Informatyka 1

Wykład IX

Przetwarzanie tekstów

Robert Muszyński
ZPCiR ICT PWr

Zagadnienia: reprezentacja napisów znakowych, zmienne napisowe w Sun Pascalu, zgodność typów, operowanie na napisach: testowanie elementów, łączenie napisów, operacje we/wy, konwertowanie,

– Skład FoilTEX –

Reprezentacja napisów znakowych

- Tablice niepełne

```
CONST MaxString = 20;
TYPE String1 = PACKED ARRAY[1..MaxString] OF CHAR;
VAR Napis1: String1;
    Dlug_Napis1: INTEGER;
```


Reprezentacja napisów znakowych

- Tablice niepełne

```
CONST MaxString = 20;
TYPE String1 = PACKED ARRAY[1..MaxString] OF CHAR;
VAR Napis1: String1;
    Dlug_Napis1: INTEGER;
```

```
CONST MaxString = 20;
TYPE String2 = RECORD
        Znaki: PACKED ARRAY[1..MaxString] OF CHAR;
        Dlugosc: INTEGER
     END;
```

Reprezentacja napisów znakowych

- Tablice niepełne

```
CONST MaxString = 20;
TYPE String1 = PACKED ARRAY[1..MaxString] OF CHAR;
VAR Napis1: String1;
    Dlug_Napis1: INTEGER;
```

```
CONST MaxString = 20;
TYPE String2 = RECORD
        Znaki: PACKED ARRAY[1..MaxString] OF CHAR;
        Dlugosc: INTEGER
     END;
```

- Tablice ze znacznikiem (np. `chr(0)`)

```
CONST MaxString = 20;
TYPE String1 = PACKED ARRAY[1..MaxString] OF CHAR;
```

Reprezentacja napisów w Sun Pascalu

```
VAR napis1: ARRAY [1..25] OF CHAR;   { Zmienna napisowa o rozmiarze 25 }
```

Reprezentacja napisów w Sun Pascalu

```
VAR napis1: ARRAY [1..25] OF CHAR;   { Zmienna napisowa o rozmiarze 25 }
    napis2: STRING;                  { Zmienna napisowa o rozmiarze 80 }
    napis3: ALFA;                    { Zmienna napisowa o rozmiarze 10 }
```

Reprezentacja napisów w Sun Pascalu

```
VAR napis1: ARRAY [1..25] OF CHAR;    { Zmienna napisowa o rozmiarze 25 }
    napis2: STRING;                   { Zmienna napisowa o rozmiarze 80 }
    napis3: ALFA;                     { Zmienna napisowa o rozmiarze 10 }
    napis4: VARYING [25] OF CHAR;     { Zmienna napisowa o zmiennym rozmiarze }
                                      { (maksymalnie 25 znakow) }
```

Reprezentacja napisów w Sun Pascalu

```
VAR napis1: ARRAY [1..25] OF CHAR;    { Zmienna napisowa o rozmiarze 25 }
    napis2: STRING;                   { Zmienna napisowa o rozmiarze 80 }
    napis3: ALFA;                     { Zmienna napisowa o rozmiarze 10 }
    napis4: VARYING [25] OF CHAR;     { Zmienna napisowa o zmiennym rozmiarze }
                                      { (maksymalnie 25 znakow) }
    napis5: ARRAY [76..100] OF CHAR; { Jeszcze jedna zmienna napisowa }
```

Reprezentacja napisów w Sun Pascalu

```
VAR napis1: ARRAY [1..25] OF CHAR;   { Zmienna napisowa o rozmiarze 25 }
    napis2: STRING;                  { Zmienna napisowa o rozmiarze 80 }
    napis3: ALFA;                    { Zmienna napisowa o rozmiarze 10 }
    napis4: VARYING [25] OF CHAR;    { Zmienna napisowa o zmiennym rozmiarze }
                                     { (maksymalnie 25 znakow) }
    napis5: ARRAY [76..100] OF CHAR; { Jeszcze jedna zmienna napisowa }
BEGIN
  napis1 := 'wartosc1';  napis2 := 'wartosc2';  napis3 := 'wartosc3';
  napis4 := 'wartosc4';  napis5 := 'wartosc5';
```

Reprezentacja napisów w Sun Pascalu

```
VAR napis1: ARRAY [1..25] OF CHAR;   { Zmienna napisowa o rozmiarze 25 }
    napis2: STRING;                  { Zmienna napisowa o rozmiarze 80 }
    napis3: ALFA;                    { Zmienna napisowa o rozmiarze 10 }
    napis4: VARYING [25] OF CHAR;    { Zmienna napisowa o zmiennym rozmiarze }
                                     { (maksymalnie 25 znakow) }
    napis5: ARRAY [76..100] OF CHAR; { Jeszcze jedna zmienna napisowa }
BEGIN
  napis1 := 'wartosc1';  napis2 := 'wartosc2';  napis3 := 'wartosc3';
  napis4 := 'wartosc4';  napis5 := 'wartosc5';
  writeln('>',napis1,'<'); writeln('>',napis2,'<'); writeln('>',napis3,'<');
  writeln('>',napis4,'<'); writeln('>',napis5,'<');
END.
```

Reprezentacja napisów w Sun Pascalu

```
VAR napis1: ARRAY [1..25] OF CHAR;   { Zmienna napisowa o rozmiarze 25 }
    napis2: STRING;                  { Zmienna napisowa o rozmiarze 80 }
    napis3: ALFA;                    { Zmienna napisowa o rozmiarze 10 }
    napis4: VARYING [25] OF CHAR;    { Zmienna napisowa o zmiennym rozmiarze }
                                     { (maksymalnie 25 znakow) }
    napis5: ARRAY [76..100] OF CHAR; { Jeszcze jedna zmienna napisowa }
BEGIN
  napis1 := 'wartosc1';  napis2 := 'wartosc2';  napis3 := 'wartosc3';
  napis4 := 'wartosc4';  napis5 := 'wartosc5';
  writeln('>',napis1,'<'); writeln('>',napis2,'<'); writeln('>',napis3,'<');
  writeln('>',napis4,'<'); writeln('>',napis5,'<');
END.
```

```
>wartosc1                    <
```

Reprezentacja napisów w Sun Pascalu

```
VAR napis1: ARRAY [1..25] OF CHAR;   { Zmienna napisowa o rozmiarze 25 }
    napis2: STRING;                  { Zmienna napisowa o rozmiarze 80 }
    napis3: ALFA;                    { Zmienna napisowa o rozmiarze 10 }
    napis4: VARYING [25] OF CHAR;    { Zmienna napisowa o zmiennym rozmiarze }
                                     { (maksymalnie 25 znakow) }
    napis5: ARRAY [76..100] OF CHAR; { Jeszcze jedna zmienna napisowa }
BEGIN
  napis1 := 'wartosc1';  napis2 := 'wartosc2';  napis3 := 'wartosc3';
  napis4 := 'wartosc4';  napis5 := 'wartosc5';
  writeln('>',napis1,'<'); writeln('>',napis2,'<'); writeln('>',napis3,'<');
  writeln('>',napis4,'<'); writeln('>',napis5,'<');
END.
```

```
>wartosc1                <
>wartosc2                                                              <
>wartosc3  <
```

Reprezentacja napisów w Sun Pascalu

```
VAR napis1: ARRAY [1..25] OF CHAR;   { Zmienna napisowa o rozmiarze 25 }
    napis2: STRING;                  { Zmienna napisowa o rozmiarze 80 }
    napis3: ALFA;                    { Zmienna napisowa o rozmiarze 10 }
    napis4: VARYING [25] OF CHAR;    { Zmienna napisowa o zmiennym rozmiarze }
                                     { (maksymalnie 25 znakow) }
    napis5: ARRAY [76..100] OF CHAR; { Jeszcze jedna zmienna napisowa }
BEGIN
  napis1 := 'wartosc1';  napis2 := 'wartosc2';  napis3 := 'wartosc3';
  napis4 := 'wartosc4';  napis5 := 'wartosc5';
  writeln('>',napis1,'<'); writeln('>',napis2,'<'); writeln('>',napis3,'<');
  writeln('>',napis4,'<'); writeln('>',napis5,'<');
END.
```

```
>wartosc1                 <
>wartosc2                                                              <
>wartosc3  <
>wartosc4<
```

Reprezentacja napisów w Sun Pascalu

```
VAR napis1: ARRAY [1..25] OF CHAR;   { Zmienna napisowa o rozmiarze 25 }
    napis2: STRING;                  { Zmienna napisowa o rozmiarze 80 }
    napis3: ALFA;                    { Zmienna napisowa o rozmiarze 10 }
    napis4: VARYING [25] OF CHAR;    { Zmienna napisowa o zmiennym rozmiarze }
                                     { (maksymalnie 25 znakow) }
    napis5: ARRAY [76..100] OF CHAR; { Jeszcze jedna zmienna napisowa }
BEGIN
  napis1 := 'wartosc1';  napis2 := 'wartosc2';  napis3 := 'wartosc3';
  napis4 := 'wartosc4';  napis5 := 'wartosc5';
  writeln('>',napis1,'<'); writeln('>',napis2,'<'); writeln('>',napis3,'<');
  writeln('>',napis4,'<'); writeln('>',napis5,'<');
END.
```

```
>wartosc1                 <
>wartosc2                                                              <
>wartosc3  <
>wartosc4<
>wartosc5                 <
```

Zgodność typów napisowych w Sun Pascalu

Zmienne napisowe są ze sobą zgodne przy następujących zastrzeżeniach:

- Stała napisowa przypisywana do zmiennej napisowej o stałym rozmiarze nie może być dłuższa od zmiennej;

Zgodność typów napisowych w Sun Pascalu

Zmienne napisowe są ze sobą zgodne przy następujących zastrzeżeniach:

- Stała napisowa przypisywana do zmiennej napisowej o stałym rozmiarze nie może być dłuższa od zmiennej;
- Przypisywanie wartości zmiennych napisowych o stałym rozmiarze jest możliwe, jeśli zmienne te są równych długości;

Zgodność typów napisowych w Sun Pascalu

Zmienne napisowe są ze sobą zgodne przy następujących zastrzeżeniach:

- Stała napisowa przypisywana do zmiennej napisowej o stałym rozmiarze nie może być dłuższa od zmiennej;
- Przypisywanie wartości zmiennych napisowych o stałym rozmiarze jest możliwe, jeśli zmienne te są równych długości;
- Jeśli napis przypisywany do zmiennej napisowej jest dłuższy od tej zmiennej, to zostaje on skrócony do długości zmiennej;

Zgodność typów napisowych w Sun Pascalu

Zmienne napisowe są ze sobą zgodne przy następujących zastrzeżeniach:

- Stała napisowa przypisywana do zmiennej napisowej o stałym rozmiarze nie może być dłuższa od zmiennej;
- Przypisywanie wartości zmiennych napisowych o stałym rozmiarze jest możliwe, jeśli zmienne te są równych długości;
- Jeśli napis przypisywany do zmiennej napisowej jest dłuższy od tej zmiennej, to zostaje on skrócony do długości zmiennej;
- Jeśli napis przypisywany do zmiennej napisowej o stałym rozmiarze jest krótszy do tej zmiennej, to zostaje on uzupełniony znakami spacji;

Operowanie napisami w Sun Pascalu

- kopiowanie napisów (:=) — zgodnie z regułami przedstawionymi na poprzednim slajdzie;

Operowanie napisami w Sun Pascalu

- kopiowanie napisów (:=) — zgodnie z regułami przedstawionymi na poprzednim slajdzie;
- określanie długości napisu (funkcja `Length`) — zwraca aktualna długość napisu;

Operowanie napisami w Sun Pascalu

- kopiowanie napisów (:=) — zgodnie z regułami przedstawionymi na poprzednim slajdzie;
- określanie długości napisu (funkcja `Length`) — zwraca aktualna długość napisu;
- porównywanie napisów (=, <>, <, <=, >, >=,) — operatory =, <> tylko napisy równej długości;

Operowanie napisami w Sun Pascalu

- kopiowanie napisów (:=) — zgodnie z regułami przedstawionymi na poprzednim slajdzie;
- określanie długości napisu (funkcja `Length`) — zwraca aktualna długość napisu;
- porównywanie napisów (=, <>, <, <=, >, >=,) — operatory =, <> tylko napisy równej długości;
- dopasowywanie wzorców (funkcja `Index`) — zwraca indeks wskazujący na pierwsze wystąpienie wzorca w napisie;

Operowanie napisami w Sun Pascalu

- kopiowanie napisów (:=) — zgodnie z regułami przedstawionymi na poprzednim slajdzie;
- określanie długości napisu (funkcja `Length`) — zwraca aktualna długość napisu;
- porównywanie napisów (=, <>, <, <=, >, >=,) — operatory =, <> tylko napisy równej długości;
- dopasowywanie wzorców (funkcja `Index`) — zwraca indeks wskazujący na pierwsze wystąpienie wzorca w napisie;
- łączenie napisów (funkcja `Concat` (vel +) oraz procedura `Stradd`) — pozwalają na kombinowanie ze sobą napisów;

Operowanie napisami w Sun Pascalu

- kopiowanie napisów (:=) — zgodnie z regułami przedstawionymi na poprzednim slajdzie;
- określanie długości napisu (funkcja `Length`) — zwraca aktualna długość napisu;
- porównywanie napisów (=, <>, <, <=, >, >=,) — operatory =, <> tylko napisy równej długości;
- dopasowywanie wzorców (funkcja `Index`) — zwraca indeks wskazujący na pierwsze wystąpienie wzorca w napisie;
- łączenie napisów (funkcja `Concat` (vel +) oraz procedura `Stradd`) — pozwalają na kombinowanie ze sobą napisów;
- wybieranie fragmentów napisu (funkcja `Substr`) — zwraca wskazany fragment napisu;

Operowanie napisami w Sun Pascalu

- kopiowanie napisów (:=) — zgodnie z regułami przedstawionymi na poprzednim slajdzie;
- określanie długości napisu (funkcja `Length`) — zwraca aktualna długość napisu;
- porównywanie napisów (=, <>, <, <=, >, >=,) — operatory =, <> tylko napisy równej długości;
- dopasowywanie wzorców (funkcja `Index`) — zwraca indeks wskazujący na pierwsze wystąpienie wzorca w napisie;
- łączenie napisów (funkcja `Concat` (vel +) oraz procedura `Stradd`) — pozwalają na kombinowanie ze sobą napisów;
- wybieranie fragmentów napisu (funkcja `Substr`) — zwraca wskazany fragment napisu;
- pomijanie wiodących znaków białych (funkcja `Trim`);

Operowanie napisami — przykłady

- Testowanie elementów

```
FUNCTION MalaLitera(c: CHAR): BOOLEAN;
BEGIN
  MalaLitera := ('a' <= c) AND (c <= 'z')
END; {MalaLitera}
```

Operowanie napisami — przykłady

- Testowanie elementów

```
FUNCTION MalaLitera(c: CHAR): BOOLEAN;
BEGIN
  MalaLitera := ('a' <= c) AND (c <= 'z')
END; {MalaLitera}
FUNCTION DuzaLitera(c: CHAR): BOOLEAN;
BEGIN
  DuzaLitera := ('A' <= c) AND (c <= 'Z')
END; {DuzaLitera}
```

Operowanie napisami — przykłady

- Testowanie elementów

```
FUNCTION MalaLitera(c: CHAR): BOOLEAN;
BEGIN
  MalaLitera := ('a' <= c) AND (c <= 'z')
END; {MalaLitera}
FUNCTION DuzaLitera(c: CHAR): BOOLEAN;
BEGIN
  DuzaLitera := ('A' <= c) AND (c <= 'Z')
END; {DuzaLitera}
FUNCTION JestLitera(c: CHAR): BOOLEAN;
BEGIN
  JestLitera := MalaLitera(c) OR DuzaLitera(c)
END; {JestLitera}
```

Operowanie napisami — przykłady

- Testowanie elementów

```
FUNCTION MalaLitera(c: CHAR): BOOLEAN;
BEGIN
  MalaLitera := ('a' <= c) AND (c <= 'z')
END; {MalaLitera}
FUNCTION DuzaLitera(c: CHAR): BOOLEAN;
BEGIN
  DuzaLitera := ('A' <= c) AND (c <= 'Z')
END; {DuzaLitera}
FUNCTION JestLitera(c: CHAR): BOOLEAN;
BEGIN
  JestLitera := MalaLitera(c) OR DuzaLitera(c)
END; {JestLitera}
```

```
{...}
  tylko_male := TRUE;
  FOR i := 1 TO Length(napis) DO
    IF NOT MalaLitera(napis[i]) THEN
      tylko_male := FALSE;
```

Operowanie napisami — przykłady

- Testowanie elementów

```
FUNCTION MalaLitera(c: CHAR): BOOLEAN;
BEGIN
  MalaLitera := ('a' <= c) AND (c <= 'z')
END; {MalaLitera}
FUNCTION DuzaLitera(c: CHAR): BOOLEAN;
BEGIN
  DuzaLitera := ('A' <= c) AND (c <= 'Z')
END; {DuzaLitera}
FUNCTION JestLitera(c: CHAR): BOOLEAN;
BEGIN
  JestLitera := MalaLitera(c) OR DuzaLitera(c)
END; {JestLitera}
```

```
{...}
  tylko_male := TRUE;
  FOR i := 1 TO Length(napis) DO
    IF NOT MalaLitera(napis[i]) THEN
      tylko_male := FALSE;
```

- Łączenie napisów

```
VAR napis: varying [12] of char;
BEGIN
  napis := 'To ' + 'jest ' + 'napis.';
```

Operowanie napisami — przykłady

- Testowanie elementów

```
FUNCTION MalaLitera(c: CHAR): BOOLEAN;
BEGIN
  MalaLitera := ('a' <= c) AND (c <= 'z')
END; {MalaLitera}
FUNCTION DuzaLitera(c: CHAR): BOOLEAN;
BEGIN
  DuzaLitera := ('A' <= c) AND (c <= 'Z')
END; {DuzaLitera}
FUNCTION JestLitera(c: CHAR): BOOLEAN;
BEGIN
  JestLitera := MalaLitera(c) OR DuzaLitera(c)
END; {JestLitera}
```

```
{...}
  tylko_male := TRUE;
  FOR i := 1 TO Length(napis) DO
    IF NOT MalaLitera(napis[i]) THEN
      tylko_male := FALSE;
```

- Łączenie napisów

```
VAR napis: varying [12] of char;
BEGIN
  napis := 'To ' + 'jest ' + 'napis.';
  napis := Concat('Jezeli ',napis);
```

Operowanie napisami — przykłady

- Testowanie elementów

```
FUNCTION MalaLitera(c: CHAR): BOOLEAN;
BEGIN
  MalaLitera := ('a' <= c) AND (c <= 'z')
END; {MalaLitera}
FUNCTION DuzaLitera(c: CHAR): BOOLEAN;
BEGIN
  DuzaLitera := ('A' <= c) AND (c <= 'Z')
END; {DuzaLitera}
FUNCTION JestLitera(c: CHAR): BOOLEAN;
BEGIN
  JestLitera := MalaLitera(c) OR DuzaLitera(c)
END; {JestLitera}
```

```
{...}
  tylko_male := TRUE;
  FOR i := 1 TO Length(napis) DO
    IF NOT MalaLitera(napis[i]) THEN
      tylko_male := FALSE;
```

- Łączenie napisów

```
VAR napis: varying [12] of char;
BEGIN
  napis := 'To ' + 'jest ' + 'napis.';
  napis := Concat('Jezeli ',napis);
  napis := 'Poczatek';
  Stradd(napis, ' i koniec.');
```

Operowanie napisami — przykłady

- Testowanie elementów

```
FUNCTION MalaLitera(c: CHAR): BOOLEAN;
BEGIN
  MalaLitera := ('a' <= c) AND (c <= 'z')
END; {MalaLitera}
FUNCTION DuzaLitera(c: CHAR): BOOLEAN;
BEGIN
  DuzaLitera := ('A' <= c) AND (c <= 'Z')
END; {DuzaLitera}
FUNCTION JestLitera(c: CHAR): BOOLEAN;
BEGIN
  JestLitera := MalaLitera(c) OR DuzaLitera(c)
END; {JestLitera}
```

```
{...}
  tylko_male := TRUE;
  FOR i := 1 TO Length(napis) DO
    IF NOT MalaLitera(napis[i]) THEN
      tylko_male := FALSE;
```

- Łączenie napisów

```
VAR napis: varying [12] of char;
BEGIN
  napis := 'To ' + 'jest ' + 'napis.';
  napis := Concat('Jezeli ',napis);
  napis := 'Poczatek';
  Stradd(napis, ' i koniec.');
```

- Operacje we/wy

```
VAR napis: varying [12] of char;
    plik : text;

{...}
READLN/WRITELN(napis);
READLN/WRITELN(plik,napis);
```

```
FUNCTION WczytajWyraz( plik: TEXT; VAR s: String ): INTEGER;
VAR c : CHAR;
    i : INTEGER;
BEGIN    {wpierw pomijamy nielitery uwazajac na EOF/EOLN}
  c := ' ';                      {spacja nie jest litera}
  REPEAT
    WHILE ((NOT EOF(plik)) AND EOL(plik)) DO
      READLN(plik);
    IF (NOT EOF(plik)) THEN READ(plik,c)
  UNTIL (EOF(plik) OR JestLitera(c));
  {byc moze byla to litera, wtedy zapamietujemy}
  IF NOT JestLitera(c) THEN
    i := 0
  ELSE
  BEGIN
    i := 1;  s[1] := c;
  END;
 {czytamy tylko litery az do: EOF, EOL, lub nie-litery}
  WHILE (JestLitera(c) AND ((NOT EOL(plik)) AND (NOT EOF(plik)))) DO
  BEGIN
    READ(plik,c);
    IF JestLitera(c) THEN
      IF i < Length[s] THEN
      BEGIN
        i := i + 1;  s[i] := c;
      END
      ELSE
        WRITELN('**Blad: przekroczona dlugosc napisu')
  END;
  WczytajWyraz := i;
END; {WczytajWyraz}
```

```
FUNCTION WczytajWyraz( plik: TEXT; VAR s: String ): INTEGER;
VAR c : CHAR;
    i : INTEGER;
BEGIN    {wpierw pomijamy nielitery uwazajac na EOF/EOLN}
  c := ' ';                      {spacja nie jest litera}
  REPEAT
    WHILE ((NOT EOF(plik)) AND EOL(plik)) DO
      READLN(plik);
    IF (NOT EOF(plik)) THEN READ(plik,c)
  UNTIL (EOF(plik) OR JestLitera(c));
  {byc moze byla to litera, wtedy zapamietujemy}
  IF NOT JestLitera(c) THEN
    i := 0
  ELSE
  BEGIN
    i := 1;  s[1] := c;
  END;
 {czytamy tylko litery az do: EOF, EOL, lub nie-litery}
  WHILE (JestLitera(c) AND ((NOT EOL(plik)) AND (NOT EOF(plik)))) DO
  BEGIN
    READ(plik,c);
    IF JestLitera(c) THEN
      IF i < Length[s] THEN
      BEGIN
        i := i + 1;  s[i] := c;
      END
      ELSE
        WRITELN('**Blad: przekroczona dlugosc napisu')
  END;
  WczytajWyraz := i;
END; {WczytajWyraz}
```

```
FUNCTION WczytajWyraz( plik: TEXT; VAR s: String ): INTEGER;
VAR c : CHAR;
    i : INTEGER;
BEGIN    {wpierw pomijamy nielitery uwazajac na EOF/EOLN}
  c := ' ';                      {spacja nie jest litera}
  REPEAT
    WHILE ((NOT EOF(plik)) AND EOL(plik)) DO
      READLN(plik);
    IF (NOT EOF(plik)) THEN READ(plik,c)
  UNTIL (EOF(plik) OR JestLitera(c));
  {byc moze byla to litera, wtedy zapamietujemy}
  IF NOT JestLitera(c) THEN
    i := 0
  ELSE
  BEGIN
    i := 1;  s[1] := c;
  END;
 {czytamy tylko litery az do: EOF, EOL, lub nie-litery}
  WHILE (JestLitera(c) AND ((NOT EOL(plik)) AND (NOT EOF(plik)))) DO
  BEGIN
    READ(plik,c);
    IF JestLitera(c) THEN
      IF i < Length[s] THEN
      BEGIN
        i := i + 1;  s[i] := c;
      END
      ELSE
        WRITELN('**Blad: przekroczona dlugosc napisu')
  END;
  WczytajWyraz := i;
END; {WczytajWyraz}
```

- Konwertowanie

```
FUNCTION MalaNaDuza(c: CHAR): CHAR;
BEGIN
  IF ('a' <= c) AND (c <= 'z')
    THEN MalaNaDuza := CHR(ORD(c) - ORD('a') + ORD('A'))
    ELSE MalaNaDuza := c
END; {MalaNaDuza}
```

- Konwertowanie

```
FUNCTION MalaNaDuza(c: CHAR): CHAR;
BEGIN
  IF ('a' <= c) AND (c <= 'z')
    THEN MalaNaDuza := CHR(ORD(c) - ORD('a') + ORD('A'))
    ELSE MalaNaDuza := c
END; {MalaNaDuza}
```

```
PROCEDURE ZamienZnaki( VAR z1, z2: CHAR );
VAR z: CHAR;
BEGIN
  z  := z1; z1 := z2; z2 := z
END; {ZamienZnaki}
```

- Konwertowanie

```
FUNCTION MalaNaDuza(c: CHAR): CHAR;
BEGIN
  IF ('a' <= c) AND (c <= 'z')
    THEN MalaNaDuza := CHR(ORD(c) - ORD('a') + ORD('A'))
    ELSE MalaNaDuza := c
END; {MalaNaDuza}
```

```
PROCEDURE ZamienZnaki( VAR z1, z2: CHAR );
VAR z: CHAR;
BEGIN
  z  := z1; z1 := z2; z2 := z
END; {ZamienZnaki}
PROCEDURE OdwrocNapis( VAR s: String );
VAR i: INTEGER;
BEGIN
  FOR i := 1 TO Length(s) DIV 2 DO
    ZamienZnaki( s[i], s[Length(s) + 1 - i]);
END; {OdwrocNapis}
```

Indeks

- Reprezentacja napisów znakowych
- Reprezentacja napisów w Sun Pascalu
- Zgodność typów napisowych w Sun Pascalu
- Operowanie napisami w Sun Pascalu
- Operowanie napisami — przykłady:
  - testowanie elementów
  - łączenie napisów
  - operacje we/wy
  - konwertowanie

