



OPIS BIBLIOTEKI GRAFICZNEJ

Poniższy dokument krótko opisuje używane na laboratorium funkcje biblioteki graficznej `winbgi2.h`.

Funkcje

`graphics(int, int)` - funkcja tworzy okno graficzne o wymiarach w pikselach podanych w jej wywołaniu jako pierwszy i drugi argument. Funkcję wywołujemy zasadniczo raz w programie, w momencie, w którym ma zostać utworzone okno graficzne.

`wait()` - funkcja zatrzymuje działanie programu (sprawia, że np. okno graficzne nie znika automatycznie tuż po wykonaniu wszystkich operacji). Należy nacisnąć dowolny klawisz - wówczas okno graficzne zniknie, a program zacznie się dalej wykonywać. Przykład użycia obu funkcji:

```
void main()
{
    graphics(300, 200); // utworzy okno o szerokosci 300
                       // i wysokosci 200 pikseli

    // Tu wykonaj opieracje

    wait(); // Po naciśnięciu dowolnego przycisku okno zniknie
           // a program wykona poniższe instrukcje

    printf("Tu sie wykonuja kolejne operacje\n");
}
```

`line(int x1, int y1, int x2, int y2)` - funkcja rysująca w oknie graficznym linię o początku w punkcie o współrzędnych (x_1, y_1) i końcu w punkcie (x_2, y_2) . Należy przy tym pamiętać, że na ekranie początek układu współrzędnych znajduje się w lewym górnym rogu, oś x przebiega w prawo, zaś oś y w dół!

`circle(int x, int y, int R)` - funkcja rysująca w oknie graficznym okrąg o środku w punkcie o współrzędnych (x, y) i promieniu R . Ponadto warto zauważyć, że jako argumenty funkcji `line` i `circle` można przekazać zmienne typu zmiennoprzecinkowego (`float` lub `double`). W takim przypadku wartości te zostaną po prostu zrzutowane na typ `int` i wyświetlone z

nieco mniejszą dokładnością, niemniej w przypadku samego rysowania część dziesiątą tych liczb nie jest zwykle dla nas tak istotna. Przykład użycia obu funkcji:

```
void main()
{
    graphics(300, 200);

    // Rysujemy "celownik" - kolko z krzyzykiem w srodku
    line(150, 50, 150, 150);
    line(100, 100, 200, 100);
    circle(150, 100, 50);

    wait();
```

`clear()` - funkcja czyści okno graficzne ze wszystkich narysowanych w nim figur, jednak pozostawia samo okno graficzne.

`animate(int t)` - funkcja spowalnia działanie pętli. Dzięki temu łatwo jest osiągnąć efekt animacji, która zachodzi z określoną prędkością. Nie jest tak, że wszystkie komendy są wykonywane momentalnie - tak szybko, jak komputer potrafi je wykonać. Przykład użycia zobrazowany jest poniżej. Argument funkcji `animate` pozwala zmieniać tempo wykonania pętli.

```
void main()
{
    int maxiter = 100;
    double t = 0;

    graphics(300, 200);

    while(animate(100))
    {
        // Program animuje kolko ruszajace sie w prawo i lewo
        clear();
        circle(sin(t), 150, 20);
        t += 0.05;
        if(t>10)
            break;
```



```
}  
    wait();  
}
```

Na koniec zachęcamy jeszcze do samodzielnego zajrzenia do pliku `winbgi2.cpp`. Można w nim znaleźć wiele definicji najróżniejszych funkcji, jak np. `rectangle`, `pieslice` czy też funkcji do obsługi myszki. Zachęcamy do eksperymentowania.