

GRAFICKÉ MODELÝ DYNAMICKÝCH ÚDAJOVÝCH ŠTRUKTÚR A ICH VÝZNAM VO VYUČOVANÍ PROGRAMOVANIA

STOFFOVÁ Veronika – VÉGH Ladislav, SR

Abstrakt

Článok prináša opis grafických modelov štandardných (štandardizovaných) dynamických údajových štruktúr, ktoré umožňujú vytvoriť si správnu predstavu o ich implementácii v pamäti počítača. Správna predstava o implementácii dynamickej premennej a o vytváraní dynamických údajových štruktúr je úzko spojená s predstavou o dynamickom pridelení pamäte počítača. Správna predstava o štruktúre a výstavbe dynamických údajových štruktúr umožňuje pochopiť pravidlá o prístupe k ich jednotlivým elementom, možnosti realizácie základných operácií a funkcií, ktoré s nimi súvisia. Modely sú založené na lineárnej štruktúre pamäte (teda pamäť je chápaná ako usporiadaná množina adresovateľných jednotiek – bajtov, alebo zoskupenie bajtov). Súčasťou práce je aj opis možností využívania didaktickej aplikácie, ktorá podporuje získanie správnej predstavy o implementácii dynamických údajových štruktúr.

Kľúčové slová: dynamické údajové štruktúry, vyučovanie programovania

VISUAL MODELS OF DYNAMIC DATA STRUCTURES AND THEIR IMPORTANCE IN TEACHING PROGRAMMING

Abstract

The article deals with the description of visual models of standard dynamic data structures, which allows creating the right idea of their implementation in the memory of computers. The right idea of creating dynamic variables and dynamic data structures is closely related to the notion of dynamic allocation of memory. The right understanding of structure and construction of dynamic data structures allows comprehending rules of accessing elements in structures, possibilities of implementing basic operations and functions related to structures. The models are based on the linear structure of memory (i. e. memory is viewed as an ordered set of addressable units – bytes or the units of more bytes). In the last part of this article is described a didactic application which helps understanding basic operations on single lined dynamic data structures.

Key words: dynamic variable, dynamic data structures, teaching programming

1. Úvod

Dynamické údajové štruktúry sú jedným z náročnejších tematických okruhov z programovania pre študentov – budúcich učiteľov informatiky. K ich pochopeniu je potrebná určitá abstrakcia, predstava implementácie týchto štruktúr v pamäti počítača. K ľahšiemu pochopeniu učiva je možné využiť grafické modely týchto štruktúr.

2. Dynamické údajové štruktúry

Dynamické údajové štruktúry sa vytvárajú počas behu programu. Na rozdiel od jedno- a dvojrozmerných polí, nemusíme ich veľkosť dopredu stanoviť pri deklarácii premenných. Pri dynamických údajových štruktúrach sa alokácia pamäte počítača vykoná počas behu programu, vždy keď je potrebné na základe algoritmu riešenia problému pridať nový element

k štruktúre. Jednotlivé elementy sú spojené pomocou ukazovateľov. Podľa toho, ako sú jednotlivé elementy spojené, rozlišujeme niekoľko štandardizovaných dynamických údajových štruktúr, napr. zásobník (LIFO), rad (FIFO), obojsmerný zoznam, cyklický zoznam a binárny strom.

2.1 Zásobník – LIFO

Študenti pri oboznamovaní sa s dynamickými údajovými štruktúrami ako prvý vytvoria zásobník – LIFO (last in, first out – posledný dnu, prvý von). Zásobník je štandardizovaná dynamická údajová štruktúra, ktorá sa vytvára ukladaním prvkov „nad sebou“ tak, že priamy prístup je len k posledne uloženému prvku. K ostatným nižšie (predtým) uloženým prvkom sa možno dostať len pomocou adresy, ktorá je uložená v adresnej časti ďalšieho prvku. Teda k prvku sa dostávame v opačnom poradí než boli uložené. Kapacita zásobníka je teoreticky neobmedzená.

Pred vytvorením zásobníka je potrebné predovšetkým definovať jeden element štruktúry. V programovacom jazyku Pascal je to vlastne typ **record**, definovaný nasledovným kódom:

```
type PElement = ^TElement; {ukazovateľ na TElement}
      TElement = record { element štruktúry }
                    {...údaje môžu byť informácie/premenné štandardného typu,
príp. predtým definovaného typu, napr. „meno, priezvisko: string;“ ...}
                    nasl: PElement; {ukazovateľ na
predchádzajúci/nasledujúci element }
                    end;
```

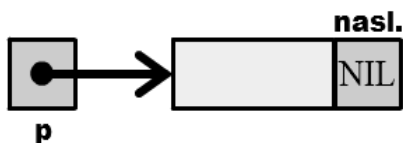
Ako nasledujúci krok je potrebné deklarovať premenné (ukazovatele). Pri tvorbe zásobníka vystačíme s dvoma ukazovateľmi:

```
var vrchol, p: PElement; {vrchol bude ukazovať na posledne uložený vrchný element
a p bude slúžiť na vytvorenie nového elementu}
```

V programe nový element vytvoríme pomocou príkazu **new**. Nový element vždy vložíme za predchádzajúci už existujúci element, na ktorý ukazuje **vrchol**. Obsah ukazovateľa **vrchol** aktualizujeme tak, aby ukazoval na nový element, ktorý sa tak stane novým vrcholom zásobníka:

```
new(p); p^.nasl := vrchol; vrchol := p;
```

Je dôležité, aby si študenti vedeli predstaviť, čo sa odohrá v pamäti počítača vykonaním príkazu „new(p);“



Obr. 1: Rezervovanie miesta v OP pre nový element podľa definície

Keď vyberáme elementy zo zásobníka pri údajovej štruktúre LIFO, naposledy pridaný element vždy odstránime ako prvý:

```
p := vrchol; vrchol := vrchol^.nasl; dispose(p);
```

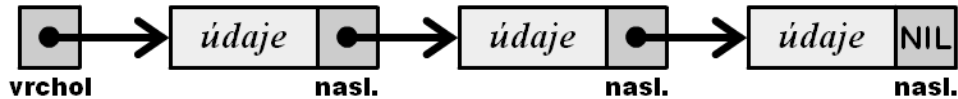
Po vytvorení zásobníka sa študenti oboznámia s ďalšími operáciami na vytvorenom jednosmernom lineárnom zozname, ako napr. vypísanie údajov z informačnej časti všetkých elementov:

```
p := vrchol;
while (p<>nil) do begin
```

... výpis údajov elementu, na ktorý ukazuje „p“ ...

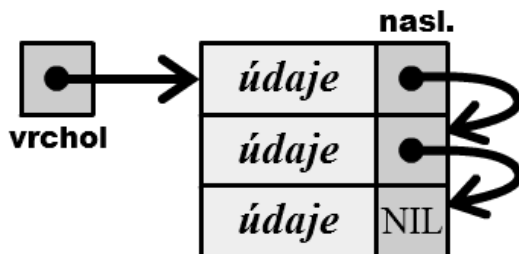
```
p := p^.nasl;  
end;
```

Po tomto kroku už študenti väčšinou sami zvládnu tvorbu ďalších funkcií, ako napr. spočítanie počtu elementov v štruktúre, vyhľadanie určitého elementu, zistenie či je zásobník prázdny, vypočítanie súčtu alebo priemeru z určitých údajov elementov atď.



Obr. 2: Grafický model zásobníka pomocou dynamickej údajovej štruktúry

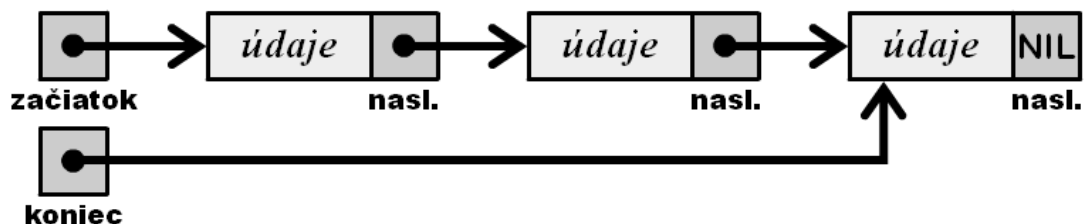
Pri tvorbe zásobníka a pri vykonávaní rôznych operácií v jednosmernom lineárnom zozname edukanti používajú didaktickú pomôcku opísanú na konci tohto článku. Vďaka takejto pomôcke študenti vedia ľahšie nastaviť ukazovatele na jednotlivé elementy a pochopiť tvorbu dynamických údajových štruktúr a spracovanie v nich uložených údajov.



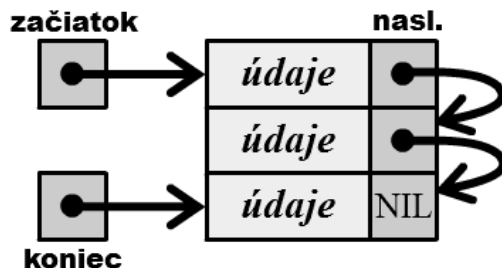
Obr. 3: Iný grafický model zásobníka

2.2 Rad – FIFO

Pri tvorbe radu pomocou dynamických premenných študenti si musia uvedomovať, že nový element musia vždy pridať na koniec lineárneho zoznamu, ale odobrať element je potrebné vždy zo začiatku radu. Kvôli tomu je nutné deklarovať o jeden ukazovateľ viac. Po pochopení tvorby zásobníka pomocou dynamických premenných študentom už väčšinou nerobí problém prepis niektorých funkcií tak, aby lineárny zoznam fungoval ako rad. Pri stavbe radu používame rovnaké stavebné elementy ako v prípade zásobníka.



Obr. 4: Model radu pomocou dynamickej údajovej štruktúry



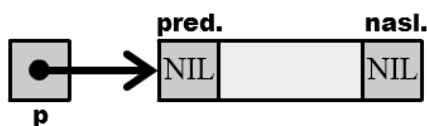
Obr. 5: Iný grafický model radu

Aj grafický model na obr. 5 môže poslúžiť ako určitá predstava, umožňujúca realizovať jednotlivé funkcie, ktoré sa viažu na štandardizovanú údajovú štruktúru rad: zaradiť nové element do radu na koniec radu, vybrať element zo začiatku radu, spočítať elementy v rade, určiť poradové číslo určitého elementu atď.

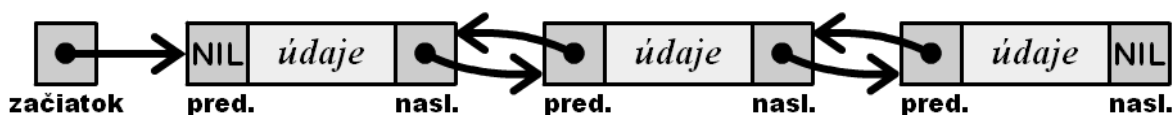
2.3 Obojsmerné zoznamy

Ďalším krokom pri osvojení vedomostí o tvorbe dynamických údajových štruktúr môže byť tvorba obojsmerných zoznamov. Pri týchto údajových štruktúrach je do každého elementu pridaný ďalší ukazovateľ, ktorý vždy obsahuje adresu predchádzajúceho elementu. Keďže je to len menšia zmena pridaná do jednosmerných zoznamov, edukanti s menšou pomocou zvládnu pridanie nového ukazovateľa do jednosmerných dynamických štruktúr a tak tvorbu zásobníka a radu pomocou obojsmerných lineárnych zoznamov.

Treba si uvedomiť čo vyvolá pridanie ďalšieho ukazovateľa do definície dynamického elementu. Môžeme si pomôcť grafickým modelom pre: **new(p)**; v tomto prípade (pozri obr. 6).

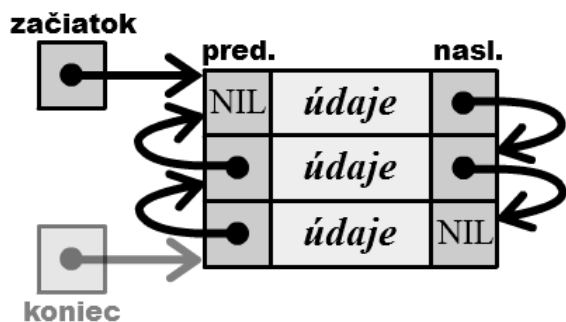


Obr. 6: Rezervovanie miesta v OP pre nový element obojsmerného zoznamu



Obr. 7: Model obojsmerného lineárneho zoznamu

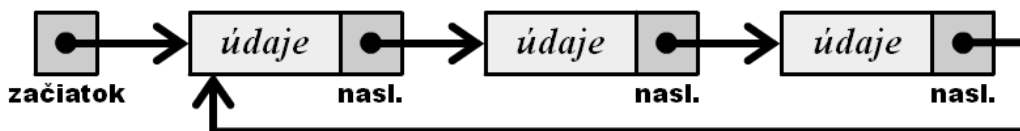
Keďže po tomto kroku študenti už získajú potrebné zručnosti práce s dynamickými údajovými štruktúrami, môžu dostať aj zložitejšie úlohy, ako napr. tvorbu usporiadaných zoznamov, t. j. lineárnych zoznamov pri ktorých elementy sú usporiadané. Problémom tu môže byť správne nájdenie miesta nového elementu a nastavenie potrebných ukazovateľov. Grafický model umožňuje vykonštruovať v akom poradí a akým spôsobom je potrebné aktualizovať jednotlivé ukazovatele v prípade zaradenia nového elementu a odvodiť postup riešenia. Grafický model pomáha napr. aj pri hľadaní elementu na vynechanie podľa hodnoty údaj, čo môže byť súčasťou riešenia určitej úlohy, a ukazuje ako preorientovať ukazovatele, aby dynamicky zoznam bol správne zredukovaný. Keďže podľa grafického modelu vidieť, že obojsmerný lineárny zoznam možno prechádzať v oboch smeroch, je účelné doplniť ďalší „vonkajší“ ukazovateľ na koniec zoznamu.



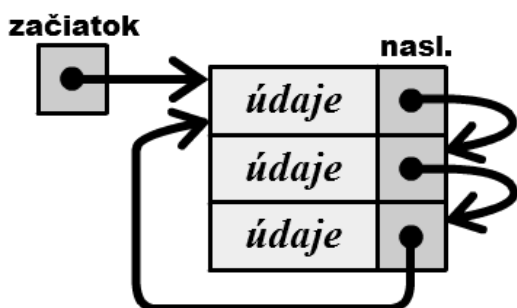
Obr. 8: Iný grafický model obojsmerného lineárneho zoznamu, ktorý je doplnený ukazovateľom na koniec

2.4 Cyklické zoznamy

Ďalšou zaujímavou úlohou môže byť tvorba cyklických zoznamov (tak jednosmerných, ako aj obojsmerných), pri ktorých nasledovníkom posledného elementu je vždy prvý element. Je to len menšia zmena pridaná do lineárnych zoznamov, ktorej implementáciu študenti zvládnu bez problémov.



Obr. 9: Model jednosmerného cyklického zoznamu



Obr. 8: Iný grafický model
jednosmerného cyklického zoznamu

2.5 Binárne stromy

Realizácia binárnych stromov pomocou dynamických údajových štruktúr môže byť zložitejšou úlohou. Pri realizácii môže každý element obsahovať dva ukazovatele, jeden na ľavého a jeden na pravého následníka, alebo aj tretí ukazovateľ na predka. V prvom prípade je možné binárny strom prechádzať len od koreňa smerom dolu, v druhom prípade v oboch smeroch. Binárny strom môže byť vhodnou údajovou štruktúrou pre mnohé úlohy, kde vzťahy medzi prvkami možno analogicky vyjadriť ako binárny strom. Napr. môže vyjadriť aj vzťah medzi dieťaťom a jeho rodičmi.

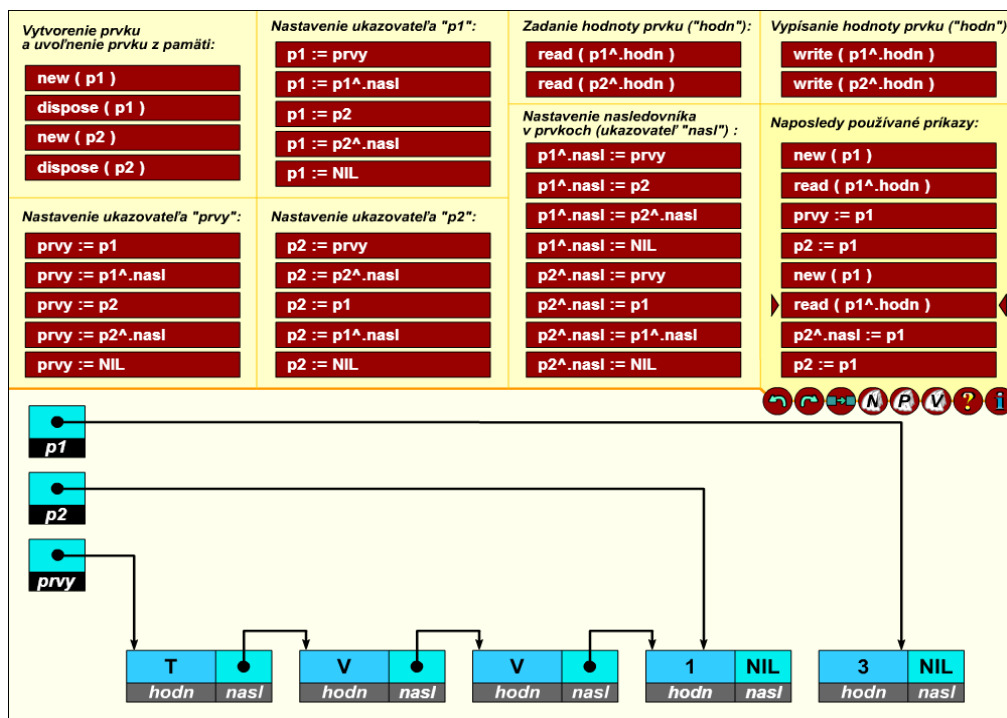
3. Didaktická aplikácia – Dynamické údajové štruktúry

Keďže pochopenie jednosmerných lineárnych zoznamov je kľúčovou otázkou k pochopeniu práce s dynamickými údajovými štruktúrami, pre študentov sme vytvorili didaktickú pomôcku. Didaktická aplikácia okrem demonštrovania základných dynamických lineárnych údajových štruktúr obsahuje interaktívny editor na ich tvorbu. V hornej časti aplikácie používateľ si môžu vybrať rôzne príkazy, ktoré sú zaradené do ôsmich skupín. K dispozícii sú 3 vonkajšie ukazovatele a dynamický element je implementovaný ako v prípade zásobníka v časti 2.1. Informačná časť elementu môže obsahovať reťazec znakov. Používateľ v dolnej časti okna, po kliknutí na určitý príkaz hneď vidí jeho účinok, výsledok jeho realizácie, ktorý sa prejaví zmenou v grafickom modeli. Všetky zmeny, ktoré sa odohrajú interaktívnou voľbou z príkazovej ponuky, sa zobrazia postupne na „pracovnej ploche“. Tak môže používateľ experimentovať pridávaním a odoberaním prvkov a určovaním hodnoty adresnej časti prepájať ich do želanej štruktúry.

Záver

Podľa našich skúseností študenti ľahšie pochopia prácu s dynamickými štruktúrami ak majú k dispozícii grafický model, ktorý využijú na znázornenie postupu riešenia. Grafický model môže použiť na myšlienkovú implementáciu ľubovoľnej dynamickej neštandardnej

údajovej štruktúry, ktorý zjednoduší riešenie problému. Fungovanie ukazovateľa a dynamickej premennej – explicitne definovaného stavebného elementu údajovej štruktúry možno naštudovať pomocou reprezentovanej didaktickej aplikácie. Používateľ – študent môžu experimentovaním na grafickom modeli počas tvorby programov pochopiť aj zložitejšie operácie vykonané nad dynamickými údajovými štruktúrami.



Obr. 10: Didaktická aplikácia na tvorbu jednosmerných zoznamov

Literatúra

- 1 STOFFA, V.: *Az elektronikus tanulás lehetőségei a programozás elsajátításában*. In: *AGRIA MEDIA 2002 : Az elektronikus tanulás a III. évezred pedagógiai kihívása*. Eger, 2003, s. 480 - 488. ISBN 963-9417-09-2.
- 2 STOFFOVÁ, V. – VÉGH, L.: Szemléltető animációk a programozásban. In: *INFODIDACT 2010, 3. Informatika Szakmódszertani Konferencia*. Szombathely, 2010.
- 3 SIK LÁNYI, C. – GEISZT, Z. – KÁROLYI, P. – TILINGER, Á. – MAGYAR, V.: Virtual Reality in special needs early education, *The International Journal of Virtual Reality*, 2006, 5(4): 55-68. ISSN1081-1451: <http://www.ijvr.org/issues/issue4/7.pdf>
- 4 SCHANDA, J. – SIK LÁNYI, C.: "Colour Correct Digital Museum on the Internet, Virtual Reality". *Proc. HCII 2007, Lecture Notes in Computer Science, LNCS 4563*, 2007, pp. 708-717.

Lektoroval: **Prof. Dr. Ing. Imrich Okenka, PhD.**

Kontaktné adresy:

Veronika Stoffová, Prof. Ing. CSc., tel. +421 35 3260 605, e-mail: NikaStoffova@seznam.cz

Ladislav Végh, PaedDr., tel. +421 35 3260 760, e-mail: veghl@selyeuni.sk

Univerzita J. Selyeho, Ekonomická fakulta, Katedra matematiky a informatiky,
Bratislavská cesta 3322, 945 01 Komárno, Slovenská republika