

PASCAL – OBSŁUGA PLIKÓW

1. Typ wyliczeniowy

- W Pascalu istnieje możliwość tworzenia własnych typów. Przekonałeś się już o tym tworząc typy rekordowe. Warto jeszcze zauważyć, że tworzenie własnych typów pozwala na tworzenie zmiennych o z góry określonych wartościach. Poniższy przykład pokazuje również w jaki sposób wykorzystasz własny typ do indeksowania tablicy:

```
program typ_wyliczeniowy;
type
  t_dzien_tygodnia=(poniedzialek, wtorek, sroda, czwartek, piatek, sobota,
  niedziela);
const
  tydzien: array[t_dzien_tygodnia]of string = ('poniedzialek', 'wtorek',
  'sroda', 'czwartek', 'piatek', 'sobota', 'niedziela');
  {ta tablica bedzie posiadac siedem pol indeksowanych dniami tygodnia}
  {w kazdym polu umieszczamy string z nazwa dnia tygodnia}
var
  dzien:t_dzien_tygodnia;
begin
  for dzien:=poniedzialek to niedziela do
    begin
      writeln(tydzien[dzien]); {wypisujemy kolejne elementy tablicy}
    end;
  readln;
end.
```

- WAŻNE:** typów wyliczeniowych nie możemy ani wypisać, ani wprowadzić w normalny sposób, bowiem to nie są zmienne typu `string`, należy o tym pamiętać. Aby wartość typu wyliczeniowego wypisać na ekranie, należy zamienić go dopiero na `string` (Np. przez wykorzystanie dodatkowej tablicy zawierającej zmienne `string` przyjmujące jej wartości, jak w przykładzie)
- Zadanie 1** (1pkt): Napisz program przechowujący w pamięci tablicę indeksowaną typem `pora_dnia` (o możliwych wartościach: `rano`, `przedpoludnie`, `popoludnie`, `wieczor`, `noc`). Kolejne pola tablicy to zmienne typu `BOOLEAN`.
- Inicjalizacja zmiennych:** Na początku programu przypisz wszystkim porom dnia w tablicy wartość `FALSE`, oprócz nocy, która będzie `TRUE`.
- Program ma pozwolić użytkownikowi sprawdzenie w grafiku, o jakiej porze dnia ma jeszcze wolny czas (wartość `TRUE`), czyli po dialogu wstępnym (Czy chcesz sprawdzić kiedy masz wolny czas?) ma przeszukać tablicę w poszukiwaniu wartości `TRUE`. Ostatecznie podaje wynik: porę dnia.

2. Obsługa plików

- Aby Pascal wiedział, jak zinterpretować plik na którym chcemy pracować, należy określić jego strukturę, czyli typ pliku. Każdy plik, na którym będziemy pracować, musi zostać zadeklarowany, podobnie jak zmienna. Będzie to tzw. `zmienna_plikowa`. Jej typ może przyjąć postać (zależności od struktury pliku na którym chcemy pracować):

```
var
plik_textowy: Text;           {plik tekstowy, jak w notatniku}
plik_elementowy: file of Typ; {plik złożony ze zmiennych typu Typ}
plik_amorficzny: file;       {plik amorficzny, nieokreślony}
```

- W programie będziemy wykorzystywać `zmienną_plikową`, którą będziemy traktować jak wirtualny plik, aby jednak móc zapisać do fizycznego pliku, musimy ją powiązać z tym plikiem za pomocą procedury:
`assign (zmienna_plikowa, nazwa_pliku);`
- `nazwa_pliku` może być również ścieżką do tego pliku
- aby zainicjalizować korzystanie z pliku, należy go otworzyć przy pomocy któregoś z poleceń (do wyboru):

```
reset (zmienna_plikowa);    {otwiera plik w trybie do odczytu}
rewrite (zmienna_plikowa);  {otwiera plik w trybie do zapisu lub tworzy
nowy, jeśli nie istniał, uwaga: kasuje poprzednią zawartość }
append (zmienna_plikowa);   {otwiera plik w trybie do zapisu, pozwala
dopisywać do istniejącej zawartości }
```

- pliki tekstowe: **dostęp sekwencyjny** – oznacza to, że aby dojść do jakiegoś konkretnego miejsca w pliku należy przejść po kolei całą zawartość (jak kasecie wideo)
- możemy wspierać się funkcjami:

```
eoln (zmienna_plikowa):boolean; {jeśli koniec wiersza: TRUE}
eof (zmienna_plikowa):boolean;  {jeśli koniec pliku: TRUE}
```

- pliki elementowe: **dostęp swobodny** – można wybierać miejsce do którego chcemy dotrzeć wybierając numer elementu, za którym chcemy się znaleźć (jak pliki w katalogu). Poruszamy się w oparciu o instrukcje:

```
seek (zmienna_plikowa, pozycja); {ustawia wskaźnik na zadanej pozycji}
FilePos (zmienna_plikowa): LongInt; {podaje aktualne położenie wskaźnika}
FileSize (zmienna_plikowa): LongInt; {informuje o rozmiarze pliku w
elementach}
```

- elementy pliku liczone są od 0, czyli zapisy:

```
seek (plik, 5)           {ustawia na 6 element, za 5}
seek (plik, FileSize(plik)) {ustawia za ostatnim elementem, na końcu pliku}
```

- Aby zapisać do pliku dane przechowywane w buforze pamięci i zakończyć pracę z plikiem, należy skorzystać z procedury
`close (zmienna_plikowa);`

- Przykład: Poniższy program może pełnić funkcję bardzo prostego edytora. Po wstępnym komunikacie wszystko co zostanie napisane na ekranie trafi do pliku. Pusty wiersz kończy edycję.

```

program zad1;
  uses crt;
Const
  nazwa_pliku='pascal_tekst.txt';
Var
  plik:text;
  wiersz:string;
Begin
  TextColor(black);
  TextBackGround(lightgray);
  clrscr;
  write('Wszystko co napiszesz zostanie zapisane do pliku: ');
  writeln(nazwa_pliku);
  writeln('Kazdy wiersz koncz ENTEREM.');
```

writeln('Aby zakonczyc, wcisnij ENTER w pustym wierszu.');

```

  write('_____');
```

writeln('_____');

```

  writeln('');
  assign(plik, nazwa_pliku);
  rewrite(plik);
  repeat
    ReadLn(wiersz);
    if (wiersz<>'') then writeln(plik, wiersz);
until (wiersz='');
```

```

  close (plik);
  clrscr;
  writeln('Zapisywanie do pliku zakonczono.');
```

writeln('Obecnie plik ma postac: ');

```

  write('_____');
```

writeln('_____');

```

  reset(plik);
  while not eof(plik) do
  begin
    readln(plik, wiersz);
    writeln(wiersz);
  end;
  close(plik);
  readln;
End.
```

- Po wykonaniu programu w twoim katalogu powinien pojawić się plik pascal_tekst.txt (jeśli poprawnie ustawiłeś zmienne środowiskowe)
- zwróć uwagę na sposób zapisu i odczytu z pliku.

3. Zadanie 2 (1pkt):

- Przypomnij sobie program z zajęć na temat tablic i rekordów pobierający dane osobowe.
- Zmień program tak, aby dane zapisywał w pliku elementowym. Po wprowadzeniu danych dla 3 osób powinien zakończyć zapis i na wyczyszczonym ekranie wydrukować wszystkie rekordy.
- **Zadanie 3** (1pkt):
- Napisz 'wtyczkę' do powyższego programu. Bazując na stworzonym pliku `baza_danych.dat` napisz program, który zapyta użytkownika, której osoby dane chce zobaczyć, a następnie wydrukuje wskazane informacje na ekranie.