

## Wstęp

Rozpakuj projekt. Otwórz plik `projekt.sln`. W otwartym projekcie są następujące pliki: - `main.cpp` — główny plik z kodem. Tu piszemy nasz program - `winbgiz.cpp` — plik z funkcjami graficznymi - `winbgiz.h` — plik z definicjami funkcji graficznych

**Pamiętaj:** Często kompiluj projekt i patrz, czy wszystko działa!

## Pierwsze kreski

Wewnątrz funkcji `main` wpisz:

```
graphics( 200, 200);  
line( 0, 0, 200, 200);  
line( 100, 100, 100, 0);  
wait();
```

**Uwaga:** Zawsze pamiętaj o średnikach!

Skompiluj i uruchom projekt. Pierwsza linia tworzy okno grafiki, dwie następne rysują linie, zaś ostatnia czeka z zamknięciem okna na naciśnięcie dowolnego klawisza.

## Ćwiczenia

Używając funkcji `line(x1,y1,x2,y2)` i `circle(x,y,r)`, wykonaj następujące zadania:

- Zidentyfikuj, jak ułożony jest układ współrzędnych (X,Y) w oknie.
- Narysuj kwadrat.
- Narysuj ludzika.
- Narysuj koła olimpijskie.

## Zmienne

Pewne powtarzające się parametry (jak pozycję, promień, itp), możemy zastąpić zmiennymi. Następnie z nich wyliczyć odpowiednie współrzędne np:

```
int r,h;  
h = 100;  
r = 50;  
line( 10, 0, 0, h);  
line( 10, 0, 2*r, h);  
circle( 10+r, h, r);
```

Możemy używać wszelkich działań i funkcji matematycznych: +, -, \*, /, `sin()`, ....

**Pamiętaj:** Pierwsza linia deklaruje zmienną. Trzeba zadeklarować wszystkie zmienne, których będziesz używać! (szczegółowo, na kolejnych zajęciach).

Zauważ, że wartość zmiennej jest nadpisywana, więc możemy napisać:

```
int w;  
w = 50;  
circle( 10, w, 10);  
w = w + 20;  
circle( 10, w, 10);  
w = w + 20;  
circle( 10, w, 10);  
w = w + 20;  
circle( 10, w, 10);
```

W efekcie wyświetlą się cztery kółka narysowane koło siebie. Przetestuj.

## Ćwiczenia

Każdy program przetestuj dla paru ustawień zmiennych, by zobaczyć czy działa poprawnie. - Napisz program, który dla zmiennych `x,y,s`, tworzy okno o rozmiarach `x,y` i na środku narysowuje koło o promieniu `s`. - Dla zmiennej `d`, narysuj cztery dotykające się koła o średnicy `d` w prawym górnym rogu okna. - Dla zmiennej `y` narysuj koła olimpijskie w odległości `y` od górnej krawędzi. - Skopiuj poprzedni kod trzy razy i w każdym fragmencie zmodyfikuj wartość zmiennej `y`.

## Pętle

Pierwszą automatyzacją są pętle. Pętla wykonuje pewną operację, dopóki pewien warunek jest spełniony. Np:

```
int x;  
x = 0;  
while (x < 200) {  
    line(x,10,x,190);  
    x = x + 10;  
}
```

Taki program będzie wykonywany w następujący sposób: - wpisujemy 0 do zmiennej  $x$  - sprawdzamy, czy  $x < 200$  - rysujemy linię - zwiększamy zmienną  $x$  o 10 - i znów: sprawdzamy, czy  $x < 200$  - rysujemy linię - zwiększamy zmienną  $x$  o 10 - sprawdzamy, czy  $x < 200$  - rysujemy linię - ... - gdy wreszcie  $x$  przekroczy 200, pętla się skończy i program pójdzie dalej.

Ostatecznie program narysuje pionowe kreski dla kolejnych  $x = 0, 10, 20, \dots$ .

**Zauważ:** Program nie narysuje linii dla  $x = 200$ , bo komputer najpierw sprawdzi, że  $x \not< 200$  i przerwie pętlę.

## Ćwiczenia

- Napisz program, który narysuje kratkę z odstępem  $w$
- Narysuj rząd stycznych do siebie kół o promieniu  $r$ , zaczynając od lewej strony. Przemyśl: jeśli  $x$  to pozycja środka koła, to jaka wartość powinna jej być przypisana przed pętlą, o ile powinna być zwiększana i jaki warunek musi spełniać, by nie rysować poza oknem?!
- Pisząc jedną pętlę w drugiej, wypełnij cały obrazek przylegającymi kółkami.
  - Czy da się je lepiej upakować?
- Narysuj rząd kółek, których promienie zmniejszają się jak  $\frac{1}{n}$ .