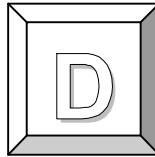
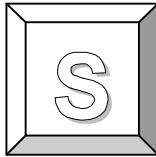
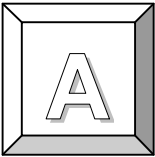
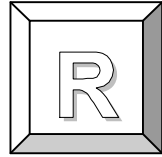
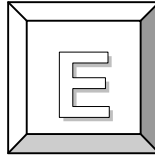
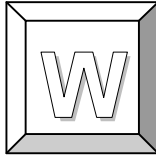
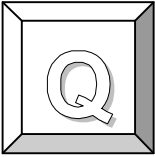
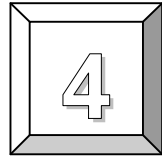
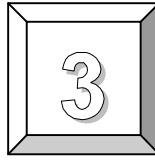
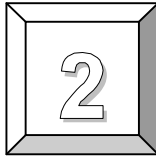
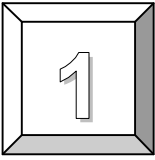
**Obsah :**

## A. Uživatelský manuál

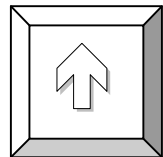
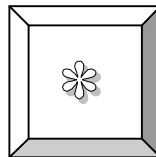
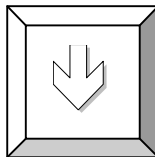
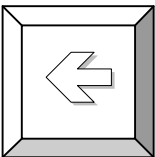
1. Charakteristika G-Pascalu .....	str. 1
2. Zavedení G-Pascalu do počítače .....	str. 2
3. Grafické pixely .....	str. 3
4. Skutečné a uživatelské souřadnice .....	str. 5
5. Popis standardních konstant, procedur a funkcí G-Pascalu .....	str. 6
5.1. Standardní konstanty .....	str. 6
5.2. Podprogramy pro inicializaci a přepínání režimů .....	str. 6
5.3. Definování okénka a uživatelských souřadnic .....	str. 8
5.4. Procedury specifické pro modul GRAFIK .....	str.10
5.4.1. Zapínání a vypínání zobrazování obsahu grafické paměti .....	str.11
5.5. Grafický kříž a pisátko .....	str.12
5.6. Procedura specifická pro Minigraf .....	str.13
5.7. Základní prostředky pro kreslení .....	str.13
6. Běhové chyby G-Pascalu .....	str.15
7. Spolupráce se standardními procedurami pro ovládání Minigrafu ....	str.16
7.1. Seznam standardních procedur pro ovládání Minigrafu .....	str.17
8. Program KOREKCE .....	str.18
9. Relativní souřadnice .....	str.19
10. Podrobněji o uživatelských souřadnicích .....	str.21

## B. Příklady použití ..... str.23

1. Program Elipsy .....	str.23
2. Demonstrace invertování na obrazovce .....	str.24
3. Opakování téhož motivu .....	str.25
4. Kreslení n-cípých hvězd .....	str.26



Grafické rutiny pro Pascal  
Autor: RNDr. Rudolf Kryl. Autor programu: Stanislav Brabec.  
Určeno pro všechny uživatele počítače IQ 151 a modulu Pascal.  
Schváleno jako příručka k programu pro všechny uživatele počítače  
IQ 151 komisí schvalování počítačových programů dne 13.9.1989.  
Vydalo Komenium Praha pod ČKL 0097/15 v roce 1989. 1. vydání.  
Odpovědný redaktor: Ing. Milan Klečka.  
Technický redaktor: M. Maňák.  
Náklad: 720 výtisků.  
K-262/89





Digitalizace, OCR a převod do elektronické podoby  
© 2008 Lukáš MACURA