

ZSI.Techniki Programowania

Wykład 4 Janusz Jabłonowski

Typy wskaźnikowe i zmienne dynamiczne

- Motywacja: dynamiczne struktury danych.
- Wskaźniki są abstrakcją adresów obiektów określonego typu w pamięci komputera.
- Typ wskaźnikowy: typ, którego elementy są wskaźnikami do zmiennych pewnego zadanego typu bazowego (najczęściej rekordowego).
- Jest wyróżniona stała **nil**, oznaczająca pusty wskaźnik (czyli na nic nie pokazujący).
- **nil** należy do każdego typu wskaźnikowego.
- Deklarowanie typu wskaźnikowego

^<Identyfikator typu bazowego>

Uwaga: typ bazowy może być zadeklarowany po typie wskaźnikowym.

- Inicjalizowanie
 - ⇒ Zmienne wskaźnikowe (jak wszystkie zmienne w Pascalu) **nie** są automatycznie inicjalizowane - trzeba to robić samemu.
 - ⇒ Są tylko trzy możliwości zainicjalizowania zmiennej wskaźnikowej p:
 - `p := nil`
Po tej operacji p nie wskazuje na żaden obiekt
 - `p := q`
Po tej operacji p i q wskazują na ten sam obiekt (q zmienna tego samego typu co p).
UWAGA: q powinno być uprzednio zainicjalizowane - inaczej p dalej nie będzie zainicjalizowane.
Ogólna postać (obejmująca oba te przypadki): `p := <wyrażenie typu wskaźnikowego>`.
 - `New (p)`
Przydziela fragment pamięci mieszczący obiekt typu bazowego i przypisuje jego adres na p.
- Porównywanie
 - ⇒ `p = q`
 - ⇒ `p <> q`
 - p i q muszą być wskaźnikami tego samego typu. W miejsce p i/lub q można wstawić **nil** lub dowolne inne wyrażenie, którego wartością jest wskaźnik odpowiedniego typu.
- Dostęp do wskazywanego obiektu:
`p^`
p musi być zainicjalizowane i różne od **nil**.
- Zwalnianie uprzednio przydzielonej pamięci:
`procedure dispose (p)`
p musi być zainicjalizowane i różne od **nil**. Uwaga: wartość p po wykonaniu dispose nie jest określona przez opis języka (czyli z punktu widzenia programisty jest zmienną niezainicjowaną).
- Dodatkowe uwagi:
Wartości funkcji mogą być typu wskaźnikowego.

- Zmienne dynamiczne (czyli to na co wskazują wskaźniki)
 - ⇒ Nie mają nazw,
 - ⇒ Są tworzone operacją new,
 - ⇒ Istnieją aż do końca wykonywania programu, lub do wykonania operacji dispose,
 - ⇒ Dostęp do nich jest możliwy jedynie dzięki wskaźnikom.
 - ⇒ Można wykonywać na nich te same operacje, co na zwykłych zmiennych tych samych typów.
- Porównanie czasu istnienia różnych rodzajów zmiennych:
 - ⇒ Zmienne globalne (cały czas działania programu),
 - ⇒ Zmienne lokalne (cały czas działania procedury/funkcji , w której są zadeklarowane, przy każdym wywołaniu procedury/funkcji powstają na nowo),
 - ⇒ Zmienne dynamiczne (od new do dispose lub końca programu).
- Porównanie miejsc przechowywania różnych rodzajów zmiennych:
 - ⇒ Zmienne globalne (pamięć stała),
 - ⇒ Zmienne lokalne (stos),
 - ⇒ Zmienne dynamiczne (sterta).
- Typowa organizacja pamięci programu w Pascalu (pamięć stała, stos, sterta), problem fragmentacji pamięci na stercie.
- Dlaczego **nil** i <niezainicjowane>^ są tak niebezpieczne.
- Rysunki (wskaźnik p wskazujący na zmienną dynamiczną i wskaźnik p o wartości **nil**). Czemu warto je robić a dlaczego bywają niebezpieczne.
- Pułapki związane ze stosowaniem wskaźników:

1. Gubienie pamięci (już w czasie wykonywania programu)

```
var p: ^integer;
begin
  new(p);
  new(p);
end.
```

2. Gubienie pamięci (prawdopodobnie po zakończeniu programu)

```
var p: ^integer;
begin
  new(p);
  <instrukcje bez dispose>
end.
```

3. Sięganie do nieistniejącej zmiennej dynamicznej

```
var p,q: ^integer;
begin
  new(p);
  q := p;
  dispose(p);
  q^ := 7;
end.
```

- Przykłady typów zbudowanych przy użyciu typów wskaźnikowych:

```

const
  n = ...;
type
  TTab = array [1..n] of integer;
  TWskTab = ^TTab;
  TTabTab = array[1..n] of TWskTab

```

- Przykłady wyrażeń korzystających z typów wskaźnikowych:

```

var
  pint: ^integer;
  ptab: TWskTab;
  ptabtab : TTabTab;
begin
  new(pint);           {Utworzenie jednej zmiennej dynamicznej typu całkowitego}
  pint^ := 13;        {Inicjalizacja zmiennej dynamicznej (nie: wskaźnikowej)}
  new(ptab);          {Utworzenie jednej zmiennej dynamicznej będącej tablicą}
  ptab^ [1] := 44;    {Zainicjalizowanie pierwszego elementu tablicy}
  pint^ := ptab^ [1]; {Skopiowanie elementu tablicy do zmiennej całkowitej}
  new(ptabtab [1]);   {Zainicjalizowanie pierwszego elementu tablicy ptabtab i utworzenie}
                    {nowej tablicy typu TwskTab jako zmiennej dynamicznej}
  ptabtab [1]^ [1] := 13; {Zainicjalizowanie pierwszego elementu utworzonej tablicy}
  ...
end.

```

- Niestandardowe operacje na wskaźnikach w niektórych implementacjach Pascala (Free Pascal, Turbo Pascal):
 - typ Pointer,
 - **procedure** GetMem(**var** p: Pointer; size: Word); (w zależności od implementacji typ size to LongInt (FP) lub Word (TP)), **nil** gdy zabraknie pamięci.
 - **procedure** FreeMem(**var** p: Pointer; size: Word); (w zależności od implementacji typ size to LongInt (FP) lub Word (TP))
 - **procedure** Mark(**var** p: Pointer); (we FreePascalu nic nie robi)
 - **procedure** Release(**var** p: Pointer); (we FreePascalu nic nie robi)
 - **function** MemAvail: LongInt;
 - **function** MaxAvail: LongInt;
 - Uwaga: nie wolno mieszać ze sobą pary operacji GetMem/FreeMem z parą Mark/Release.

Ćwiczenia:

- Napisz program, który zawiera tablicę wskaźników do liczb całkowitych i pozwala dopisać na jej koniec nową liczbę oraz usunąć z jej końca ostatnią liczbę. Po każdej operacji program powinien wypisywać informację o liczbie utworzonych elementów i wielkości zajętej pamięci.