

Подключение библиотеки и создание графического окна

Все функции для работы с библиотекой находятся в файле `graphx.h`.

Перед вызовом функций рисования объектов необходимо открыть графическое окно с помощью функции `ConnectGraphX()`.

Для корректного завершения работы программы, по окончании работы с графическим окном необходимо вызвать функцию `CloseGraphX()`.

`int ConnectGraphX(int width, int height)` – создание графического окна и подключение к нему. Параметр `width` задаёт ширину окна в пикселях, параметр `height` задаёт высоту окна в пикселях. При корректном завершении возвращает 0.

`void CloseGraphX()` – закрытие окна и подключения.

Листинг 1 – Пример простой программы

```
#include <graphx.h>
#include <stdlib.h>

int main() {
    ConnectGraphX(800, 600);
    sleep(5);
    CloseGraphX();
    return 0;
}
```

Рисование

Все функции рисования содержат параметр `color`. Данный параметр является параметром по умолчанию, и его необязательно задавать при вызове функции. При вызове функции без указания значения этого параметра фигура будет нарисована цветом, который был использован до этого. При указании параметра `color` текущий цвет рисования будет изменён для всех последующих выводов, как если бы была вызвана функция `SetForeground(color)`.

`void Pixel (int x, int y, long int color)` – рисует точку с координатами (x, y)

`void Line (int x1, int y1, int x2, int y2, long int color)` – рисует линию с началом в точке с координатами (x1, y1) и концом в точке с координатами (x2, y2).

`void Polyline (XPoint *points, int n, long int color)` – рисует ломаную линию, задаваемую массивом точек `XPoint` размером `n`. `XPoint` – структура, описывающая координаты точки и состоящая из двух полей `x` и `y`, имеющих тип `short int`.

Листинг 2 – Пример использования `Polyline()`

```
XPoint *points;
points = new XPoint[4];
points[0].x = 100;
points[0].y = 100;
points[1].x = 300;
```

```
points[1].y = 100;  
points[2].x = 300;  
points[2].y = 300;  
points[3].x = 100;  
points[3].y = 100;
```

```
Polyline(points, N);
```

`void Rectangle(int x, int y, unsigned int width, unsigned int height, long int color)` – рисует прямоугольник, где левый верхний угол находится в точке с координатами (x, y), с шириной прямоугольника width и высотой height.

`void Ellipse(int x, int y, unsigned int width, unsigned int height, long int color)` – рисует эллипс с центром в точке с координатами (x, y), шириной width и высотой height.

`void Arc(int x, int y, int width, int height, int angle1, int angle2, long int color)` – рисует эллиптическую дугу с центром в точке с координатами (x, y), шириной width и высотой height из положения angle1 до angle2. Углы angle1 и angle2 указываются в градусах.

`void Text(int x, int y, char *string, int length, long int color)` – рисует текст string длиной length с началом в точке с координатами (x, y).

Работа с цветом

Для обозначения цветов в библиотеке используется тип long int. Чтобы преобразовать тройку цветовых компонент (красный, зелёный, синий) в конкретный цвет, используется функция **unsigned long RGB (unsigned char R, unsigned char G, unsigned char B)**, значения параметров изменяются от 0 до 255.

`void SetForeground(long int color)` – задаёт цвет рисуемых линий.

`void SetBackground(long int color)` – задаёт цвет фона окна.

В отличие от функций описанных в разделе **Рисование**, для функций FillRectangle(), FillArc(), FillEllipse(), FillPolygon() параметр color является обязательным. При указании параметра color текущий цвет рисования будет изменён для всех последующих выводов, как если бы была вызвана функция SetForeground(color).

`void FillRectangle(int x, int y, unsigned int width, unsigned int height, long int color)` – заполняет цветом color прямоугольную область шириной width и высотой height, где левый верхний угол находится в точке с координатами (x, y).

`void FillArc(int x, int y, int angle1, int angle2, long int color)` – заполняет цветом color область описываемую эллиптической дугой с центром в точке с координатами (x, y), шириной width и высотой height из положения angle1 до angle2. Углы указываются в градусах.

`void FillEllipse(int x, int y, unsigned int width, long int color)` - заполняет цветом `color` эллипс с центром в точке с координатами `(x, y)`, шириной `width` и высотой `height`.

`void FillPolygon(XPoint *points, int n, long int color)` – заполняет цветом `color` многоугольник, задаваемый массивом точек `XPoint` размером `n`. `XPoint` – структура, описывающая координаты точки и состоящая из двух полей `x` и `y`, имеющих тип `short int`.

Листинг 3 – Пример использования `FillPolygon()`

```
XPoint *points;
points = new XPoint[4];
points[0].x = 100;
points[0].y = 100;
points[1].x = 300;
points[1].y = 100;
points[2].x = 300;
points[2].y = 300;
points[3].x = 100;
points[3].y = 100;

FillPolygin(points, N, RGB(255, 0, 0));
```

Управление графическим окном

`void Clear()` – очищает содержимое графического окна.

Ввод с клавиатуры

`int InputChar()` – возвращает номер введенной клавиши типа `int`. На время действия данной функции работа программы приостанавливается до первого ввода с клавиатуры.

Компиляция

После установки библиотеки в систему программа компилируется с параметром линковщика `-lgraphx`.

Пример:

```
g++ -o main main.cpp -lgraphx
```