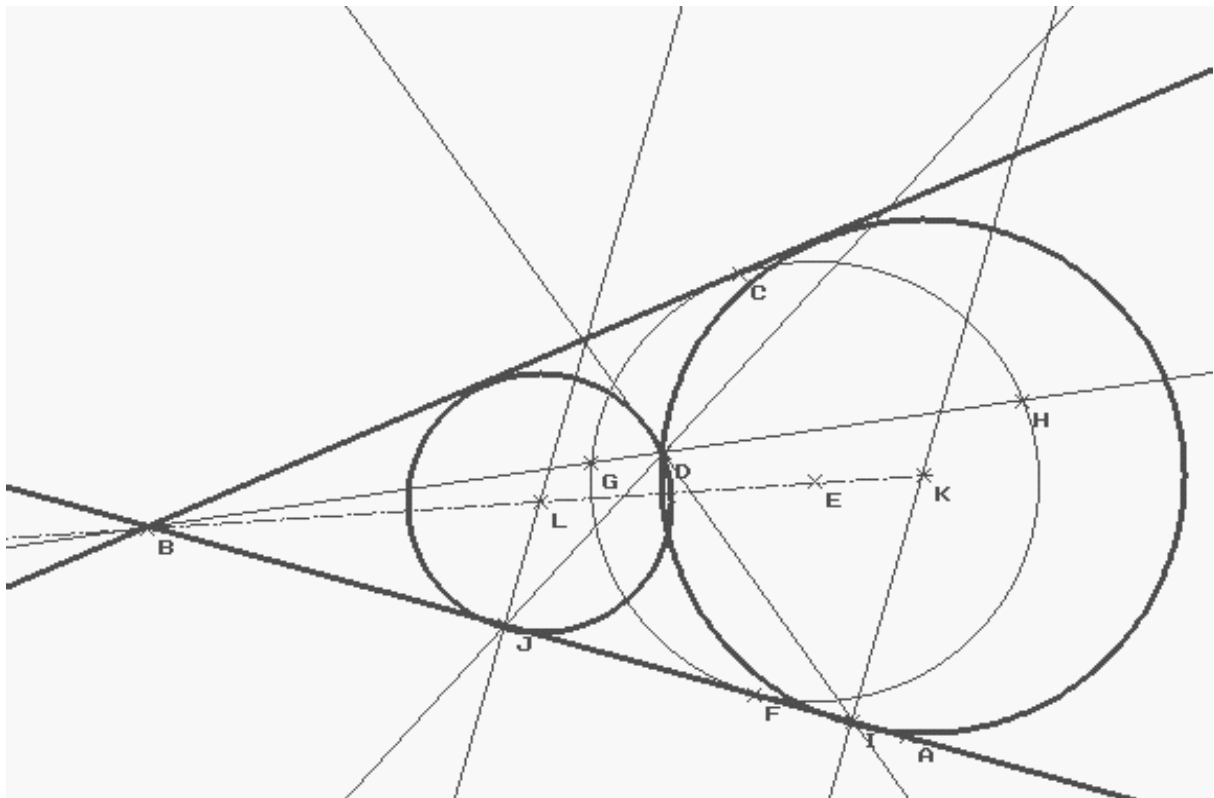


GEOM

Líbor Švéda



Popis a ovládání programu

Na úvod bych Vás chtěl všechny požádat o prominutí zdržení distribuce programu GEOM. Opravdu jsme s Liborem nečekali takový zájem z Vaší strany. Váš zájem nám udělal velikou radost, ale současně nás velmi zavázal. Přiznáváme, že dokumentace k programu a souhrn demonstračních úloh nebyly ve stavu, ve kterém bychom je mohli dodat vážným zájemcům, nehledě na to, že tato stránka věci může dost značně ovlivnit to, zda program u Vás „zapadne“ mezi spousty nepoužívaných programů, nebo ve Vás probudí zájem, a nějaký čas jej budete opravdu využívat ve výuce, což by nás oba nesmírně potěšilo.

Programů jakým je GEOM je podle mého názoru velice málo. Jistě existuje spousta programů, které umějí vykreslovat různé geometrické útvary, určovat jejich průsečíky apod. GEOM je však svým způsobem ojedinělý. Vznikal postupně na základě mých dalších a dalších neskromných přání, připomínek a požadavků. Mnohokrát musel Libor dost podstatně upravit svůj program, aby se mi zavděčil. Určitě jsem ho i několikrát pěkně naštvál, i když to nikdy nedal najevo.

Naší společnou snahou bylo program postupně co nejvíce dynamizovat. Nespokojili jsme se jen se statickým výsledkem na obrazovce, chtěli jsme více. Chtěli jsme ukázat, jakým způsobem ovlivňuje zadání úlohy konečný výsledek. Postupně tak vznikl program, jehož ovládání nebude asi pro mnohé příliš příjemné, který ovšem může učitelům matematiky nejen na základní škole velice usnadnit práci při vysvětlování nové látky. Ovládání programu z příkazové řádky vyžaduje pouze dvě věci - pochopit filozofii programu a naučit se několik příkazů. Myslím si, že každý koho program svojí filozofií zaujme, se jej naučí ovládat velice rychle.

Doufáme, že ke snadnějšímu pochopení přispějí výukové soubory **VYUKA1.GM2** a **VYUKA2.GM2**, které po spuštění programu GEOM můžete načíst klávesou **L** a zadáním názvu VYUKA1 nebo VYUKA2 (**bez přípony !!!**) a předvést si syntaxi jednotlivých příkazů a jejich funkci. Dále předkládáme sadu demonstračních souborů DEMO1.GM2, DEMO2.GM2 ... atd., které si můžete natisknout (čisté ASCII soubory) a všechny vyzkoušet. Podle mého názoru je: *Nejlepší způsob, jak se naučit s GEOM pracovat, je porovnávat syntaxi příkazů v daném demonstračním souboru s tím, co se děje na obrazovce monitoru.*

V programu je jistě celá řada drobnějších či větších „much“, které však lze vychytat pouze díky Vaší pomoci při jeho testování. Použití některých písmenek při zapisování syntaxe příkazů není příliš mnemotechnické - bohužel abeceda má pouze tolik znaků kolik má. Lepší a jednodušší ovládání programu by bylo možné pouze

v případě vytvoření grafického interface či speciálního znakového preprocesoru, který by umožnil „polidštit“ příkazový jazyk, prostřednictvím kterého sdělujete své požadavky programu GEOM. To by bylo za stávající situace pravděpodobně pracnější než vývoj samotného programu. Nikdo z Vás si navíc ani nedovede představit, jaká omezení ovladatelnosti programu vyplývají právě z těch možností programu, které jsou podle mého názoru ty nejcennější a nejzajímavější (čtení příkazů a jejich provádění ze souboru, dynamické překreslování konstrukce apod.). Program lze podle Libora vylepšit úplným přepracováním této koncepčně původní verze, na základě Vašich připomínek, nápadů a kritických postřehů. Jelikož je kombinací všech příkazů velmi mnoho a nebylo je tedy všechny možné odladit, může se stát, že program někdy „spadne“. Berte to prosím s rezervou, poznamenejte si při jaké příležitosti se to stalo a napište nám. Vaše dopisy a připomínky, náměty, nápady, pochvaly i kritika pro Libora i pro mne budou jistě odměnou .

Na závěr bych chtěl, nejen jako Liborův učitel matematiky, zcela sentimentálně Liborovi poděkovat za jeho fantastický přístup a za to, že mám ten nejskvělejší pocit, jaký může učitel vůbec kdy poznat - pocit, že učitel a student si mohou velmi dobře rozumět, vzájemně se ctít a respektovat.

Díky !!!

Jaroslav Hajtmar

Připomínky a náměty zasílejte prosím na adresy :

Libor Švéda
Svobodov 8
789 01 Zábřeh
tel. 0648/22833
e-mail: hajtmar@usa.net

nebo

RNDr. Jaroslav Hajtmar
Gymnázium Zábřeh
nám. Osvobození 20
789 01 Zábřeh
tel. 0648/411138, tel./fax.0648/22380
e-mail : hajtmar@usa.net

GEOM

Libor Švéda

Většina geometrických úloh na základní škole se řeší manipulací se třemi základními typy objektů. Jedná se o body, přímký (úsečky, polopřímky) a kružnice. Za použití tužky, pravítka a kružítko jsme schopni tyto konstrukční úlohy sestrojít.

Každá konstrukční úloha je vlastně řešena posloupností základních úkonů. Např. chceme-li sestrojít Thaletovu kružnici s průměrem daným úsečkou, najdeme nejprve střed této úsečky, pak opišeme kružnici s tímto středem a poloměrem určeným středem úsečky a krajním bodem úsečky. Stejně tak při konstrukci kružnice trojúhelníku opsané si sestrojíme osy dvou stran trojúhelníka, najdeme jejich průsečík a nakonec opišeme kružnici.

Program GEOM mohl vzniknout právě proto, že každou konstrukční úlohu můžeme rozložit na základní kroky, jakými jsou: sestrojení bodu, přímký, polopřímky, úsečky a kružnice. Pokud umíme navíc nalézt průsečíky těchto elementárních útvarů, nic nám již nebrání v počítačovém zpracování dané konstrukční úlohy

Filozofie programu

Jakýkoli objekt, se kterým program GEOM pracuje, je určen množinou bodů. Např. úsečka pomocí dvou krajních bodů, kružnice pak pomocí středu a poloměru, který můžeme zadat buď číselně nebo jako úsečku, jejíž velikost si program sám spočítá a použije jako poloměr.

Komunikace s programem je vedena prostřednictvím klávesnice a příkazy jsou zadávány na příkazovou řádku. Příkazy, které jsou tvořeny skupinou znaků, jsou navíc průběžně ukládány do textového souboru. Do tohoto souboru se ukládají jen příkazy, které tvoří postup konstrukce. Neukládají se příkazy pro posun bodů, zvětšení resp. zmenšení apod. Tyto je možné používat pouze interaktivně a jsou vyvolány stisknutím „horké“ klávesy.

Protože se jednotlivé příkazy čtou nejen z klávesnice, ale i z textového souboru, je možné si předem připravit libovolným textovým editorem soubor příkazů, popisujících postup konstrukce. Program GEOM pak postupně sled těchto příkazů vykoná. Na druhé straně je možné úlohu, kterou jsme interaktivně vyřešili, uložit do textového souboru pro pozdější použití.

Jedinečnou vlastností programu je možnost přemístit kterýkoli z bodů určujících konstrukci, přičemž dojde k aktuálnímu překreslení úlohy (samozřejmě nemá smysl přesouvat bod, který je vytvořen v průběhu konstrukce).

Protože program hojně pracuje s pevným diskem počítače, je dobré mít program GEOM nainstalován na disku, který není komprimován. Stejně tak pomalý pevný disk může výrazně zpomalit některé užitečné funkce programu. Program zatím nebyl v širším měřítku testován, ani nebyly zjišťovány jeho nároky na hardware. Zkoušel jsem jej i na 286 s 1MB RAM. Program mi fungoval, potíže jsou pouze s EGA adaptérem, na němž dochází při vykreslování kružnic ke zkreslení.

Pokud by někdo přece jen potřeboval provozovat GEOM na takovéto konfiguraci, ozvěte se, problém lze vyřešit, ale zatím jsem se jím nezabýval. Konfigurace, na níž by měl GEOM solidně běžet, je podle mne 386 DX s 2-4 MB RAM a VGA adaptér. Pokud se Vám bude zdát program hodně pomalý, věřte, že při přepočítávání složitější konstrukční úlohy procesor rozhodně nezahálí.

Jestliže se Vám bude program líbit a budete ho používat, napište, co by se dalo ještě zlepšit. Pokud narazíte na jakékoliv chyby, napište také - pomůže to vylepšit snad budoucí kompletně přepracovanou verzi. Pokud se Vám GEOM líbit nebude, (díky příkazové řádce asi nebudete sami) máte ještě možnost si zakoupit něco podobného u firmy pana Pachnera.

Seznam použitelných příkazů s příklady

Všechny prováděcí příkazy se zadávají na příkazovou řádku označenou „promptem“ >.

- n** K **vytvoření nového bodu** nám slouží písmenko n. Pokud chceme vytvořit nový G bod o souřadnicích 3 a -2.5, stačí napsat **nG(3,-2.5)**. Tento způsob může být nepohodlný. Někdy se nám může hodit, pokud můžeme souřadnice bodu zadávat pomocí kurzorových kláves interaktivně. Pak stačí napsat **nG**. V tomto případě se bod G objeví na souřadnicích [0,0], a my jej můžeme přesunout na požadované místo. Pak stačí zmáčknout ENTER a bod je zadán. V případě, že bychom potřebovali, a to se může často stát, vytvořit bod G na libovolném místě na nějaké přímce, stačí napsat **nG/oAB**. Tento příkaz nám umožní vytvořit bod G na ose úsečky AB, případně na jiné přímce syntakticky zadané za znakem / (viz. určení přímky níže). Pomocí kláves „+“ nebo „-“ si bod G, pohybující se po zadané přímce, posuneme do těch míst, kde ho zhruba potřebujeme mít umístěn, a pak stiskneme ENTER. Tím máme zaručeno, že se na dané přímce bude opravdu nalézat (dokud ho nepřesunete jinam).
- d** Program GEOM dokáže pracovat s body označenými písmeny abecedy. Je jich poměrně dost. Ale v komplikovanějších příkladech se může stát, že jsme vyčerpali zásobu písmen a přesto potřebujeme nový bod. To se dá vyřešit **vymazáním** již nepotřebného **bodu**, který jsme použili někdy dříve, ale dále už jej nebudeme potřebovat, tzn. nepoužijeme jeho jméno v žádném příkaze. Pak stačí napsat **dG** a bod G můžeme používat dále.
- t** Máme-li nějaké dva body, např. A a B, a potřebujeme najít **střed úsečky AB**, stačí napsat **tAB**. Program najde střed této úsečky. Ze středu se stane bod, jehož jméno bude nejnižší volné (dosud nepoužité) písmeno abecedy. V našem případě by měl jméno C. Příkaz „t“ neumí vykreslit samotnou úsečku, jejíž střed chceme spočítat. Je zřejmé, že oba krajní body pomyslné úsečky, jejíž střed chceme spočítat, musí v okamžiku, kdy použijeme příkaz „t“ existovat.
- u** Chceme-li zobrazit (nebo použít jako parametr k výpočtu průsečíků - viz. níže) nějakou **úsečku**, například úsečku AB, stačí napsat **uAB**. Úsečku je navíc možné nechat vykreslit několika typy čar. Pokud chceme použít jiný než obyčejný typ, je potřeba mezi „u“ a jména krajních bodů úsečky napsat znak odpovídající dané čáře. Jsou to:
- „!“ silná čára
 - „#“ čerchovaná čára
 - „%“ čárkovaná čára
 - „\$“ tečkovaná čára
 - „@“ neviditelná čára (použití u průsečíků objektů)
- Všechny možné kombinace jsou tedy následující : **uAB, u!AB, u#AB, u%AB, u\$AB, u@AB**
- p** Parametry jsou stejné jako u úsečky, jen místo úsečky se počítá a zobrazuje **přímka**. Např. **pAB, p!XY, p#KL, p%UV, p\$RS, p@PQ** (body musí opět samozřejmě existovat)
- z** Parametry jsou opět stejné jako u úsečky, jen místo úsečky se počítá a zobrazuje **polopřímka**. Například **z#AB** vytvoří čerchovanou polopřímku AB. V mnoha případech ale potřebujeme polopřímku opačnou k zadané a to se dá provést tak, že za příkaz „zAB“ přidáme „/“, tedy **zAB/**, čímž dostaneme polopřímku opačnou k AB.
- o** Pomocí příkazu „o“ můžeme vytvořit **osu** dané **úsečky**. Například osu úsečky AB sestrojíme pomocí příkazu **oAB**. Tento příkaz nám ji vykreslí, nebo ho můžeme použít jako parametr při výpočtu průsečíků (viz. níže). Typ čáry je samozřejmě možné měnit stejně jako u úsečky, přímky či polopřímky.
- q** Pomocí „q“ můžeme vytvořit **osu úhlu** tvořeného třemi body. Tak například osa úhlu ABC se vypočítá a vykreslí příkazem **qABC**. Typ čáry je samozřejmě možné měnit. Např. **q#AVB, q%UVW** apod.
- b** Máme-li zadány dva body A a B, které nám určují úsečku či přímku, a bod C, kterým chceme vést **rovnoběžku**, pak ji sestrojíme příkazem **bcAB**, tedy „rovnoběžka bodem C s AB“. Typy čar opět můžeme měnit dle libosti. Př. **b%PKL, b\$MNO** apod.
- m** Slouží pro vytvoření kolmice procházející daným bodem (M) a kolmé k nějaké přímce (AB). Takže příkaz, kterým takovou kolmici sestrojíme je **mMAB**. Typy čar opět k dispozici.
- k** Podobně jako můžeme vytvořit přímku či úsečku, můžeme vytvořit i **kružnici**. Kružnici se středem v bodě A a poloměrem velikosti úsečky CD vytvoříme příkazem **kACD**. Kružnice se dá vykreslit pouze normálně, silně nebo vůbec, tzn. že možné příklady jsou následující : **kSUV, k!ERT, k@NNU**. Všechny určující body musí existovat.

c Někdy potřebujeme vytvořit kružnici s poloměrem, který zadáme nikoli jako úsečku, ale jako číselnou hodnotu. K tomu slouží příkaz „c“. Kružnice se středem v A a poloměrem 3.2 dílku sestrojíme cA3.2. Kružnice se opět dá vykreslit pouze normálně, silně nebo vůbec. (cS5.1 , c!E0.3 , c@N5.25).

x Je to jeden z nejdůležitějších příkazů. Umožňuje vytvořit bod (y) jako **průsečík dvou útvarů**. Příkaz funguje tak, že za „x“ následuje první objekt (např. uAB), následuje čárka a pak druhý objekt. Pokud zadáme například x pAB, pDC, vypočítá se nejen průsečík obou přímk, ale zároveň se i vykreslí. Pokud bychom napsali x p#AB, p!DC, vypočítá se nejen průsečík, ale přímka AB se vykreslí čerchovaně, zatímco přímka DC čarou silnou. V případě že průsečík neexistuje (např. průsečík dvou úseček) průsečík se prostě nevypočítá. Pokud chceme pouze spočítat a vykreslit průsečík(y) a nikoliv již dané útvary, musíme v případě obou protínajících se útvarů použít typ čáry @. Př. x p@AB, p@DC

...

Další příklady:

<u>xk!AAB,p@EF</u>	průsečík kružnice se středem v A a poloměrem AB, která se vykreslí silně a přímky EF, která se ale nevyskreslí vůbec.
<u>xkAAB,c!D5</u>	průsečík kružnice se středem v A a poloměrem AB, která se vykreslí normálně, a kružnice se středem v D a poloměrem 5, která se vykreslí silně.
<u>xzEF,u!AB</u>	průsečík polopřímky EF a úsečky AB, která se vykreslí silně.
<u>xp#KL,p#RT</u>	průsečík dvou přímk KL a RT, které se obě vykreslí čerchovaně.

i Chceme-li získat nějaké **informace** o dosud zadaných objektech, použijeme příkaz „i“. Dá se použít takto:

i	vypíše souřadnice všech bodů, které jsme doposud použili.
iA	vypíše souřadnice daného bodu A
iAB	vypíše délku úsečky AB
iABC	vypíše velikost orientovaného úhlu ABC počítáno v kladném směru.

h Velice výkonný příkaz sloužící k **přesouvání** daného bodu. Pokud zapíšeme hA, můžeme pomocí kurzorových kláves přesunout bod A na novou pozici a po stisknutí ENTER se konstrukce přepracuje (pokud přesouváme bod, který není závislý na minulých bodech). To ovšem není všechno. Mnohem přitažlivější by bylo, kdyby se konstrukce přepracovávala postupně, podle toho, kam jsme bod právě přesunuli. K tomu slouží příkaz h!A, který způsobí, že po každém stisku kurzorové klávesy se konstrukce přepracuje. V tomto případě (ale jen v tomto případě) je možné zjišťovat průběžně nějakou informaci. Např. h!A/iAB nám po každém stisku kurzorové klávesy nejen přepracuje konstrukci, ale i vypíše požadovanou informaci.

Je důležité si uvědomit, že při určité poloze zadaného bodu není námi zadaný postup schopen zkonstruovat správné úlohu, takže někdy může vyjít i nesmysl.

a Tento příkaz se používá při vytváření šablon v textovém editoru. Tento příkaz funguje tak, že vše co bezprostředně následuje za „a“ se bere jako komentář, který se vypisuje na příkazovou řádku. V případě aZadej bod A se vypíše Zadej bod A a okamžitě se provádí následující příkaz.

Dá se použít např:

<u>aZadej bod A</u>	Vypíše se Zadej bod A
<u>aA</u>	a pomocí kurzorových kláves zadáme polohu bodu A

V případě, že použijeme a!Ted vytvořím úsečku AB, vypíše se komentář a program čeká na stisk libovolné klávesy.

Příkaz „a“ se tedy využívá při vytváření šablony mimo vlastní program. Předem připraveným komentářem tak můžete komentovat každý jednotlivý krok postupu. (Příklady použití v souborech *.GM2 na disketě).

v **Vypíná** resp. **zapíná** zobrazení osového kříže. To přispěje k lepší přehlednosti na obrazovce.

j Při přesouvání obrazovky se **vypíná** resp. **zapíná vykreslování** celé konstrukce. Takže stiskneme j, přesuneme se po obrazovce na požadované místo (kontrolujeme si to pomocí osového kříže), a opětným stisknutím j se konstrukce opět zobrazí. (U pomalých počítačů je to jediný možný způsob, jak při překreslování nepřijít o nervy.)

e Přesune **střed osového kříže** do **středu obrazovky** (bod o souřadnicích [0,0] do středu obrazovky)

g **Přepočítá měřítko** tak, aby pokud možno všechny body na byly zobrazeny na obrazovce.

+ a - Zvětšení resp. **zmenšení** konstrukce. Konstrukci je možné zvětšit či zmenšit dle potřeby.

„**kurzorové klávesy**“ **Posun celé konstrukce po obrazovce** v patřičných směrech. Spolu se zvětšením či zmenšením je možné se podívat na libovolné detaily celé konstrukce.

l a s Načtení resp. **uložení** souboru. Na požádání napište jméno souboru, pokud chcete, aby se načel (uložil) ze stejného adresáře, ve kterém je GEOM, nebo celou cestu, pakliže tomu tak není. Předem připravený interaktivní postup je možné příkazem **s** uložit pro pozdější použití či demonstraci v hodině.

r Provede se **resetování** programu GEOM. Všechny předešlé příkazy jsou ztraceny, dočasný soubor, do kterého se průběžně všechny příkazy zapisují, je též smazán.

„**F1**“ **Nápověda.**

„**F2**“ Umožňuje nastavit parametry programu. Pomocí kurzorových kláves nahoru a dolů si vybíráme položku, kterou chceme změnit a pomocí zbylých kurs. kláves nebo zadáním čísla měníme, co potřebujeme. Můžeme měnit:

barva pozadí	kurs. kl vlevo a vpravo
barva osového kříže	kurs. kl vlevo a vpravo
barva čar a písma	kurs. kl vlevo a vpravo
zadání meze pro stanovení nuly	-jaké číslo se už má považovat za nulu (standartně 0,0001)
délka kroku při zadávání bodu	-jak velký je krok při jednom stisknutí kurzorové klávesy
počet desetinných míst	-počet des. míst, na které se vypisují informace

„**F3**“ **Vygeneruje** soubor daného jména, který obsahuje konstrukci převedenou do formátu PLT. Takto vygenerovaný soubor se dá použít k vytisknutí konstrukce na plotterech nebo laserových tiskárnách HP. Na disketě je v adresáři PRINTGL nahrán softwarový emulátor plotteru **PRINTGL.EXE**. Tímto emulátorem (shareware) je možné soubory *.PLT prohlédnout na obrazovce a poté vytisknout na tiskárně (nejlépe na laserové HP LJ IIP kompatibilní) v rozlišení až 300 dpi. (Kopii obrazovky nedoporučuji tisknout - brrrr.)

V souboru **CTIMNE.602** (T602) se dozvíte podrobnosti.

„**F8**“ **Vymaže** poslední příkaz (Undo) z dočasného textového souboru. Dá se tak postupně vymazat celá konstrukce.

„**ESC**“ **Konec** programu.

Konkrétní příklady některých příkazů

u!AB	silná úsečka AB
p#EF	čerchovaná přímka EF
c!D5.5	kružnice se středem v D, poloměrem 5.5 jednotky a vykreslí se silně
nG/q#ABC	pomocí kurs. kl. vytvoříme bod G ležící na ose úhlu ABC, která se vykreslí čerchovaně
xp@AB,m@DEF	průs. přímky AB a kolmice bodem D na EF, z nichž ani jedna se nevykreslí
xk@EFD,z%GH	průs. kružnice se středem v E a poloměrem FD, která se nevykreslí, a čárkované polopřímky GH

Příklady příkazů pro sestrojení některých úloh

Kružnice trojúhelníku opsaná

nA(-5,-5)
nB(6,-1)
nC(1,5)
uAB
uBC

uAC
xo#AB,o#AC
o#BC
k!DDA

Kružnice trojúhelníku vepsaná

nA(-5,-5)

nB(6,-1)
nC(1,5)
uAB
uAC
uBC
xq#ABC,q#BCA
q#BAC
xm@DAB,p@AB
k!DDE

a!Kruznice vepsana
aZadej bod A (pomoci kurs. kl.)
nA
aZadej bod B (pomoci kurs. kl.)
nB
aZadej bod C (pomoci kurs. kl.)
nC
a!Vytvorim trojuhelnik ABC
uAB
uAC
uBC
a!Sestrojim prusecik os uhlu
xq#ABC,q#ACB
q#BAC
a!A ted najdeme polomer kruznice vepsane
xm@DAB,p@AB
u!DE
a!Ted muzu vytvorit hledanou kruznici
k!DDE
a!A uloha je hotova

Ukázka šablony pro sestrojání kružnice vepsané

Takovýto soubor, který může být vytvořen jednoduchým textovým editorem(Norton Comander Shift+F4) a posleze načten programem GEOM by nás provedl konstrukční úlohou krok za krokem

Různé poznámky na závěr :

- * Při zadávání příkazů je nejlépe nepoužívat mezi jednotlivými písmeny mezery, samozřejmě kromě komentáře u příkazu „a“.
- * Při chybových hlášeních čeká program většinou na stisk klávesy a teprve poté pokračuje v normální činnosti. Pokud se program neukončí klávesou <ESC>, ale náhodou „spadne“ (občas se stane), zůstane na disku soubor DOCASNY.GM2, který obsahuje příkazy zaznamenané až do okamžiku havárie.
- * Do souboru se ukládají příkazy : n, d, t, u, p, z, m, q, o, b, k, c, x.
- * Pro urychlení programu je dobré používat „kešovací“ SMARTDRIVE (viz. AUTOEXEC.BAT a CONFIG.SYS).
- * Další příklady a demonstrační ukázky jsou na disketě v souborech *.GM2, které je možné si prohlédnout či načíst do T602 a vytisknout.
- * V *.GM2.souborech se používá kodování bratří Kamenických !!! (Nesehnal jsem jiný ovladač češtiny do TURBO-PASCALU). Pokud snad máte něco proti Kamenickým, pište tedy bez HACKU a bez CAREK.

Program GEOM dávám k dispozici jako **freeware** - lze jej tudíž šířit bez omezení a bezplatně. Nepřeji si, aby byl komerčně kýmkoliv využíván a aby byl jakoliv pozměňován obsah demonstračních či dokumentačních souborů na distribuční disketě.

Libor Švéda

Ukázka možností programu GEOM :

Vygenerované PLT soubory je možné pomocí emulátoru PRINTGL vytisknout na různých typech tiskáren. Na obrázcích je ukázka tisku připravených demonstračních úloh na laserové tiskárně HP4L (300x300 DPI). Původní velikost každého z obrázků je A4 !!! Nižší kvalita uvedených obrázků je způsobena scanováním, mnohonásobným zmenšením a opětovným tiskem (pozn. aut.).

