

Ćwiczenie 1- Programowanie obiektowe i modułowe 1

1. Zapisz jako aplikację konsolową ZESP.DPR następujący program:

```
type TZesp = object
    re, im: real;
    procedure UstawZesp(x, y: real);
    procedure CzytajZesp;
    procedure WyswZesp;
    procedure WyznaczWyk (var modul, fi: real);
    procedure WyswWyk;
end;

procedure TZesp.UstawZesp(x, y: real);
begin
    re:=x;
    im:=y;
end;

procedure TZesp.CzytajZesp;
{ czytanie liczby zespolonej w postaci algebraicznej }
begin
    write('re=');readln(re);
    write('im=');readln(im);
end;

procedure TZesp.WyswZesp;
{ wyświetlanie liczby w reprezentacji algebraicznej }
begin
    writeln(re:0:2, ' + ', im:0:2,'i');
end;

procedure TZesp.WyznaczWyk(var modul, fi: real);
{ konwersja na postać wykładniczą }
begin
    modul := sqrt(re*re +im*im);
    if re<>0 then fi := arctan(im/re) else fi := pi/2;
end;

procedure TZesp.WyswWyk;
{ wyświetlanie liczby w reprezentacji wykładniczej }
var m, f: real;
begin
    WyznaczWyk(m, f);
    writeln(m:0:2, ' exp(', f:0:2,'i)');
end;

var a:TZesp;
begin
    writeln('Podaj liczbę zespoloną');
    a.CzytajZesp;
    writeln('Postac algebraiczna: ');
    a.WyswZesp;
    writeln('Postac wykladnicza: ');
    a.WyswWyk;
end.
```

Program zawiera definicję typu obiektowego TZesp przeznaczonego do zapisu liczb zespolonych. Zadeklarowane metody przeznaczone są do wprowadzania i wyświetlania liczb zespolonych w notacji algebraicznej ($a+bi$) oraz do konwersji i wyświetlania liczb w postaci wykładniczej ($m \cdot e^{i \cdot \phi}$). Przeanalizuj program. Uruchom program dla przykładowych zestawów danych.

2. Opracuj nową modułową wersję programu. Definicję typu obiektowego umieść w module **m1** zapisanym w pliku M1.PAS. Program główny umieść w pliku P1.DPR. Przetestuj program.
3. Opracuj moduł **m2** i zapisz go w pliku M2.PAS. Moduł powinien zawierać definicje typu TZespolone dziedziczącego składowe typu TZesp. Dwie nowe metody typu przeznaczone są do konwersji liczby zespolonej z postaci wykładniczej do algebraicznej oraz wprowadzania liczby w postaci wykładniczej.

```
type TZespolone = object(TZesp)
    procedure UstawWyk(modul, fi: real);
    procedure CzytajWyk;
end;
```

Zdefiniuj poszczególne metody:

UstawWyk – ustalenie wartości części rzeczywistej i urojonej liczby na podstawie modułu i argumentu danych parametrami,

CzytajWyk – wprowadzanie liczby w postaci wykładniczej z konwersją do postaci algebraicznej,

Uwaga: W module **m2** powinno być umieszczone odwołanie do modułu **m1**.

Do konwersji na postać algebraiczną wykorzystaj wzory:

$$re = modul \cdot \cos(fi), \quad im = modul \cdot \sin(fi);$$

Opracuj i zapisz w pliku P2.DPR program testujący moduł **m2**.

4. Zapisz jako aplikację konsolową KALK.DPR następujący program:

```
uses m2;
type TKalk = object
    a, b: TZespolone;
    procedure UstawDane(x, y: TZespolone);
    procedure Suma(var wynik: TZespolone);
    procedure Różnica(var wynik: TZespolone);
    procedure Iloczyn(var wynik: TZespolone);
    procedure Iloraz(var wynik: TZespolone);
end;

procedure TKalk.UstawDane(x, y: TZespolone);
begin
    a:=x;
    b:=y;
end;

procedure TKalk.Suma(var wynik: TZespolone);
begin
    wynik.re:=a.re+b.re;
    wynik.im:=a.im+b.im;
end;
```

```

procedure TKalk.Roznica(var wynik: TZespolone);
begin
{ wstaw zapis metody wyznaczającej różnicę dwu liczb zespolonych }
end;

procedure TKalk.Iloczyn(var wynik: TZespolone);
var m1, f1, m2, f2: real;
begin
a.WyznaczWyk(m1, f1);
b.WyznaczWyk(m2, f2);
wynik.UstawWyk(m1*m2, f1+f2);
end;

procedure TKalk.Iloraz(var wynik: TZespolone);
begin
{ wstaw zapis metody wyznaczającej iloraz dwu liczb zespolonych }
end;

var a, b, c, d: TZespolone;
      w: TKalk;
begin
writeln('Podaj liczbe zespolona A');
a.CzytajZesp;
writeln('Podaj liczbe zespolona B');
b.CzytajZesp;
writeln('Podaj liczbe zespolona C');
c.CzytajZesp;
w.UstawDane(a, b);
w.Sum(a, b, d);
w.UstawDane(c, d);
w.Iloczyn(d);
writeln('Wynik dzialania (A + B) * C');
d.WyswZesp;
{ uzupełnij program o instrukcje testujące nowe procedury }
end.

```

Program zawiera definicję typu obiektowego TKalk, którego metody realizują działania na liczbach zespolonych. Zadeklarowane metody przeznaczone są do ustalania wartości dwu liczb zespolonych oraz do wykonywania czterech podstawowych działań na tych liczbach. W programie zamieszczono kompletne definicje metod dla dwóch działań: dodawania i mnożenia. Uzupełnij definicje metod Roznica i Iloraz. Przetestuj program dla przykładowych zestawów danych.

Wskazówka: Działania na liczbach zespolonych:

Dodawanie:

$$(x_{re} + i*x_{im}) + (y_{re} + i*y_{im}) = (x_{re}+y_{re}) + i*(x_{im}+y_{im})$$

Odejmowanie:

$$(x_{re} + i*x_{im}) - (y_{re} + i*y_{im}) = (x_{re}-y_{re}) + i*(x_{im}-y_{im})$$

Mnożenie i dzielenie w reprezentacji wykładniczej:

$$m_1 * \exp(i*\phi_1) * m_2 * \exp(i*\phi_2) = m_1 * m_2 * \exp(i*(\phi_1 + \phi_2))$$

$$m_1 * \exp(i*\phi_1) / m_2 * \exp(i*\phi_2) = m_1 / m_2 * \exp(i*(\phi_1 - \phi_2))$$