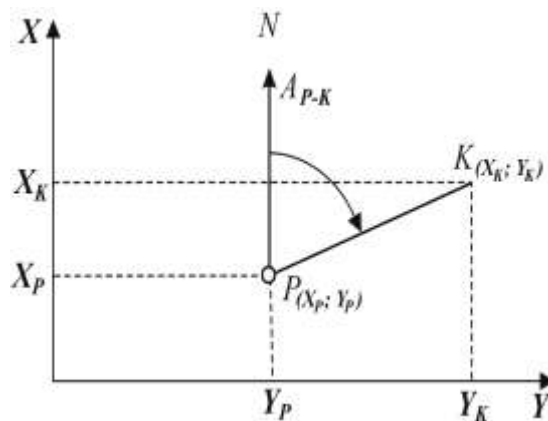


## Obliczenie azymutów ze współrzędnych punktów

### Azymuty i długości - tablica struktur, funkcje, zapis wyników do pliku

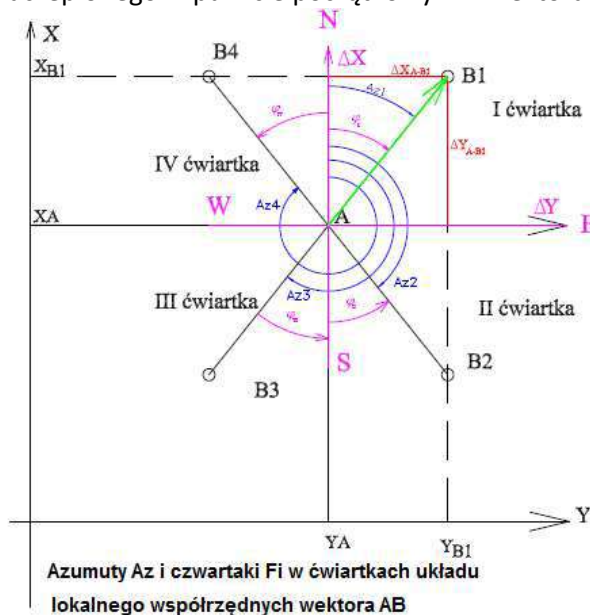
#### Układ współrzędnych geodezyjnych i pojęcie azymutu

W geodezji obowiązuje geodezyjny, kartezjański, prostokątny układ współrzędnych, w którym oś X skierowana jest pionowo do góry (kierunek północy) a oś Y jest skierowana poziomo w prawo. Kąty i azymuty liczone są w prawo – odwrotnie niż w układzie matematycznym, ale zawsze od osi X do osi Y. Wzory matematyczne są takie same w obu układach.



Dowolny punkt P w tym układzie określamy parą współrzędnych zapisywaną jako  $P(X_p; Y_p)$ . Azymutem lub kątem kierunkowym odcinka (wektora) PK nazywamy kąt zawarty pomiędzy kierunkiem północy (dodatni kierunek osi X) a kierunkiem do drugiego punktu (K) (określającego azymut obliczanego odcinka), liczony zgodnie z ruchem wskazówek zegara (prawoskrętnie) i mogący przyjmować wartości od  $0^\circ$  do  $400^\circ$ , lub  $0^\circ - 360^\circ$  lub  $0 - 2\pi$  [rad].

#### Azymut w układach ćwiartek układu współrzędnych, zaczepionego w punkcie początkowym A wektora AB



## Obliczanie azymutów wektora AB, o początku A i końcu B

### WZORY

Oznaczenia:

Az - azymut, Fi - czwartak, Psi - azymut + 50[grad] d - długość

N, E, S, W - oznaczenia kierunków świata: ( północ, wschód, południe, zachód)

$DX ( AB ) = XB - XA$   $DY ( AB ) = YB - YA$  - przyrosty współrzędnych

$Fi = |\arctan ( DY/DX )|$  - czwartak

$Psi = Fi + 50g$   $\tan ( Psi ) = ( DX + DY ) / ( DX - DY )$

### WZORY

Oznaczenia:

Az - azymut, Fi - czwartak, Psi - azymut + 50[grad] d - długość

N, E, S, W - oznaczenia kierunków świata: ( północ, wschód, południe, zachód)

$DX ( AB ) = XB - XA$   $DY ( AB ) = YB - YA$  - przyrosty współrzędnych

$Fi = |\arctan ( DY/DX )|$  - czwartak

$Psi = Fi + 50g$   $\tan ( Psi ) = ( DX + DY ) / ( DX - DY )$

Azymut A i czwartak Fi w gradach

Ćwiartka	Zakres azymutu	Azymut i czwartak	Znak DX	Znak DY
I (NE)	0 - 400	$Az = Fi$	+	+
II (SE)	100-200	$Az = 200 - Fi$	-	+
II (SW)	200-300	$Az = 200 + Fi$	-	-
III (NW)	300-400	$Az = 400 - Fi$	+	-

## Algorytmy obliczania azymutów – wektor PK (początek P, koniec K)

### I metoda

1. Przyrosty współrzędnych odcinka PK:

$$\Delta X_{PK} = X_K - X_P$$

$$\Delta Y_{PK} = Y_K - Y_P$$

2. Uwzględnienie znaków przyrostów współrzędnych

jeżeli  $\Delta X_{PK} > 0$  oraz  $\Delta Y_{PK} > 0$  to  $A_{PK} = \arctg \frac{\Delta Y_{PK}}{\Delta X_{PK}}$

jeżeli  $\Delta X_{PK} > 0$  oraz  $\Delta Y_{PK} < 0$  to  $A_{PK} = \arctg \frac{\Delta Y_{PK}}{\Delta X_{PK}} + 400^g$

jeżeli  $\Delta X_{PK} < 0$  to  $A_{PK} = \arctg \frac{\Delta Y_{PK}}{\Delta X_{PK}} + 200^g$

### 3. Przypadki szczególne, gdy kierunek wektora PK jest równoległy do osi układu współrzędnych

$$\begin{aligned} \text{a) } & \text{gdy } \Delta X_{PK} = 0 \text{ i } \left\{ \begin{array}{l} \Delta Y_{PK} > 0 \text{ to } A_{PK} = 100^{\text{g}} \\ \Delta Y_{PK} < 0 \text{ to } A_{PK} = 300^{\text{g}} \end{array} \right\} \\ \text{b) } & \text{gdy } \Delta Y_{PK} = 0 \text{ i } \left\{ \begin{array}{l} \Delta X_{PK} > 0 \text{ to } A_{PK} = 0^{\text{g}} \\ \Delta X_{PK} < 0 \text{ to } A_{PK} = 200^{\text{g}} \end{array} \right\} \end{aligned}$$

Azymut odcinka (wektora) KP =  $A_{PK} + 200^{\text{g}}$

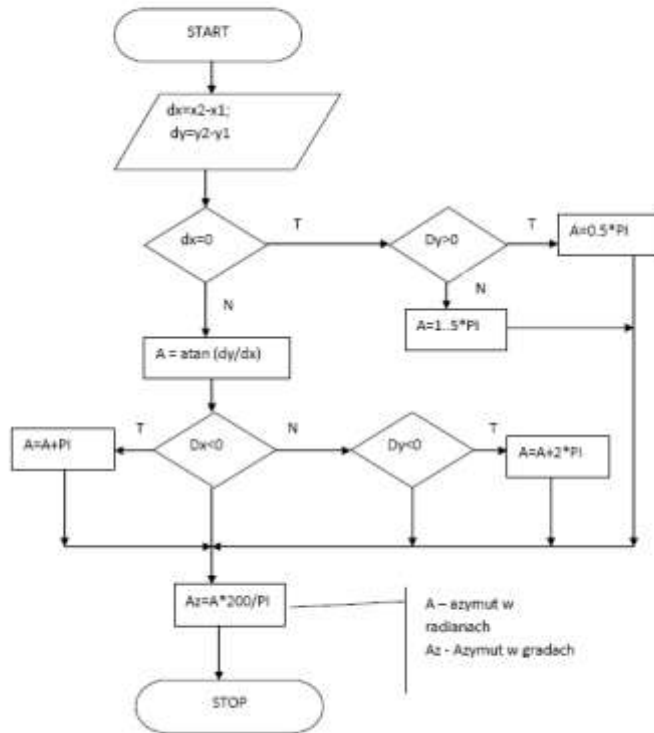
**Pseudokod funkcji obliczenia azymutu w wersji angielskiej** ( if- jeśli, else – w przeciwnym razie

```
{
Pi=4*arctg(1); RG=200/pi; RS=180/pi // Pi, RO[grad], RO[stop]
dx=x2-x1; dy=y2-y1 // przyrosty
if dx=0 then {if dy>0 then a=pi/2 else a=1.5*pi;} // azymut w radianach dla dx=0
else
{ if dx<0 then a=a+pi else a=a+2*pi} // azymut w radianach dla sx<>0
az=a*rg;
return az // zwróć wartość azymutu w gradach
}
```

#### Zapis w programie

```
double az1(double x1, double y1, double x2, double y2)
{
double pi, rg, rs, dx, dy, d, a, az ;
pi = 4.0 * atan(1.0); rg = 200.0 / pi; rs = 180.0 / pi;
dx=x2-x1; dy=y2-y1;
if (dx==0)
{
if (dy>0) a = pi / 2; else a = 1.5 * pi;
} // END IF dx==0
else // if (dx != 0)
{
a = atan(dy / dx);
if (dx < 0) a = a + pi;
else
{ if (dy < 0) a = a + 2 * pi; }
}
az = a * rg; // obliczenie azymutu w gradach
printf ("Azymut w az1 = %10.4lf [grad]\n", az);
return az;
}
```

SCHEMAT BLOKOWY OBLICZENIA AZYMUTU  
 na podstawie danych współrzędnych 2 punktów : początku wektora P(x1, y1) i końca K(x2, y2)



## II Metoda – wykorzystanie funkcji atan2(dy, dx) języka C i $M\_PI = PI$

200/M\_PI – RO[gradowe]

$$\Delta X_{PK} = dx = x_2 - x_1; \quad \Delta Y_{PK} = dy = y_2 - y_1;$$

Zapis w języku C++

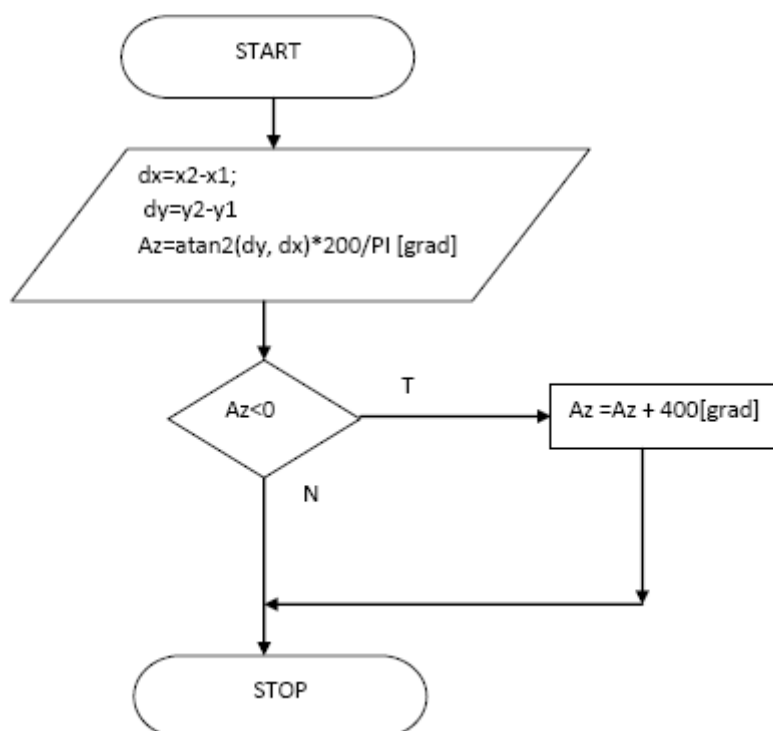
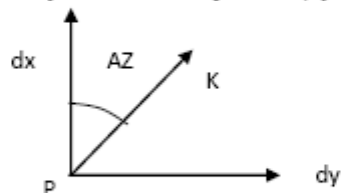
```

double az2(double x1, double y1, double x2, double y2)
{
    double dx, dy, result; double PI=M_PI;
    dx=x2-x1; dy=y2-y1;
    result = atan2(dy,dx) * 200.0/M_PI;
    if (result <0) result+=400.0;
    printf("Azymut w az2 = %f\n", result);
    return result;
}
    
```

## SCHEMAT BLOKOWY OBLICZENIA AZYMUTU

na podstawie danych współrzędnych 2 punktów:  
początku wektora P(x1, y1) i końca K(x2, y2)

Metoda z wykorzystaniem funkcji atan2(dy,dx)



### III Metoda – tradycyjne wykorzystanie czwartaków

Uwzględnienie ćwiartek układu współrzędnych – w – ćwiartka (1 ... 4)

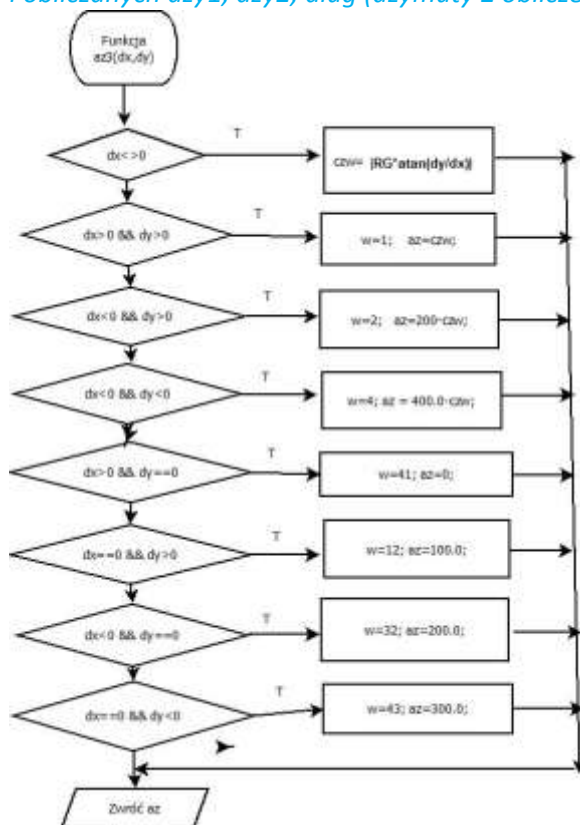
w=41 – granica ćwiartki 4 i 1; w=12 – granica ćwiartki 1 i 2 itd.

Zapis w języku C np. postaci:

```
if (dx>0 && dy>0) {w=1; az = czw; break; } // I ćwiartka  
if (dx<0 && dy>0) {w=2; az = 200.0-czw; break; }  
if (dx<0 && dy<0) {w=3; az = 200.0+czw; break; }  
if (dx>0 && dy<0) {w=4; az = 400.0-czw; break; } // IV ćwiartka  
if (dx>0 && dy==0) {w=41; az=0; break; } // granica ćwiartek 4 i 1  
if (dx==0 && dy>0) {w=12; az=100.0; break; }  
if (dx<0 && dy==0) {w=32; az=200.0; break; }  
if (dx==0 && dy<0) { w=43; az=300.0; break; // granica ćwiartek 4 i 3
```

W programie zastosowana struktura pkty -

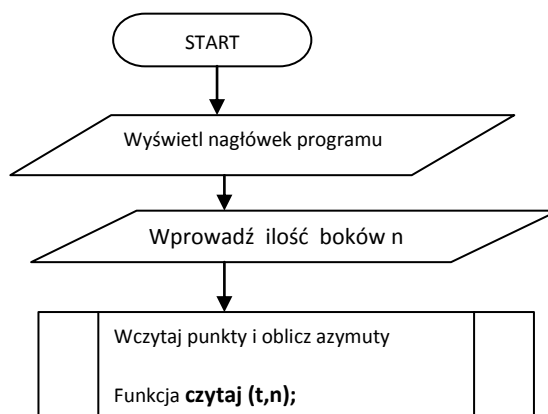
zawiera zestawienie boków - danych : BRP NRK x1 y1 x2 y2  
i obliczanych azy1, azy2, dlug (azymuty 2 obliczenia i długości)

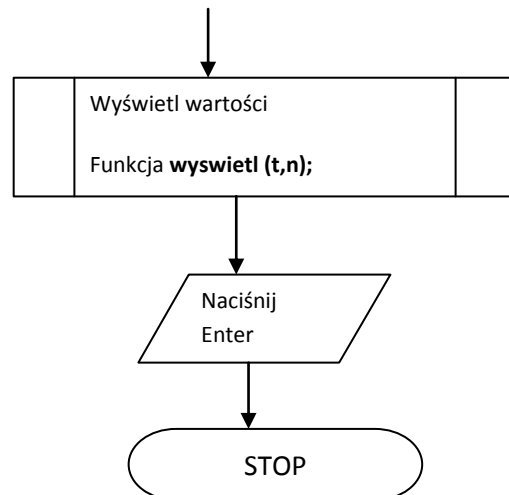


### Zapis algorytmu programu głównego w postaci listy kroków

1. Wyświetl nagłówek programu
2. Wprowadź ilość boków n
3. Zastosuj funkcję **czytaj**(t,n) – gdzie t – tablica, n – ilość punktów:  
Czytaj kolejno wykaz punktów dla podanej ilości boków: Nr1 x1 y1 oraz Nr2 x2 y2,  
Oblicz azymuty dwoma funkcjami i wprowadź dane i wyniki do tablicy t
4. Zastosuj funkcję **wyświetl**(t,n) do wyświetlenia wyników i zapisu do pliku

### SCHEMAT BLOKOWY PROGRAMU GŁÓWNEGO





### Instrukcje i funkcje funkcji głównej

```

int MAX=50; int n; pkty t[MAX]; // maksymalna ilość boków
system("cls");
puts("Program az_tabl.cpp - wersja języka C\n");
puts("Obliczenie azymutów i długości ze współrzędnych - tablica struktur, petla for\n");
puts("Numery punktów całkowite typu long int - do 2147483648\n");
printf("Wprowadź ilość boków: "); scanf("%d",&n); fflush(stdin);
czytaj (t,n); // funkcja do wczytania danych i obliczeń
wyświetl (t,n); // wyświetlenie i wydruk wyników
getch();
  
```

### Funkcje dodatkowe:

#### Definicje funkcji dodatkowych

```

// azymut ze współrzędnych - z zastosowaniem atan()
double az1(double x1, double y1, double x2, double y2)
{
double pi, rg, rs, dx, dy, d, a, az ;
pi = 4.0 * atan(1.0); rg = 200.0 / pi; rs = 180.0 / pi;
dx=x2-x1; dy=y2-y1;
if (dx==0)
{
if (dy>0) a = pi / 2; else a = 1.5 * pi;
} // END IF dx==0
else // if (dx != 0)
{
a = atan(dy / dx);
if (dx < 0) a = a + pi;
else
{ if (dy < 0) a = a + 2 * pi; }
}
az = a * rg; // obliczenie azymutu w gradach
printf ("Azymut w az1 = %10.4lf [grad]\n", az);
return az;
}
  
```

#### *// obliczenie azymutu z atan2(dy,dx)*

```

double az2(double x1, double y1, double x2, double y2)
{
  
```

```
double dx, dy, result; double PI=M_PI;
dx=x2-x1; dy=y2-y1;
result = atan2(dy,dx) * 200.0/M_PI;
if (result <0) result+=400.0;
printf("Azymut w az2 = %f\n", result);
return result;
}

// odleglosc ze wspolrzednych
double odl(double x1, double y1, double x2, double y2)
{
double dx, dy, result;
dx=x2-x1; dy=y2-y1; result = sqrt(dx*dx+dy*dy);
printf("Dlugosc = %f\n", result);
return result;
}
```