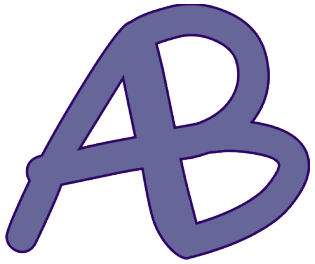


Wykład
PASCAL

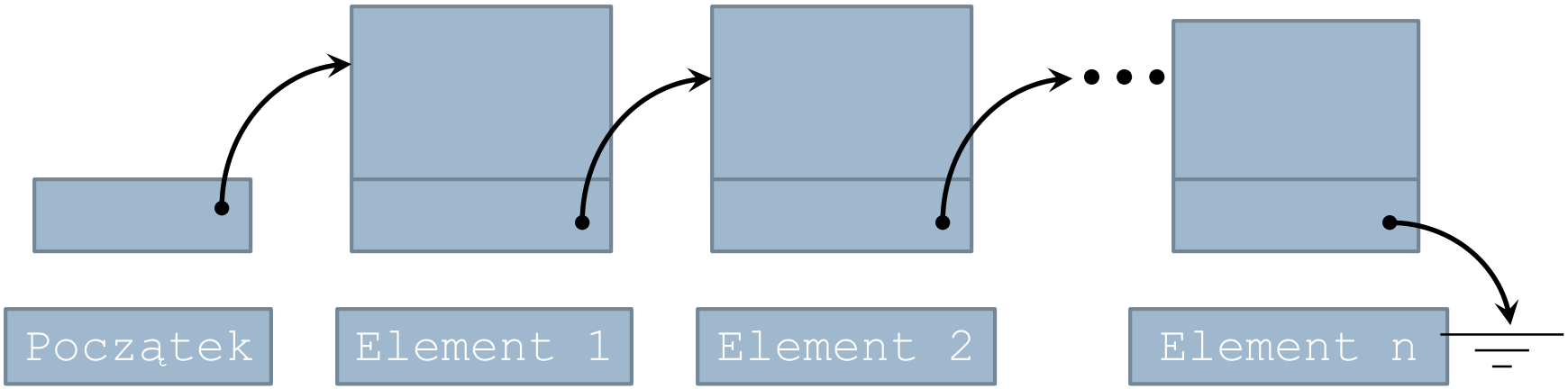
Dynamiczne struktury danych

Listy i drzewa

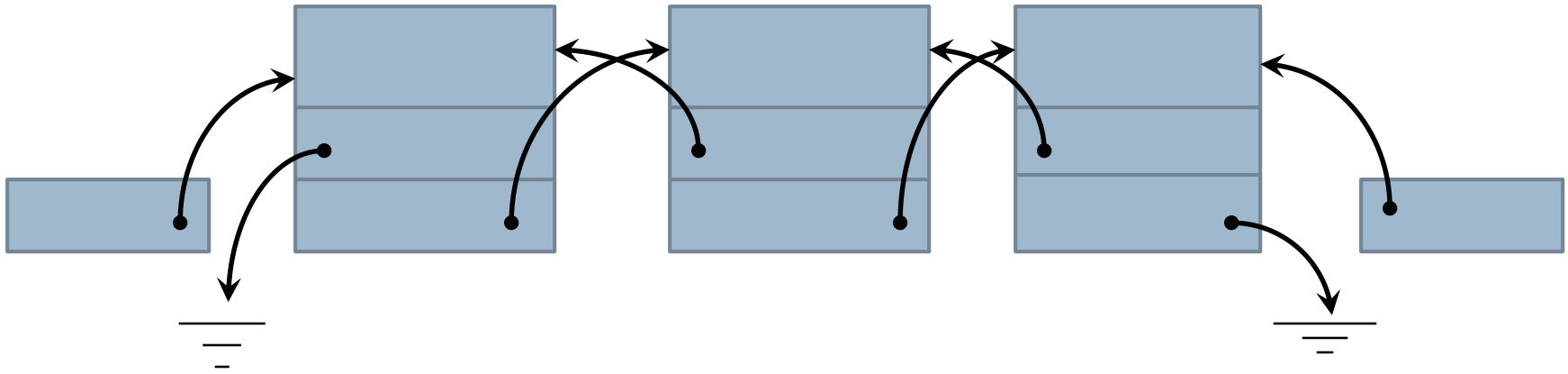
- Listy jednokierunkowe
- Listy jednokierunkowe uporządkowane
- Listy dwukierunkowe
- Listy dwukierunkowe uporządkowane
- Struktury drzewiaste

Listy z wartownikami

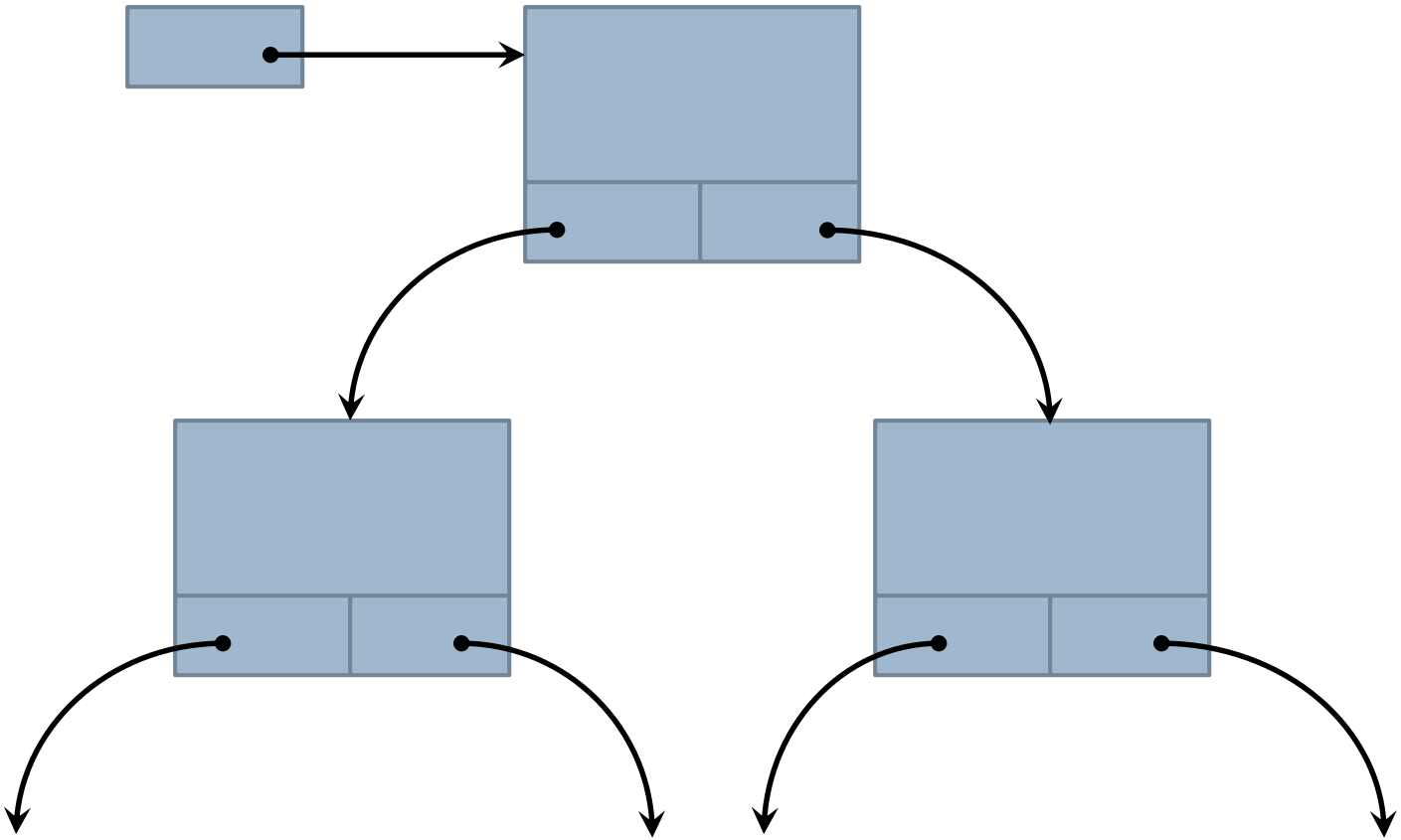
Lista jednokierunkowa



Lista dwukierunkowa

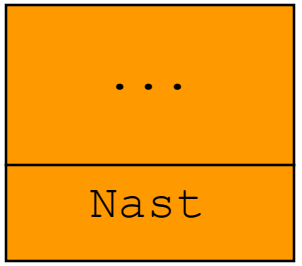


Drzewo binarne



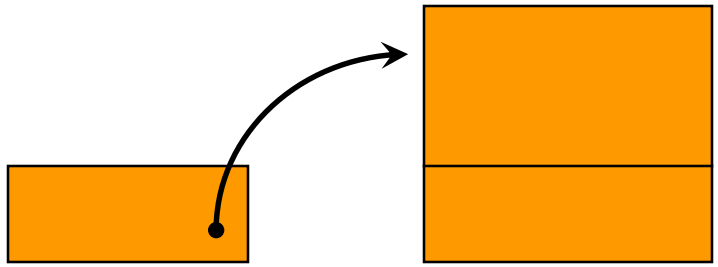
Definiowanie listy

```
type
   PElement = ^TElement;
   TElement = record
      ..............
     Nast: PElement;
   end;
.............
var
   Lista: PElement;
```



Instrukcje **New** i **Dispose**

```
var
   p: PElement;
..............
New(p);
..............
Dispose(p);
```



Wskaźnik
p
typu
PElement

Element
p^
typu
TElement

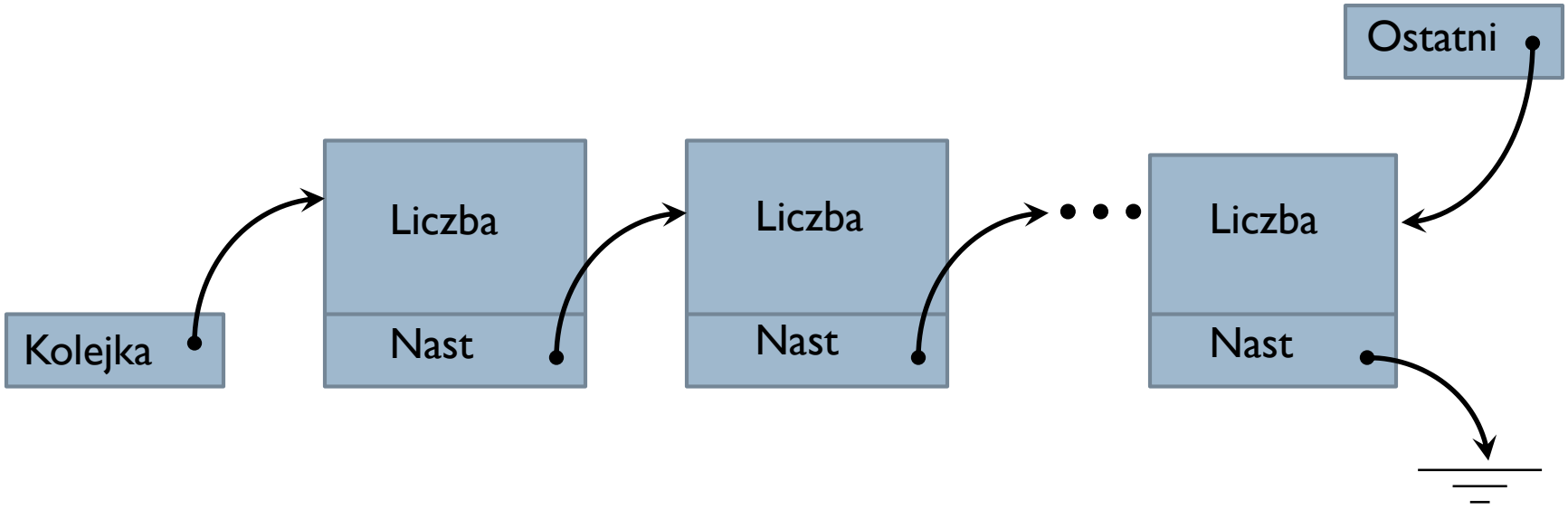
Operacje na elementach listy

- Wstawianie nowego elementu na początek listy
- Wstawianie nowego elementu po danym elemencie listy
- Usunięcie pierwszego elementu listy
- Usunięcie z listy elementu znajdującego się po danym elemencie

Przykład 1 - kolejka

W tym przykładzie za pomocą dynamicznej struktury danych – listy jednokierunkowej realizujemy kolejkę.

Wykorzystany zostanie nico zmodyfikowany schemat listy - lista jednokierunkowa z dodatkowym wskaźnikiem na jej koniec (ostatni element.



Przykład:

Definicja i deklaracja zmiennych:

```
1   program kolejka;
2   uses
3     Crt;
4   type
5     TWskaznik = ^TElement;
6     TElement  = record
7                   Liczba : Integer;
8                   Nast   : TWskaznik;
9                 end;
10  var
11    Kolejka : TWskaznik;
12    Ostatni : TWskaznik;
13    Liczba : Integer;
14    Znak : Char;
15    P : Pointer;
16
```

Przykład:

Procedura ustawia w element w kolejce – na jej koniec

```
17  procedure Ustaw (Liczba: Integer);
18  var
19    E : TWskaznik;
20  begin
21    New (E);
22    E^.Liczba := Liczba;
23    E^.Nast := Nil;
24    if (Kolejka = nil) then
25    begin
26      E^.Nast := Kolejka; Kolejka := E; Ostatni := E;
27    end
28    else
29    begin
30      Ostatni^.Nast := E;
31      Ostatni := E;
32    end;
33  end; {------------------------------ Ustaw -}
34
```

Przykład:

Funkcja zwraca wartość pierwszego elementu z kolejki, jednocześnie go z niej usuwając.

```
35  function Obsluz: Integer;
36  var
37    E : TWskaznik;
38  begin
39    if (Kolejka = nil) then
40      Obsluz := 0
41    else
42    begin
43      E := Kolejka;
44      Obsluz := E^.Liczba;
45      Kolejka := E^.Nast;
46      Dispose (E);
47      if Kolejka = nil then Ostatni := Nil;
48    end;
49  end; {------------------------------------ Obsluz}
```

Przykład:

Procedura wypisuje wszystkie elementy kolejki.

```
51  procedure WypiszKolejke;
52  var
53    E : TWskaznik;
54  begin
55    E := Kolejka;
56    while E <> nil do
57    begin
58      write (E^.Liczba, ' ');
59      E := E^.Nast;
60    end;
61  end; {------------------------------ WypiszKolejke -}
```

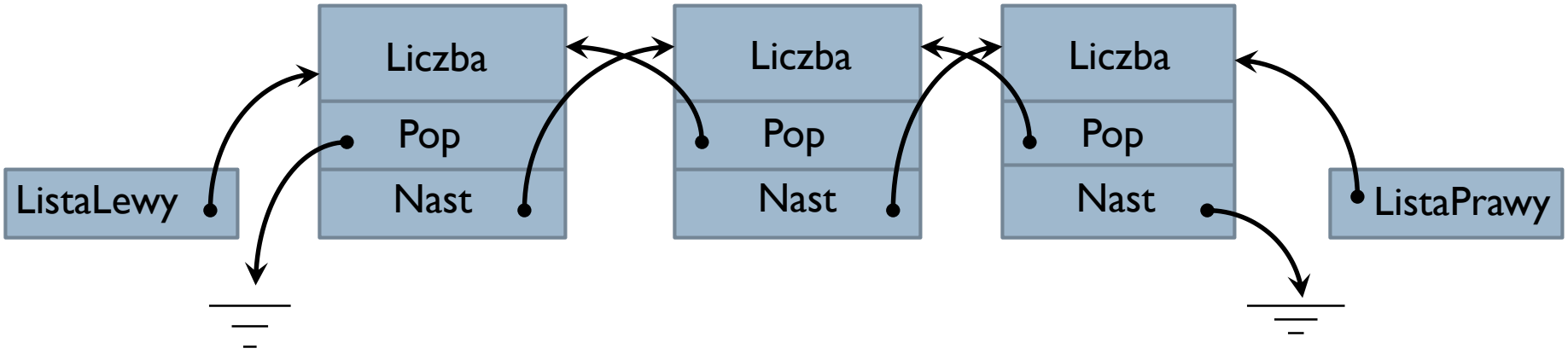
Przykład:

```
63  begin
64    Kolejka := nil;
65    Ostatni := nil;
66    repeat
67      ClrScr;
68      write ('Kolejka: ');
69      WypiszKolejke;
70      writeln;
71      writeln ('U - ustaw w kolejce, O - obsluz, K - koniec');
72      Znak := UpCase(ReadKey);
73      case Znak of
74        'U' : begin
75                write ('Podaj liczbe: ');
76                readln (Liczba);
77                Ustaw (Liczba);
78              end;
79        'O' : begin
80                Liczba := Obsluz;
81                if Liczba = 0 then
82                  writeln ('Nie ma czego obslugiwac, kolejka pusta.')
83                else
84                  writeln ('Element do obsluzenia: ', Liczba);
85              end;
86      end;
87      if ((Znak='U') or (Znak='O')) then
88      begin
89        write ('Nowa kolejka: ');
90        WypiszKolejke;
91        writeln; writeln ('Wcisnij Enter');
92        readln;
93      end;
94    until (Znak = 'K');
95  end.
```

Przykład 2 – lista jednokierunkowa

Program pokazuje podstawowe działania w liście dwukierunkowej:

- dodanie elementu na początku
- na końcu, usuniecie elementu o wskazanej wartości (kluczu)
- wypisanie listy od dowolnej strony.



Przykład 2:

Tworzenie struktury listy.

```
1  program Lista_dwukierunkowa;
2  uses
3    Crt;
4
5  type
6    TWskaznik = ^TElement;
7    TElement  = record
8                  Liczba : Byte;
9                  Pop : TWskaznik;
10                 Nast : TWskaznik;
11               end;
12 var
13   ListaLewy, ListaPrawy : TWskaznik;
14   Liczba : Byte;
15   Znak : Char;
16   P : Pointer;
```

Przykład 2:

Procedura wstawia element z lewej strony listy.

```
17
18  procedure UstawZLewej (Liczba: Byte);
19  var
20    E : TWskaznik;
21  begin
22    New (E);
23    E^.Liczba := Liczba;
24    E^.Pop := Nil;
25    if (ListaLewy = nil) then
26    begin
27      E^.Nast := Nil; ListaLewy := E; ListaPrawy := E;
28    end
29    else
30    begin
31      ListaLewy^.Pop := E; E^.Nast := ListaLewy; ListaLewy := E;
32    end;
33  end; {------------------------------ UstawZLewej -}
```

Przykład 2:

Procedura wstawia element z prawej strony listy.

```
35  procedure UstawZPrawej (Liczba: Byte);
36  var
37    E : TWskaznik;
38  begin
39    New (E);
40    E^.Liczba := Liczba;
41    E^.Nast := Nil;
42    if (ListaPrawy = nil) then
43    begin
44      E^.Pop := Nil; ListaPrawy := E; ListaLewy := E;
45    end
46    else
47    begin
48      ListaPrawy^.Nast := E; E^.Pop := ListaPrawy; ListaPrawy := E;
49    end;
50  end; {------------------------------ UstawZPrawej -}
```

Przykład 2:

```
52  procedure Usun (Liczba : Byte);
53  var
54    E, Nast, Pop : TWskaznik;
55  begin
56    if ListaLewy <> nil then
57    begin
58      E := ListaLewy;
59      while E<>nil do
60        if E^.Liczba = Liczba then
61        begin
62          if E^.Pop = Nil then { pierwszy z lewej }
63            if E^.Nast = Nil then { takze pierwszy z prawej }
64            begin
65              ListaLewy := Nil; ListaPrawy := Nil
66            end
67            else { Pierwszy, ale nie jedyny }
68            begin
69              Nast := E^.Nast; Nast^.Pop := E^.Pop; ListaLewy := Nast;
70            end
71          else { nie jest pierwszy z lewej }
72            if E^.Nast = Nil then { ale pierwszy z prawej }
73            begin
74              Pop := E^.Pop; Pop^.Nast := E^.Nast; ListaPrawy := Pop;
75            end
76            else
77            begin
78              Nast := E^.Nast; Pop := E^.Pop;
79              Pop^.Nast := E^.Nast; Nast^.Pop := E^.Pop;
80            end;
81          Pop := E;
82          E := E^.Nast;
83          Dispose (Pop);
84        end
85        else
86          E := E^.Nast
87    end;
88  end; {------------------------------------------ Usun -}
```

Przykład 2:

Procedura wypisuje wszystkie elementy listy od lewej.

```
90  procedure WypiszListeZLewej;
91  var
92    E : TWskaznik;
93  begin
94    E := ListaLewy;
95    while E <> nil do
96    begin
97      write (E^.Liczba, ' ');
98      E := E^.Nast;
99    end;
100 end; {------------------------------ WypiszListeZLewej -}
```

Przykład 2:

Procedura wypisuje wszystkie elementy listy od prawej.

```
102  procedure WypiszListeZPrawej;
103  var
104    E : TWskaznik;
105  begin
106    E := ListaPrawy;
107    while E <> nil do
108    begin
109      write (E^.Liczba, ' ');
110      E := E^.Pop;
111    end;
112  end; {------------------------------ WypiszListeZPrawej -}
```

```
114  begin
115    ListaLewy := nil;
116    ListaPrawy := nil;
117    repeat
118      ClrScr;
119      write ('Lista: ');
120      WypiszListeZLewej; write (' - od tylu: '); WypiszListeZPrawej;
121      writeln;
122      writeln ('L - ustaw z lewej, P - ustaw z prawej, U - usun, K -
         koniec');
123      Znak := UpCase(ReadKey);
124      if znak in ['U', 'L', 'P'] then
125      begin
126        repeat
127          write ('Podaj liczbe z zakresu (1..255): ');
128          readln (Liczba);
129        until Liczba>0;
130        if Znak = 'L' then
131          UstawZLewej (Liczba)
132        else if Znak = 'P' then
133          UstawZPrawej (Liczba)
134        else
135          Usun (Liczba);
136      end;
137      if ((Znak='L') or (Znak='P')) then
138      begin
139        write ('Nowa kolejka: ');
140        WypiszListeZLewej; write (' - od tylu: '); WypiszListeZPrawej;
141        writeln; writeln ('Wcisnij Enter');
142        readln;
143      end;
144    until (Znak = 'K');
145  end.
```

Literatura:

W prezentacji wykorzystano przykłady i fragmenty:

- R. Jarża, *Turbo Pascal. Szkoła programowania*, Wydawnictwo Robomatic 2000. *(dostępne w bibliotece uczelni)*
- http://pascal.kurs-programowania.pl
- http://4programmers.net
- T. M. Sadowski, *Turbo Pascal. Programowanie*, Helion 1996.