

Wstęp do programowania

Wskaźniki

Pamięć dynamiczna

- W czasie kompilacji wielkość wszystkich danych musi być znana, aby kompilator mógł ustalić konkretne adresy i zoptymalizować kod.
- Zapewne denerwuje nas często niemożność zaalokowania dokładnie takiej pamięci, jakiej potrzebujemy dla konkretnych danych – wydaje się, że jedyną co nam pozostaje, to zaalokować z zapasem (np. tablicę) i w razie czego używać tylko jej początkowego fragmentu
- To jest dość sztywne rozwiązanie, nie zdające egzaminu np. w przypadku grafów.

Alokowanie pamięci

- Nowoczesne języki programowania umożliwiają alokację pamięci w trakcie działania programu.
- Zaalokowany fragment o ustalonej i wyliczanej w czasie działania programu wielkości zajmuje fragment pamięci zarządzany przez kopiec systemowy, a jego adres nadawany jest *zmiennej wskaźnikowej*.
- Wielkość zmiennej wskaźnikowej odpowiada liczbie bajtów potrzebnych do określenia adresu (typowo 4).

Zmienne wskaźnikowe

- Zmienna wskaźnikowa powinna mieć określony typ, który adresuje.

```
type wskaznikR = ^Real;  
      wskaznikI = ^Integer;  
  
var wr : wskaznikR;  
      wi : wskaznikI;  
  
begin  
    New(wr); New(wi);  
    wr^:=3.14; wi^:=trunc(wr^);  
  
    ...  
end
```

Procedury alokacji i dealokacji

- Mamy parę procedur (czyli dwie), za pomocą których alokujemy i dealokujemy pamięć dynamiczną:
 - `New(p)` – alokuje stosowny, (zależny od wskazywanego typu) obszar wolnej pamięci, rezerwuje go oraz nadaje zmiennej `p` adres tego obszaru
 - `Dispose(p)` – zwalnia zaalokowany i zajmowany przez zmienną `p` obszar. Wartość `p` po dealokacji jest nieokreślona! (choć w praktyce najczęściej pozostaje niezmienniona :).

Nil

- Każdy typ wskaźnikowy ma uniwersalną stałą Nil, która oznacza, że zmienna nie wskazuje na nic.
- Nie wolno odwoływać się do zawartości zmiennej, która ma wartość Nil! (NULL POINTER ASSIGNMENT ERROR)\

Przykłady

- Przepisanie wartości na zmienne wskaźnikowe

```
var pa, pb: ^Integer;
```

```
...
```

```
New (pa) ; New (pb) ;
```

```
pa^ := 4; pb := pa; Write (pb^) ;
```

```
pb^ := pa^-1;
```

```
Write (pb^, pa^) ;
```

```
Dispose (pb) ;
```

- W wyniku wykonania tego kodu zostanie wypisana trójka 4,3,3, a dodatkowo stracimy bezpowrotnie zaalokowany obszar dla zmiennej pb.

Przykłady

- Przepisanie wartości na zmienne wskaźnikowe

```
var pa, pb: ^Integer;
```

```
...
```

```
New (pa) ; New (pb) ;
```

```
pa^ := 4 ; pb^ := pa^ ; Write (pb^) ;
```

```
pb^ := pa^ - 1 ;
```

```
Write (pb^, pa^) ;
```

```
Dispose (pb) ;
```

- W wyniku wykonania tego kodu zostanie wypisana trójka 4,3,4, a obszar zaalokowany dla zmiennej pb zostanie prawidłowo zwolniony. .

Dealokacja w Javie

- W niektórych językach programowania (Java) nie musimy martwić się o dealokację: system sam rozpozna, czy zaalokowany obszar jest wskazywany przez jakąkolwiek zmienną i w przypadku gdy nie jest, zwolni go. Czynność tę wykonują *odśmieczacze (garbage collectors)*.
- W Pascalu jednak tak nie jest i musimy sami dbać o to, żeby nie pozostawić śmieci w systemie.
- Pamiętajmy: nie każda zadeklarowana zmienna wskaźnikowa musi mieć New! Może się przecież podpiąć do innej uprzednio zaalokowanej zmiennej.

Wskaźniki jako indeksy

- Można też myśleć o wskaźnikach, jako indeksach tablicy. W pewnym sensie pamięć operacyjna też może być wyobrażona jako jedna wielka tablica.
- Wtedy możemy sami aranżować alokację i dealokację w ramach zadeklarowanego uprzednio obszaru pamięci w tablicy.
- Wygodnie też jest używać wskaźników przy sortowaniu dużych struktur. Nie musimy bowiem przestawiać dużych elementów tablicy, lecz wystarczy pozamieniać wskaźniki: zaczynamy od permutacji identycznościowej i odwołujemy się do elementów tablicy przez osobną tablicę wskaźników.

Sortowanie na wskaźnikach

- Zatem fragmenty programu sortującego zamiast tak:

```
if A[i]>A[j] then
  begin
    pom:=A[i];
    A[i]:=A[j];
    A[j]:=pom
  end
```

Sortowanie na wskaźnikach

- ...mogą wyglądać tak:

```
if A[w[i]]>A[w[j]] then
  begin
    pom:=w[i];
    w[i]:=w[j];
    w[j]:=pom
  end
{A[w[i]]<=A[w[j]]}
```

- W ten sposób w tablicy w przechowujemy informację o aktualnej permutacji (zainicjalizowanej na identyczność).

Wskaźniki jako wyniki funkcji

- Wiemy, że typy wyników funkcji ograniczają się do typów prostych.
- Na szczęście wynikiem funkcji może być wskaźnik.
- Zatem korzystając ze wskaźników możemy sprawić, że wartość dowolnego typu stanie się wynikiem funkcji. Wystarczy przekazać ją jako wskaźnik.

Wskaźniki jako wyniki funkcji

```
type tab=^array[1..n]of Integer;
wsktab=^tab;
function SkopiujDane (A:tab):wsktab;
  {alokuje pamięć na n elementów i kopiuje
   zawartość tablicy A do niej, po czym
   przekazuje ją jako wynik}      ...
  var pA:wsktab; i:Integer;
begin
  New (pA); {Uwaga: New (SkopiujDane) źle!}
  pA^:=A;
  SkopiujDane:=pA
end;
```