

Josef H R A D I L

ZVT - OKR Ostrava

## Výběr a výuka nových programátorů v ZVT OKR

Neustálý růst počtu samočinných počítačů vyvolává potřebu vychovávat nové programátory a pracovníky pro obsluhu počítačů. Je zřejmé, že tato úloha přísluší školám, z části vysokým, ale ve větší míře středním školám. Kdyby školy tuto úlohu dobře plnily, nemuseli bychom se zabývat otázkou výběru a výuky programátorů, jak jsme nuceni toto provádět v našem závodě. Nedostatek dobrých a zkušených programátorů zvětšuje rovněž značnou fluktuauci těchto pracovníků, kteří se snaží opustit tuto náročnou profesi a přejít na duševně méně náročnou práci.

Na některých vysokých školách jsou studenti seznamováni se základy programování samočinných počítačů, učí se programovat v jazyku Algol-jazyku v praxi zřídka užívaném. Cvičné úlohy jsou obvykle numerickým řešením běžných příkladů z matematiky. Vysoké školy tak orientují studenty směrem k vědecko-technickým výpočtům, zatím co oblast hromadného zpracování dat, která v praxi převládá, je opomíjena.

Mnohem horší je však stav výuky programování na středních školách. Dá se říct, že střední školy dosud neprodukují potřebné kádry pro obsluhu a programování samočinných počítačů. Podle informace pro žáky 9. tříd ZDS zřizuje se ve školním roce 1976-1977 nový studijní obor ve skupině studijních oborů 63 - Ekonomika

a organizace, obchod a služby - nazvaný Výpočetní technika a zpracování informací. V Severomoravském kraji je zřízen tento obor pouze ve Valašském Meziříčí, a to je jistě málo. V Ostravě existoval abiturientský kurs pro obsluhu a programování počítačích strojů, ten byl však před několika roky přemístěn do Val. Meziříčí, k velké škodě výpočetních středisek na Ostravsku. S absolventy tohoto abi-kursu máme ty nejlepší zkušenosti. Jeho přemístěním jsme ztratili zdroj kvalitních kádrů nejen pro programování, ale i pro obsluhu samočinného počítače.

Přemístěním zmíněného abi-kursu nám vyvstaly problémy při doplňování programátorských kádrů. Odbor programování se rozšiřoval a nezbyvalo než přijímat uchazeče, kteří se dosud s výpočetní technikou nesetkali. Vznikly problémy se zácvikem. Protože se brzy zhroutil představa, že dobře programovat se naučí každý, začal se používat výběrový test ICL. Upravoval se postup zácviku, náplň i forma výuky. Ve spolupráci s pracovníky Oberového psychologicko-sociologického střediska OR OKR jsme dospěli k nynějším představám o optimálním způsobu výběru a zácviku nových programátorů. Těmito otázkami se chci zabývat v tomto mém příspěvku.

### 1. Posouzení psychické způsobilosti uchazečů na profesi programátora

Literatura, která je z oblasti psychologického výběru k dispozici, a do jisté míry i naše zkušenosti vedou k závěru, že práce v profesi programátora je náročná nejen na odborné znalosti, ale rovněž na psychické schopnosti a vlastnosti pracovníků.

Jaké by měl mít schopnosti ideální programátor? Jsou to: abstrakce, chápání vztahů, usuzování, praktické početní myšlení, teoretické početní myšlení, tvořivost, pružnost, kritičnost, smysl pro detaily, jasnost vyjadřování.

Z osobních vlastností jsou významné tyto: emocionální stabilita, odolnost vůči zátěži, klid, sebevědomí, samostatnost, důslednost, ochota ke spolupráci.

Výčet potřebných schopností a osobních vlastností je takový,

že se jimi může pochlubit jen malý okruh jedinců. Větší je okruh těch, kteří se uvedeným požadavkům přibližují. Vybrat vyhovující jedince, to je úkolem výběrových testů. Je si však třeba uvědomit, že žádný výběrový test nedává 100% spolehlivost, že je zde jistá pravděpodobnost, kdy nadprůměrný jedinec z důvodů okamžité indispozice při testu neuspěje. Opečný případ je však méně pravděpodobný. Test ukazuje předpoklady uchazeče, nezaručuje však dobrou pracovní morálku a tím i dobré pracovní výsledky.

### 1.1 Výběrový test firmy ICL

Od roku 1968 je v našem závodě k dispozici výběrový test pro programátory anglické firmy ICL.

Test obsahuje 10 úloh, které mají být vyřešeny během jedné hodiny. Úlohy se týkají manipulací s čísly v příhrádkách pomocí čtyř základních aritmetických operací. Správné vyřešení úlohy na minimální počet instrukcí je hodnoceno deseti body, za každou instrukci navíc se sráží jeden bod. Čas a obtížnost úloh jsou voleny tak, aby jen mimořádně schopný jedinec dokázal vyřešit všech deset úloh.

Výběrový test ICL nám nepostihuje všechny vlastnosti a schopnosti potřebné pro úspěšný výkon v programování. Ukazuje některé významné schopnosti jako je usuzování, praktické početní myšlení a tvořivost. Jiné potřebné vlastnosti a schopnosti však téměř nepostihuje, např. jasnost vyjadřování, abstrakci, odolnost proti zátěži, důslednost a jiné.

Domnívám se, že tento test dobře postihuje hlavní schopnosti potřebné pro programování malých detailů a tudíž úlohy pro programování ve strojově orientovaném jazyku. Je méně úspěšný v odhalování schopnosti pro sestavování složitějších a rozsáhlejších algoritmů.

### 1.2 Ověřená spolehlivost testu ICL

I když je test ICL v ZVT k dispozici od roku 1968, byl důsledněji uplatňován až od roku 1973. Od tohoto roku máme uchovány výsledky všech testovaných uchazečů. Předem chci však upozornit

na tu skutečnost, že i když každý uchazeč byl testován, váha výsledku testu pro přijetí rostla postupně, takže prakticky teprve loňského roku měly výsledky testu přiměřenou důležitost.

V současné době máme uchovány bodové výsledky 52 uchazečů na profesi programátora. Z tohoto počtu je 21 vysokoškoláků. Byli testováni i vysokoškoláci, i když by se zdálo, že je to zbytečné vzhledem k vysokým požadavkům na uchazeče o studium na vysoké škole. Ukázalo se, že všichni vysokoškoláci neprokázali testem očekávanou nadprůměrnost.

Několik čísel získaných z dosažených výsledků:

Průměrné bodové hodnocení:	42,3	bodů
Průměr vysokoškoláků:	51,3	"
Průměr středoškoláků:	36,2	"
Dosažené minimum:	10	"
Dosažené maximum:	86	"

K těmto číslům je možno dodat, že vyšší průměr vysokoškoláků je zcela logický daný výběrovostí vysokých škol, že z dřívějších testů jsou známy i vyšší hodnoty (92, 98). Uvedená čísla neukazují, že několik vysokoškoláků nedosáhlo 30 bodů, jeden získal pouze 17 bodů.

Z testovaných 52 uchazečů bylo přijato 17 s bodovým průměrem 43, což dokazuje, že výsledky testu měly informativní charakter a neovlivňovaly zásadně rozhodnutí o přijetí (to platí především o letech 1973 - 1974).

To, že byli přijímáni pracovníci s širokou škálou bodových výsledků, umožňuje zpětně posoudit úspěšnost a vhodnost výběrového testu ICL. Pracovní výsledky přijatých pracovníků na jedné straně a bodová hodnota na druhé straně poskytla učinit tyto závěry:

U pracovníků, kteří nedosáhli 30 bodů jsou pracovní výsledky podprůměrné nebo sotva průměrné. Pracovníci nad 55 bodů mají předpoklady pro úspěšnou práci v programování. Ti, kteří jsou svou bodovou hodnotou uvnitř intervalu 30 - 55 bodů, jsou převážně průměrnými programátory; někteří z nich jsou podprůměrní, ale několik lze považovat za nadprůměrné. Bodový rozsah 30 - 50 tvoří pásmo, které neumožňuje dostatečně posoudit vhodnost uchazeče pro pro-

fesi programátora. Více než 50% testovaných leží v tomto intervalu neurčitosti.

Celkově lze přijmout závěr, že výběrový test ICL dává ve značném procentu případů nespolehlivou odpověď v otázce vloh pro programování.

### 1.3 Návrh zlepšeného způsobu výběru programátorů v ZVř OKR

Větší váha výsledků testu ICL pro rozhodnutí o přijetí a skutečnost, že test ICL má výše uvedené nedostatky, nás vedly ke snaze doplnit patřičně výběrový test ICL s cílem snížit pásmo neurčitosti.

Poněvadž jsme si byli vědomi náročnosti tohoto úkolu, zainteresovali jsme na řešení pracovníky Oborového psychologicko-sociologického střediska OŘ OKD (jmenovitě ss. Hegnera, prom. psych. a Znebejánka, prom. soc. ). Tito pracovníci navrhli z existujících psychodiagnostických pomůcek vhodnou testovou baterii. Tomu ovšem předcházelo "pozorování programátorů při práci" pro poznání charakteru práce programátora a specifikaci požadovaných vlastností a schopností, čímž si de facto ověřovali údaje z literatury. Navržená testová baterie byla úspěšně ověřena na vzorku našich programátorů, jejichž jsou dobře známy.

Testová baterie obsahuje řadu testů, jejichž provedení je časově náročné a vyžaduje přítomnost psychologa, jenž testy zadává a hodnotí. Testy sítávají v držení Oborového psychologicko-sociologického střediska.

Možnost testování uchazečů ve zmíněném středisku zlepšuje dosavadní způsob výběru. Navržený zlepšený způsob je tento:

- a) Provedení testu ICL,
- b) Pokud výsledky testu ICL jsou v pásmu neurčitosti, jsou ve spolupráci s odborným psychologem provedeny následující psychodiagnostické pomůcky:
  - bl) Test struktury inteligence (trvá se cca 90 minut) -  
-Očekává se zde jako relevantní nadprůměrný výkon zejména v subtestech GE (schopnost abstrakce, zobecňování, tvoření pojmů), AR (praktické početní myšlení), AN (kombinační schopnost, chápání vztahů, důslednost myšle-

ni), NU (teoretické početní myšlení, induktivní myšlení s čísly), IN (tvoření úsudku, konkrétní praktické myšlení),

b2) Šestnáctifaktorový dotazník Cattell 16 PF (doba sejmání 60 minut) - Žádoucí je vyšší skóre ve faktoru C (emocionální stabilita), G (svědomitost, síla superega), N (sofistikovanost), Q<sub>2</sub> (soběstačnost), nízké skóre ve faktorech O (náchylnost k pocitu viny), Q<sub>4</sub> (dráždivost, napjatost),

b3) Bourdonův škrtačí test, čtverečková forma - zejména nadprůměrný výkon v koncentrované pozornosti a v odolnosti proti psychické únavě (zkouška trvá cca 20 minut),

c) Zhodnocení testů a ve spolupráci s oddělením pro kádrovou a personální práci a návrh na přijetí či odmítnutí.

## 2. Závuk programátorů

Postup závuku a jeho způsob vychází ze skutečnosti, že noví pracovníci přicházejí jednotlivě nebo nanejvýš ve dvojicích či trojicích současně a že tyto pracovníci dosud s výpočetní technikou nepřišli do styku a když ano, tak nikdy neprogramovali. (Pokud byl přijat pracovník s jistými zkušenostmi, pak se jeho závuk redukoval podle úrovně jeho znalostí a zkušeností.)

### 2.1 Dřívější postup závuku

Do roku 1973 byl závuk začínajících programátorů praktikován tak, že nový pracovník po nástupu byl přímo přidělen do pracovního týmu, u něhož se výhledově jevila potřeba tým posílit. Vlastní závuk si vedoucí týmu řídil sám, vycházel přitom ze svých představ a menších či větších zkušeností v této činnosti. V tomto závuku byl nový pracovník veden různou intenzitou podle pracovního zatížení vedoucího týmu.

Tím, že každý vedoucí týmu měl své představy, jak se má programovat, jaké zásady je třeba dodržovat, docházelo za tohoto stavu k nejednotnosti ve výchově nových programátorů a zesilovala se izolovanost jednotlivých pracovních týmů.

Postupné narůstání úkolů v týmech způsobilo, že vedoucí týmu se bránili přijímat do svých řad nové začínající programátory, protože to ve skutečnosti znamenalo zeslabení výkonnosti týmu po dobu 3 až 5 měsíců.

Tyto problémy a nedostatky popsaného způsobu zácviku a nezbytnost důsledného uplatňování normalizovaného programování a všech platných standardů jednotným způsobem vedly ke změně způsobu zácviku nových programátorů, jež se uskutečnila od roku 1973.

## 2.2 Současná forma zácviku nových programátorů

Zácvik řídí a provádí jediný, k tomu účelu určený programátor - školitel. Ten po dobu 3 měsíců, což je obvyklá doba zácviku začínajících programátorů, řídí a vede zácvik, během něhož se má nový pracovník seznámit s programováním natolik, že při zařazení do pracovního týmu má mít takové znalosti a zkušenosti, že je schopen napsat v jednom vyšším programovacím jazyce samostatně jednoduchý program.

Zácvik má převážně ráz samostudia, které je denně doplňováno a uměřováno školitelem v délce 1 až 2 hodin. Nový pracovník před započatím zácviku obdrží osnovu látky pro zaškolení, základní literaturu v češtině a otázky, na něž by měl školený pracovník po skončení zácviku znát odpovědi. V době zácviku každý školený pracovník sestaví sám (nebo s pomocí školitele) tři jednoduché programy, které obsahují všechny základní výroky jazyka, který se učí. Logicky nejsou tyto programy obtížné.

Při sestavování vývojového diagramu je dbáno na přísné dodržování zásad normalizovaného programování a při psaní programů jsou školení pracovníci vedeni k důslednému uplatňování platných standardů pro programování.

## 2.3 Postup a obsah zácviku

Nejdříve se pracovník po dobu 2 týdnů seznamuje se základy programování. Seznámí se s číselnými soustavami, získá hlavní informace o samočinném počítači, na příkladech se naučí používat značek pro vývojové diagramy, je seznámen s metodou normalizovaného programování a s významem hlavních pojmů užívaných v programování.

Potom předvede školitel sám sestavení jednoduchého programu, aby již v počátcích měl školený pracovník hrubou představu o způsobu sestavování programu a významu výroků jazyka.

Následuje výklad hlavních výroků školeného jazyka (v současné době je to ANS Cobol), a to v tom rozsahu, aby nový pracovník mohl se pokusit napsat sám jednoduchý program, zadáný školitelem. Po napsání prvního programu je programovací jazyk probrán znova, ale podrobněji.

Teprve v poslední části zácviku (třetí měsíc) vypracovává školený pracovník podle zadání další dva programy. Pracuje již samostatněji, má však možnost denních konzultací se školitelem. Ten pomáhá při ladění programů, seznamuje nového pracovníka s nástroji pro ladění a záležitostmi programování. V období mezi zkušebními zpracováními programů jsou probírána zbylá témata: dílový jazyk, knihovny, způsob provádění oprav programů, třídění a jiné.

Na závěr zácviku jsou opakována témata podle kontrolních otázek.

Popsaný způsob zácviku nebyl zatím v praxi realizován, protože byl v této podobě navržen koncem minulého roku, a od té doby nebyl žádný nový pracovník do programování přijat. Máme však zkušenosti s formou zácviku, která se od zde popsané liší pouze ve způsobu osvojování programovacího jazyka, které je nyní, jak se domnívám, zdokonaleno, a také v tom, že ještě nebyly vypracovány kontrolní otázky.

Domnívám se, že i dřívější forma zácviku byla úspěšná.



Většina pracovníků po 3 měsících dosáhla požadované úrovně znalostí. Pokud došlo v ojedinělých případech k neúspěchu, pak jej přičítám spíše nedokonalosti ve výběru než ve způsobu zácvičku.