

(aneb jsme o rok chytřejší ?)

Ing. Dušan Streit

AGRISOFT, kooperační sdružení, Frýdek-Místek, Beskydská 2062

1. Smysl příspěvku

Na tomto semináři - o jehož úrovni nejlépe svědčí, nemýlím-li se, již patnáctý ročník, a to za neutuchajícího zájmu programátorů a příbuzných profesí - mi poněkud vadila jedna objektivní skutečnost: Každoročně se v bohaté diskusi vykrystalizuje jistý převládající názor. Vzhledem k obměně účastníků se však příští rok vrací diskuse, pokud jsou otázky stále akutní, o kousek zpět a někdy se znovu " objevuje již loni objevené ". Ideální by bylo, kdyby z průběhu každé ročníku semináře bylo zpracováno jakési resumé, s jehož obsahem by se seznámili všichni účastníci následujícího ročníku. Chápu, že je to technicky těžko proveditelné a není v možnostech pořadatelů ustavovat pracovní komise pracující v období mezi semináři.

Přesto, že si nečiním nárok na objektivní a vyčerpávající shrnutí, chtěl bych se dotknout s odstupem času některých otázek, které mě zaujaly jako dilemata " Programování "88 ". Třebaže se v některých pasážích pokusím o jakési shrnutí a třebaže i budu opakovat některé myšlenky z loňské diskuse, na které si nečiním autoraký nárok, chápejte moje vystoupení jako subjektivní příspěvek k průběhu letošního semináře. Cílem příspěvku je poskytnout výchozí základnu dalším diskusím, aby přinesly zase nové myšlenky.

2. Dilemata a dialektika

Ve svém loňském příspěvku /1/ jsem se zabýval " dialektikou ASŘ ". Na rozdíl od dialektiky představuje dilema volbu ze dvou vzájemně se vylučujících možností. Řečeno exaktně: zatímco dilema nepřesahuje rámec Booleovy algebry (vztah ano - ne), dialektika přesahuje rámec fuzzy množin (vztah vágní, neostrý, rozmazaný).

Řečeno metaforicky: dilema reprezentuje dialektiku " černobílý svět " zastavený v pohybu, dialektika představuje " barevný svět " pohybující se po spirále vývoje. Před dilema je člověk postaven tehdy, když nemá příliš času ani možnosti na výběr - na jednoduchou otázku hledá jednoznačnou odpověď. Z historického hlediska však zdánlivě jednoznačné odpovědi mohou být snadno postaveny na hlavu. I takové kategorie jako pravda a nepravda mohou být po čertech relativní. Dynamiku tohoto projevu respektuje dialektika, která by se měla vyhýbat černo-bílému vidění světa. V lůně černého se rodí bílé a dále si to představme jako iteraci negace (tak si jako programátor představují zákon negace negace). Není to alibiismus ani vyhýbání se rozhodnutí. Ale každá taková " křižovatka ", kde jsou dvě možnosti a jedna se nabízí jako jediné správná, je mi předem podezřelá. Změní se pak jeden jediný parametr, jehož hodnotu dosadí čas nebo vyšší stupeň poznání a plus se snadno zvrhne do minusu a naopak.

Proto proklamace o tom, které metody a postupy jsou jediné správné, je třeba brát s rezervou. Dnes tomu tak může být, ale jestliže chceme poznat, jak to bude zítra, podívejme se, jak to bylo včera a jaké má tato " iterace " zákonitosti. " Včera " se například po infarktu doporučoval klid, " dnes " se doporučuje přiměřená aktivita. Ale není to jenom proto, že dříve se lékaři svlíkli a dnes mají pravdu, ale taky změnami úrovně lékařské péče v celém komplexu. A těchto souvislostí bychom si měli být vždy vědomí. Dilema " ležet jako placka " versus " běhat jako s keserem " je nesmysl. Přesto se však v našem oboru takového schematizování dopouštíme.

Proto následující dilemata chápejte jako úvahy, které by měly přispět k pochopení širších souvislostí a ne k nalezení absolutní pravdy. Každou zákonitost je třeba nejdříve poznat a naše svoboda je v umění využít možného. Přijivme se tedy na dialektice.

3. Shora dolů či shora nahoru

V literatuře /2/ a /3/ byl položen velice solidní základ pro diskusi. Přesto jsem se neubráníl dojmu, že autor velice jednostranně posuzuje dnešní úlohu metody shora dolů, jako by již její možnosti byly vyčerpány a bylo by tedy třeba podpořit její dialektický protějšek. Mochci polemizovat hned na začátku, uvažujte tedy se mnou (a s dalšími diskutujícími loňského semináře, jejichž vyřčenky jsem si vypůjčil).

3.1. Metoda shora dolů

Metoda shora dolů je založena na tom, že využijeme danou řešitelkou ("součástkovou") základnu; tato zůstává na relativně nejobecnější úrovni. Konkrétní úlohu pak hierarchicky dekomponujeme a kritériem dekompozice není řešitelská základna, ale úloha sama, s cílem, aby se postupným zjemňováním problému co nejvíce zjednodušil. Ale pozor, u dekompozice dochází často k nepochopení této metody: rozhodující není dílčí část dekomponovaného systému, ale hierarchické vazby. Tyto externí vazby jsou určující pro definování podřízených složek, na které se tudíž můžeme dívat jako na černé skříňky s determinujícími parametry. Na dané rozlišovací úrovni se tedy můžeme abstrahovat od vnitřní realizace podřízených složek, musíme je však definovat takovou konstrukcí vazeb (přísunem aktuálních parametrů), abychom neopominuli nic z toho, co naší rozlišující úroveň determinovalo shora. Jako bychom dílčími prostředky řešení postupně substituovali výchozí cíl. Metoda shora dolů tedy zůstává na procedurální úrovni řešení, nelse ji chápat jako prosté sjednocení dílčích funkcí.

Zabezpečení vazeb a předávání parametrů musí být nějak koordinováno a řížováno. V programu tuto funkci zastane kořenný segment (může být i hierarchicky rozptýlen a nebýt tedy v plném rozsahu permanentní). Nad úroveň programu musí při projektování tuto úlohu plnit některý řešitel (nebo i hierarchicky organizovaný tým řešitelů); nesmí však pracovat administrativní metodou " rozděl a panuj ", ale s vědomím, že uhlídání všech vazeb je řešitelsky nejnáročnějším a nejpodstatnějším problémem.

Tato metoda je jakousi paralelou demokratického centralismu. Směřuje od cílů k prostředkům. Dílčím závěrem může být to, že je pro konkrétní úlohy oprávněně preferována. Kdybychom ji však uplatnili ad absurdum, dodnes bychom plácali stavby z písku, vody a dalších základních surovin.

3.2. Metoda zdola nahoru

Metoda zdola nahoru znamená, že povyšujeme řešitelskou základnu na syntetičtější úroveň a součástkovou základnu tedy prefabrikujeme. Také při uplatnění této metody můžeme postupovat hierarchicky. Ta úplně spodní základna řešení je obecnější a rozsahem širší, než konkrétní úloha vyžaduje. Postupně ji syntetizujeme, konkretizujeme a zužujeme a tím si vytváříme mocnější nástroje řešení. Mocnější, ale účelovější a čím dál tím méně procedurální. Abstrahujeme-li se od vnitřní realizace, více to připomíná skládání funkcí, které nemají být vnitřně zkoumány. Tím, že se blížíme od spodu k úloze, která je na vrcholu, vazby jsou nepřímé - jako bychom postupně překonávali propast od obecného ke konkrétnímu. Z hlediska dané úlohy nemáme záruku té nejefektivnější cesty, existuje nebezpečí, že " pro stromy nevidíme les ". Na dané úrovni řešíme tedy celou vrstvu pomocí obecnějších (elementárnějších) prostředků, které jsou již vyřešeny na nižší úrovni. Ale teprve na vyšší úrovni se prokáže, zda byly vyřešeny vyhovujícím způsobem pro další krok směrem ke konkrétnímu problému. Řešitelská základna se však nedá zvýšit až na hranici konkrétní úlohy. Jestliže základnu povýšíme příliš blízko konkrétní úloze, ztratí se původní smysl základny jako obecného nástroje.

Musíme si uvědomit, že postupem zdola nahoru obohacujeme " obsah " a omezujeme " rozsah " (viz. a.ributy " pojmu " jakožto filosofické kategorie). Kritériem musí být obecnost, která v tomto kontextu znamená opakování (opakované využití). Shora k základně, kterou chceme povýšit, musejí tedy přicházet impulsy ne z jediné konkrétní úlohy, ale z určité třídy úloh, obecně chápaných. (Jak prohlásil anglický lord: " Nezná Anglii, kdo zná pouze Anglii ".) Proto považují za chybné nalézat " obecná

řešení " - a metodou zdola nahoru je realizovat - v prostředí konkrétních úloh se snahou je přenášet do úloh odlišných. Tento přístup bych nazval agendovým, jakožto protiklad přístupu technologického, který by měl provázet uplatnění této metody.

Agendový přístup v kombinaci s metodou zdola nahoru vede k tomu, že příliš pozdě zjistíme, že syntetizovaná řešitelská základna nevyhovuje pro jiné konkrétní řešení, nebo že pro přílišnou prefabrikaci není efektivní. Musíme se pak spustit o nějakou vrstvu níže, tedy k pracnější části pyramidy.

Je třeba upozornit na to, že povyšování řešitelské základny sice vede k potencionálně většímu komfortu při řešení, ale vždy za cenu nižší vnitřní efektivity oproti ideálnímu řešení s využitím základny nižší. Proto je třeba postupovat kompromisně, systémově a technologicky. Základna je vhodné povyšovat až tehdy, když chybějící vyšší úroveň se projevuje mechanickým opakováním nižších úrovní.

Demonstrujme si problém na dětské stavebnici se součástkovou základnou na úrovni šroubků, maticek, osiček, koleček a dalších elementárních komponentů. Součástí této stavebnice bývá vždy vyobrazení možných sestav. Je tedy známo předem, pro jakou třídu řešení je stavebnice určena. Je však také ponecháno místo vlastní fantazii. Jestliže by díly byly prefabrikovány do komplexnějších nedělitelných celků, měli bychom snadnější práci při skládání, ale zároveň zúžený výběr možných sestav. Ad absurdum by se nejednalo o stavebnici, ale řekněme o konkrétní hračku. Ze stejných důvodů zneužití metody zdola nahoru musíme kvůli haléřovým položkám kupovat náhradní díly v cenách až o několik řádů vyšších. Proto k efektivnímu užití metody zdola nahoru patří i normy a standardy.

Tato metoda je jakousi paralelou prototypového přístupu. Směřuje od prostředků k cílům. Dílčím závěrem může být to, že pro úlohy obecné či mlhavě definované je oprávněně preferována. Kdybychom jí však uplatnili ad absurdum, znamenalo by to konstrukci zvláštního počítače na konkrétní úlohu.

3.3. Dialektika obou metod

Podíváme-li se dialekticky na vztah obou metod, můžeme je chápat jako odraz těchto vzájemně souvisejících pojmů:

- a) obecné - zvláštní,
- b) abstrakce - determinace,
- c) pyramida - strom,
- d) syntéza - analýza,
- e) kompozice - dekompozice,
- f) funkční - procedurální,
- g) indukce - dedukce.

Pozorný čtenář vidí, že první pojem vždy souvisí s metodou zdola nahoru, druhý s metodou shora dolů.

Díleční poznámky:

ad a)

Obecnost pojmu (úlohy) je tím menší, čím je pojem (úloha) obsažnější (sématicky plnější) a rozsahem aplikace (pragmaticky) užší. Obecnost je tedy nepřímo úměrná "obsahu" a přímo úměrná "rozsahu". O obecném řešení můžeme mluvit tehdy, jestliže reprezentuje alespoň jistou třídu problémů. Čím jsou nástroje řešení syntetičtější a sématicky plnější, tím jsou účelovější. Hranici mezi obecným a zvláštním by měla tvořit řešitelská základna. Při povyšování řešitelské základny (postup zdola nahoru) je třeba upozornit na nebezpečný paradox: Pokud zobecnujeme využívání stávající řešitelské základny v opakovatelné standardní prefabrikáty, opakovatelnost je zabezpečena vnitřní unifikací řešení, ne už tak větším rozsahem využití (viz. např. normované programování, makroinstrukce nebo i vyšší programovací jazyky). Tato vyšší řešitelská základna je v konečném důsledku vždy účelovější než ta původní. Metaforicky řečeno je obecné uvězněno uvnitř a zvláštní se dere ven. Tento paradox má důsledek v tom, že zatímco bychom očekávali, že nejširší - co do počtu prvků - je základna nejelementárnější, bývá zpravidla opak pravdou. Souvisí to s tím, že zdola nahoru klesá potencionální využitelnost (obecnost) základny. V Mendělejevově periodické soustavě je také více prvků, než jejich stavebních částí (protony, neutrony, elektrony).

Vyváženost mezi vnitřním obsahem a vnějším rozsahem, mezi účelovostí a opakovatelností je měřítkem efektivní obecnosti řešitelské základny. Protože to nejobecnější znamená v tomto kontextu to nejelementárnější.

ad b)

K obecnému dospíváme abstrakcí (odhlážením od některých znaků), ke zvláštnímu determinací (přidáváním znaků). Při abstrakci tedy odhlážíme od nepodstatných znaků a tento postup je předpokladem pro tvorbu obecné řešitelské základny. Musíme si však uvědomit: čím abstraktnější - tím blíže hardware. Abstraktní úloha je pro nás taková, kterou lze formulovat jako "řešitelnou na počítači". Musíme tedy postupně přidávat takové znaky, které pro naše prostředí mají význam (a nepodstatné znaky ze svých úvah eliminovat). Determinujeme tak vhodný typ počítače, operační systém atd.

ad c)

Metodu zdola nahoru si můžeme představit jako pyramidu, kde jednotlivé vrstvy tvoří hierarchii základny. Z hlediska "rozsahu" stojí pyramida na své nejširší základně, z hlediska "obsahu" na vrcholu. Jsme-li abstraktní až do krajnosti, od spodu jsou vrstvy tvořeny takto (jsou odděleny lomítkem): KVARKY, LEPTONY/ELEKTRONY, NUKLEONY/ATOMY-PRVKY/MOLEKULY-SLOUČENINY/SUROVINY/...
.../HARDWARE/SOFTWARE/KONKRÉTNÍ ÚLOHA-MODEL. Viz obr. 1.

Připomeňme se pouze, že vrstvy v základně nemusí být striktně odděleny přímkou (spíše lomnou čarou nebo křivkou). Tvorbou vyšší základny nemusí totiž ztratit význam všechny prvky nižší základny. Např. výrobou panelů neztrácí svoji přímou upotřebitelnost ani beton apod.

Naproti tomu postup shora dolů reprezentuje strom. Zásadou je, aby všechny vazby byly vertikální, aby tedy při dekompozici horizontální vazby nehrály žádnou roli. Určitým prohřeškem proti čistotě této metody je, jestliže nižší prvek je provázán s více než s jedním vyšším prvkem. I v tom je však dialektika tvůrčího procesu v reálných podmínkách. Teprve až postup shora dolů naváže na základnu, stává se z opakovaného využití řešitelských prostředků pravidlo.

ad d), e) - viz. literatura /1/ zejména obr. na str. 15

ad f)

Při metodě shora dolů postupujeme od cíle, který přesně známe, k prostředkům řešení. Víme tedy "co", ale nevíme ještě podrobně "jak". U každého nižšího prvku jsou rozhodující jeho vnější (funkční) vlastnosti, které se realizují ve vnitřní (procedurální) struktuře vyššího prvku. Protože postupujeme shora dolů, určující je procedurální hledisko; konstrukce (vnitřní stavba) nadřazeného prvku je rozhodující pro funkční vlastnosti podržčeného prvku.

Při postupu zdola nahoru je tomu opačně. Postupujeme od stávajících prostředků, které přesně známe, k cíli, který je vzdálen. Víme tedy, jakých základních prostředků použijeme, ale nevíme podrobně, co budeme řešit. U vyššího prvku je rozhodující jeho vnitřní stavba, která je zabezpečena funkčními vlastnostmi nižších prvků. Protože postupujeme zdola nahoru, určující je funkční hledisko; funkční vlastnosti nižších prvků jsou limitující pro konstrukci (vnitřní stavbu) vyššího prvku.

ad g)

Citace z literatury /4/: " Přes různé pokusy vést ostrou hranici mezi dedukcí a indukcí jsou obě metody ve skutečnosti vnitřně spjaty ".

3.4. Vzájemná podmíněnost obou metod - směr dilemat

Obě metody nejen že jsou dialekticky závislé, ale jedna bez druhé ani nemá význam. Metoda shora dolů by sama o sobě znamenala dekompozici do takové hloubky, ve které bychom se snadno "utopili", protože od problému k základně by nebylo možné dohlédnout. Uvědomme si, že jako řešitelská základna musí pro nás existovat minimálně počítač s instrukční sítí, dnes již i programovací jazyky a operační systémy a další podpůrné prostředky. Nehledě na to, že individualistický přístup k vytváření "součástkové" základny shora by vedl k anarchii v řešitelských prostředcích.

Stejně tak metoda zdola nahoru není prakticky sama o sobě schopna dospět na vrchol řešení. Shora dolů musí chodit impulsy o cílech řešení. Nehledě na to, že řešitelská základna vzniklá přeceněním této metody by byla obecně nepoužitelná.

Překvapivě tak docházíme ke zjištění, že extrémy v obou případech vedou ke stejným negativním důsledkům. Nejen že obě metody na sebe navazují, ale dokonce se prolínají. Např. modulární programování jistě vychází z postupu shora dolů. Ale jestliže při tom uplatníme aspekt zobecnění v tom smyslu, že modul je v programu nebo i mimo program opakovaně volán z různých míst, povýšíme si tak řešitelskou základnu a vlastně tedy uplatníme přístup zdola nahoru. Analogicky, jestliže záměrně povyšujeme řešitelskou základnu, řekněme formou utility, která je příliš rozsáhlá, můžeme si při její řešení pomoci postupem shora dolů. Často si pomáháme parametrizací a opět jí můžeme chápat v obou souvislostech. Při postupu shora dolů jsou parametry nositeli vazeb, tedy subjekty řešení, při postupu zdola nahoru parametry vytvářejí vyšší hladinu řešení, jsou tedy objekty řešení.

Celý tvůrčí proces má iterační charakter. Dilema netkví v tom, zda shora dolů nebo zdola nahoru, ale zda postupujeme systémově či nikoliv. Jedná se o to, aby iterace neprobíhala živelně od extrému k extrému s řešitelskými multiplicitami, ale po spirále vývoje.

K tomuto problému bych chtěl pod čarou poznamenat, že ačkoliv teď metoda zdola nahoru získává na popularitě díky prototypovému přístupu a programovému vybavení personálních počítačů, vnitřní zdroje metody shora dolů nejsou zdaleka vyčerpány, a to především v organizaci práce řešitelských týmů. Při řešení rozsáhlých systémů je možné použít stejných metod jako v rámci programu se všemi důsledky pro efektivní dělbu práce mezi objekty i subjekty řešení.

4. Dialektika vztahu řešitel - uživatel

Především je třeba říci, že ASŘ se musí přizpůsobit podniku, ve kterém je zaváděn, a to ve smyslu priority jeho produkční a hospodářské funkce. Neplatí to však už tak zcela o podnikovém informačním a rozhodovacím systému, tedy o jeho administrativě; pokud ta se vyvinula v prostředí, které nepočítalo s automatizací těchto činností, musí se administrativa přizpůsobit novým podmínkám, jinak by nebyly využity potencionální možnosti výpočetní

techniky. Může to znamenat změnu v oběhu dokladů, formě a obsahu řízení a kontroly, ale také v odměňování a uplatňování chozrasčotu a norem, ve funkčních schématech a pracovních náplních apod. Výpočetní technika je stejně tak prostředkem k zabezpečení základních funkcí podniku jako administrativa a celá podniková nadstavba. Upozorňují na to proto, že podniky nebývají většinou připraveny k těmto změnám, které by teprve zhodnotily investice do výpočetní techniky. Jako by nákupem výpočetní techniky se podnik zbavil odpovědnosti za její využití a zcela ji přenechal na řešitele. Ti ovšem nemají dostatek pravomocí k tomu, aby zabezpečili změny, na které jsem upozornil. Navíc se podceňuje řešitelská, implementační a aplikační pracnost a přeceňují se okamžité možné efekty, které jako by čekaly jen na to, až se počítač "připojí do zásuvky". Optimismus pak přechází v rozčarování, které dopadne především na řešitele. Málokdy se dialekticky chápe oboustranná nutnost přizpůsobování se podnikové infrastruktury a ASŘ.

V tomto kontextu nabývají nového významu zásadní projekční materiály, tedy projektový úkol a úvodní projekt systému, proti jejichž přísně systémovému pojetí a přístupu shora dolů se hledaly protiargumenty a teď se často nacházejí v prototypovém přístupu. Bylo již vysvětleno, že tento rozpor má dialektický charakter a nic nemění na tom, že projekty, které vymezují koncepci, prostředí a předpoklady, mají klíčový význam. Tuto koncepci musí navrhovat odborníci, kterým podnik vytvoří takové prostředí, kde jí bude možno realizovat. Je třeba zde uplatnit stejnou míru pravomocí a odpovědnosti. Nelze před řešitele klást nepřekonatelné bariéry "objektivních překážek" a subjektivních zájmů, kdy do koncepce zasahuje tolik nekompetentních (ale vlivných) osob, že se původní koncepce zvrhne ve svou karikaturu. Ne každý řešitel je hrdina, aby si "svůj" projekt a své profesionální postavení uhájil a ve jménu idejí a dobré věci nesl svou kůži na trh. Role řešitelů často připomíná pověstného Dona Quijota a jeho boj s větrnými mlýny nebo také bájněho Sisyfa.

V této souvislosti je třeba upozornit na vytvořivší se jakýsi "kult uživatele", který i řešitel uznává a je nucen se uživateli podbízet se všemi důsledky nerovnoprávného vztahu.

Samozřejmě, že teprve v uživatelském prostředí se realizuje ASŘ a nic nelze namítat ani proti "uživatelsky přítulným" systémům. Odpovědnost, a proto i rozhodující slovo, by však měl mít řešitel. Uživatele reprezentuje ne konkrétní osoba, ale celý podnik. Konkrétní člověk může mít obavy z přeškolení, z nové náplně práce, z toho, že prostřednictvím počítače mu bude vidět do karet, hrají zde roli i vnitropodnikové zájmy např. ve vztahu chodrasčotu a vnitropodnikové fakturace, uživatel může mít strach ze ztráty svého postavení apod. V rukou těchto lidí tedy nemůže být osud ASŘ, a proto podpora řešitelů ze strany vedení podniku je nezbytná. Stejně tak administrativní zásahy do koncepce shora jsou nepřijatelné.

Znovu je třeba opakovat, že zpracování dat je výrobním procesem. Stejně jako ve sféře přípravy materiální výroby má rozhodující slovo technolog s ekonomem, musí mít obdobné postavení i řešitel. Automatizace představuje nejen vyšší stupeň mechanizace, ale i změnu technologie. Dělníci si také nemohou vybírat, na kterém stroji, lince budou pracovat a na kterém nikoliv. Historicky se zde vyvinula samozřejmá technologická kázeň. V okruhu TH pracovníků tomu tak ve vztahu k výpočetní technice není. Zde se nacházíme v etapě skryté formy ludditského hnutí. Takto by se socialistická demokracie projevovala a v západních zemích se ani nepřipouští.

Tedy počítač tady je pro uživatele (ne aby měl programátor práci). Ale není tady ani pro uživatele jako osobu, tedy subjekt řízení, ale pro podnik, tedy objekt řízení, jehož je součástí.

5. Ještě několik poznámek k vztahu angličtina - čeština

My programátoři už se musíme smířit s tím, že naším světovým jazykem je angličtina. Za "kosmopolitismus" se už dnes netrestá.

Ve vztahu k uživatelské komunikaci se často diskutuje o formální a jazykové stránce dialogu. Je třeba si uvědomit, že komunikaci s uživatelem částečně zabezpečují jednotlivé programy, částečně jazyk řízení prací nebo též procesor uživatelských

příkazů, dále monitor operačního stavu a jeho obyčejné rutiny. Samozřejmě bychom se měli snažit o homogenní český (nebo slovenský) dialog a odfiltrovat cizí texty. Pochopitelně to neznamená překládat operační systém, ale takový způsob dialogu, že pokud "spadne" uživatelská komunikace do angličtiny či jiného cizího jazyku, bude to pro uživatele signalizovat výjimečný stav. Musí se pak obrátit na odbornou obsluhu systému.

Pokud se týká textových editorů zde již myslím problém české komunikace přestává existovat, jak bylo na tomto semináři již nejménou prezentováno odborníky s vlastními zkušenostmi.

6. Závěrem

Jistá roztržitost mého příspěvku souvisí s naznačeným záměrem poskytnout základ letošní diskusí na bázi mého subjektivního shrnutí některých otázek diskuse loňské. Tento příspěvek je však i pokračování linie, kterou jsem prezentovat loni v /1/, a sice poskytnout podklady pro tvrzení, že černo-bílé kontradiktorní vidění světa je v našem oboru škodlivé.

Literatura

- /1/ Lexa I.: Metodická ASŽ, Sílina. Programování "88, DT ČS VTS Ostrava, 1986
- /2/ Lexa I.: Shora-dolů nebo zdola-nahoru. Sborník Programování "87, DT ČS VTS Ostrava 1987
- /3/ Lexa I.: O jednom riziku postupu shora-dolů. Sborník Programování "88, DT ČS VTS Ostrava 1988
- /4/ Malý encyklopedický slovník A - Ž
ACADEMIA 1972

