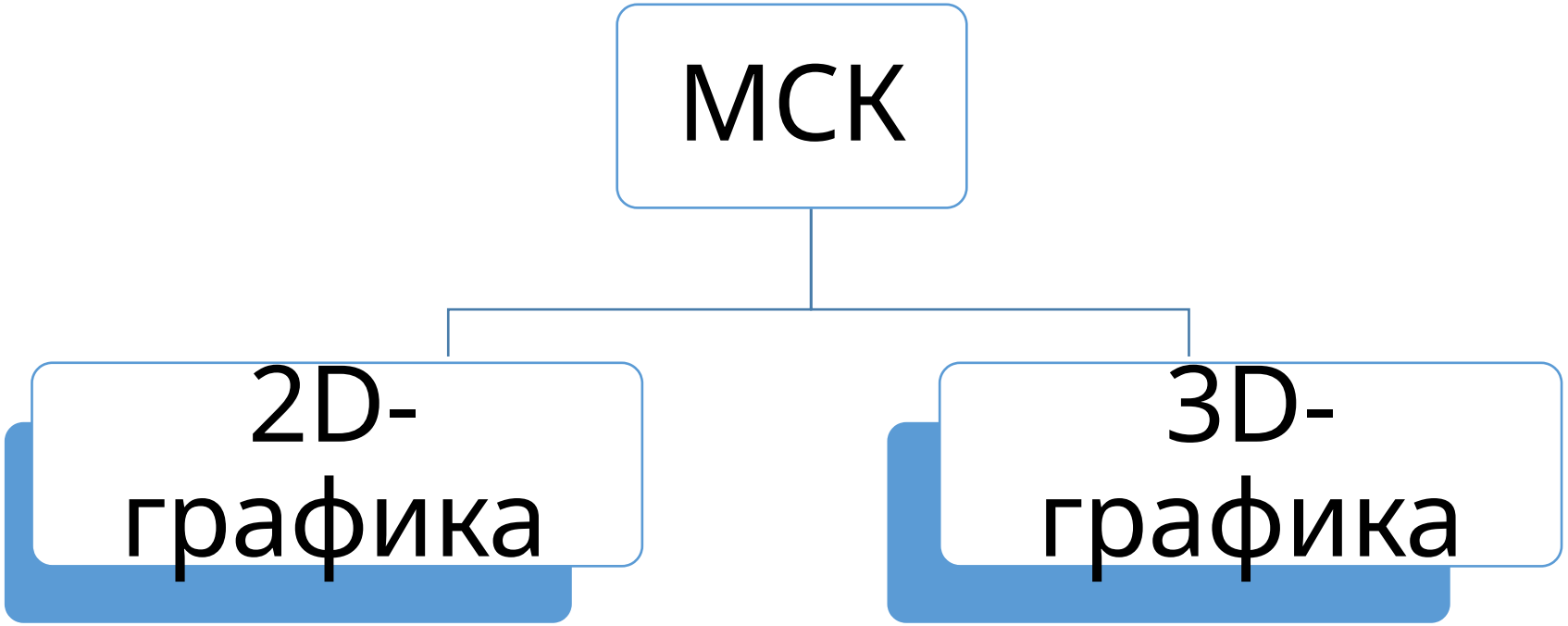
Создание программ построения графиков с использованием графических примитивов

Мировые и экранные координаты

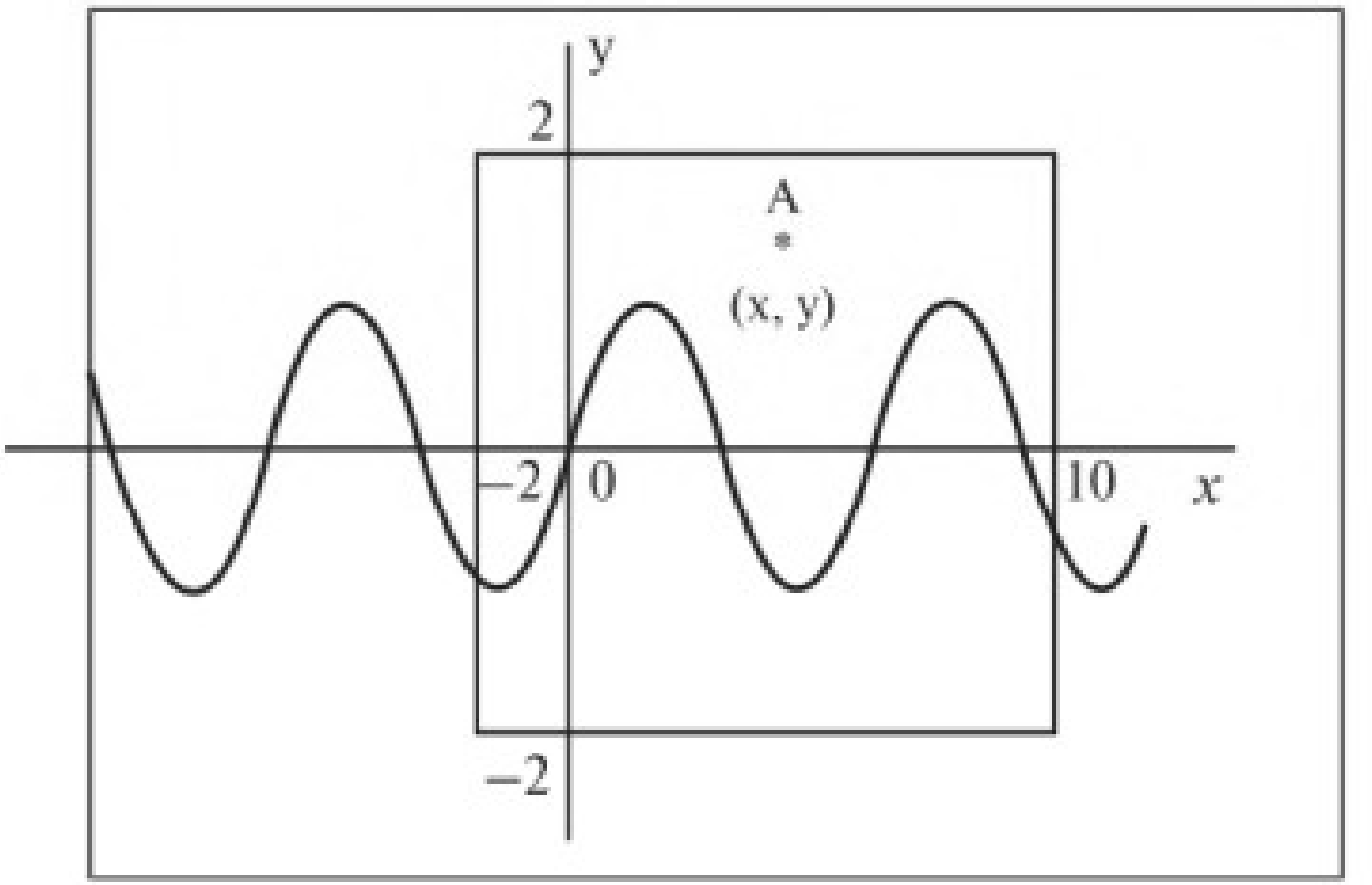
Вне зависимости от природы процесса, каждая точка графика характеризуется двумя числами, координатами этой точки. При отображении двумерных картин на экране компьютера используются приемы и алгоритмы 2D-графики. Независимо от реального смысла координат, двумерные координаты обозначаются как (x, y)

Мировые и экранные координаты

Координатная система, в которой описываются реальные объекты или процессы – это мировая система координат.



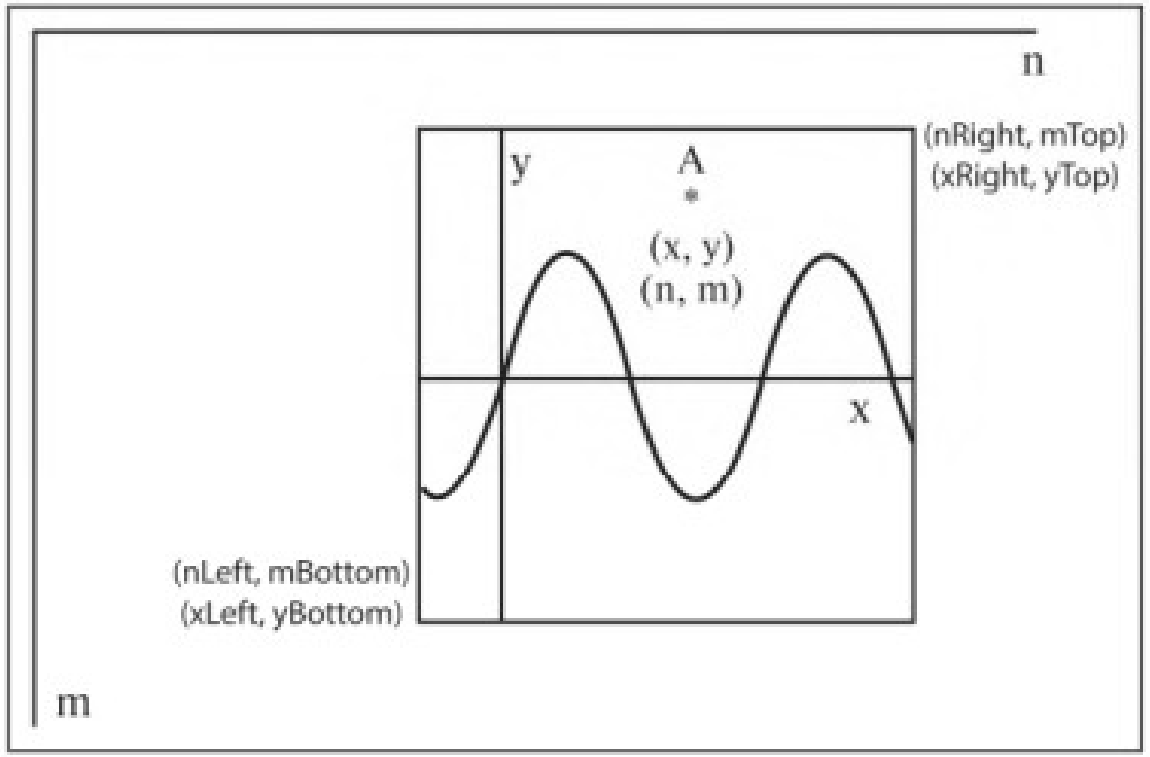
Y=sin(x) в мировой системе координат



В МСК поле ввода определяется, задав координаты левого нижнего угла (xLeft, yBottom) и координатой правого верхнего угла (xRight, yTop).

На рисунке:
xLeft=-2
xRight=10
yBottom=-2
yTop=2

Экранная система координат



ЭСК связана с устройством вывода – экраном компьютера, принтером и т.д.

Начало ЭСК – (0, 0) – верхний левый угол экрана.

Ось n направлена слева направо, ось m – сверху вниз.

Единица измерения – пиксель.

Поле вывода может располагаться в любой точке экрана, задав в ЭСК координаты левого нижнего угла (nLeft, mBotton) и координаты правого верхнего угла (nRight, mTop)

Сопоставление координат точки A в МСК (x,y) и ЭСК (n,m)

$$N=(x-xLeft)/(xRight-xLeft)*(nRight-nLeft)+nLeft;$$

$$M=(y-yBottom)/(yTop-yBottom)*(mTop-mBottom)+mBottom;$$

Т.е. мировые координаты (x,y) точки A будут изменяться в пределах:

$$xLeft \le x \le xRight,\ yBottom \le y \le yTop$$

экранные координаты (n,m) точки A будут изменяться в пределах:

$$nLeft \le n \le nRigh,\ mBottom \le m \le mTop$$

Сопоставление координат точки A в МСК (x,y) и ЭСК (n,m)

Соотношения, представленные на предыдущем слайде оформляют в виде двух inline-функций xn() и ym().

Inline-функция – это функция, тело которой подставляется в каждую точку вызова вместо того, чтобы генерировать код.

Фрагмент кода

```
//переход от x к пикселю n
inline int xn(double x)
{
  return (int)((x - xLeft)/(xRight - xLeft)*(nRight -
- nLeft)) + nLeft;
}

//переход от y к пикселю m
inline int ym(double y)
{
  return (int)((y - yBottom)/(yTop - yBottom)*(mTop -
- mBottom)) + mBottom;
}
```

Программа рисования синусоиды

Программа состоит из :

Главной функции WinMain(),

Функции главного окна WndProc() – сюда приходят сообщения

Функции Line_Paint() – рисование синусоиды

Функции xn()

Функции ym()

Используются для перевода мировых координат в экранные (пиксели)

Библиотеки:

```
#include <windows.h>
#include <math.h>
#include <tchar.h>
```

Прототипы функций

```
//прототипы функции
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
void Line_Paint(HWND);
inline int ym(double);
inline int xn(double);
```

Описание главной функции WinMain()

```
TCHAR cname[] = _T("Class"); //имя класса окна
TCHAR title[] = _T("Sinusoid"); //заголовок окна

//главная функция
int WINAPI WinMain(HINSTANCE hInstance,
        HINSTANCE hPrevInstance,
        LPSTR lpCmdParam,
        int nCmdShow)
{
  //регистрация класса окна приложения
  WNDCLASS wc;

  wc.style = 0;
  wc.lpfnWndProc = (WNDPROC)WndProc; //функция окна
  wc.cbClsExtra = 0;
  wc.cbWndExtra = 0;
  wc.hInstance = hInstance; //дескриптор приложения
  wc.hIcon = LoadIcon(NULL, IDI_APPLICATION);
  wc.hCursor = LoadCursor(NULL, IDC_ARROW);
  wc.hbrBackground = (HBRUSH)GetStockObject(LTGRAY_
BRUSH);
  wc.lpszMenuName = NULL;
  wc.lpszClassName = cname; //имя класса окна

  if(!RegisterClass(&wc)) //регистрация класса окна
приложения
    return 0;
```

Описание главной функции WinMain()

```
//создание окна приложения
 HWND hWnd; // дескриптор окна приложения

 hWnd = CreateWindow(cname, //имя класса окна
         title, //заголовок окна
         WS_OVERLAPPEDWINDOW, // стиль окна
         CW_USEDEFAULT, // x
         CW_USEDEFAULT, // y
         CW_USEDEFAULT, // Width
         CW_USEDEFAULT, // Height
         NULL, //дескриптор окна-родителя
         NULL, //дескриптор меню
         hInstance, //дескриптор приложения
         NULL);

if(!hWnd)
  return 0;

// рисуем окно
ShowWindow(hWnd,nCmdShow);
UpdateWindow(hWnd);
```

Описание главной функции WinMain()

```
  // запускаем цикл обработки сообщений
  MSG msg; // структура для работы с сообщениями

  while(GetMessage(&msg,NULL,0,0))
  {
    TranslateMessage(&msg);
    DispatchMessage(&msg); //посылает сообщение функции
WndProc()
  }

  return 0;
}
```

Комментарии по функции WinMain()

1. Регистрация класса окна приложения. Заполняются поля структуры wc (указывается имя функции окна WndProc, в которую будут приходить сообщения, дескриптор приложения hInstance, имя класса окна cname и другие параметры).
2. Создание окна приложения. Вызывается функция CreateWindow() с одиннадцатью параметрами (имя окна cname, заголовок окна title, дескриптор приложения hInstance). При удачном создании окна система Windows присваивает этому окну дескриптор hWnd.
3. Рисование окна приложения с помощью функции ShowWindow().
4. Запуск цикла обработки сообщения с помощью функции DispatchMessage(), которая посылает обработанные сообщения функции окна WndProc().

Описание функции окна WndProc()

```
//функция окна, принимающая сообщения
LRESULT CALLBACK WndProc(HWND hWnd, UINT message,
   WPARAM wParam, LPARAM lParam)
{
  switch(message)
  {
//сообщение при обновлении окна
case WM_PAINT:
  Line_Paint(hWnd); //функция рисования
  break;

//сообщение при закрытии окна
case WM_DESTROY:
  PostQuitMessage(0); //выход из цикла сообщений
  break;

//обработка сообщений по умолчанию
default:
  return DefWindowProc(hWnd,message,wParam,lParam);
}
return 0;
}
```

Описание функции Line_Paint()

```
 //размеры поля вывода в мировой системе координат
 double xLeft, xRight, yBottom, yTop;
//размеры поля вывода в пикселях в клиентской области
//окна приложения
int nLeft, nRight, mBottom, mTop;

//функция вызывается на сообщение WM_PAINT
void Line_Paint(HWND hwnd)
{
  //размеры окна в мировых координатах и в пикселях
  xLeft = -2; xRight = 10; yBottom = -2; yTop = 2;
  nLeft = 100; nRight = 450; mBottom = 350; mTop = 50;

  //создаем массивы точек для аргумента x и функции
y = sin(x)
  const int N = 50;
  double corX[N], corY[N];
  double x, y, dx = (xRight - xLeft)/(N-1);

  for(int i=0; i<N; i++)
  {
    x = xLeft + dx*i; y = sin(x);
    corX[i] = x; corY[i] = y;
  }

  HDC hdc; //дескриптор контекста устройства
  PAINTSTRUCT ps; //структура для работы контекста
```

Описание функции Line_Paint()

```
  //получаем контекст устройства <hdc> для окна <hwnd>
  hdc = BeginPaint(hwnd, &ps);

  HBRUSH hbrush, hbrushOld; //дескрипторы кистей
  HPEN hpen1, hpen2, hpen3, hpenOld; //дескрипторы
перьев

  //создаем кисть <hbrush1>, стиль – сплошной, цвет
– синий
  hbrush = CreateSolidBrush(RGB(0,0,200));
  //выбираем кисть <hbrush> в контекст устройства
<hdc>,
  //запоминаем дескриптор старой кисти <hbrushOld>
  hbrushOld = (HBRUSH)SelectObject(hdc,hbrush);

  //создаем перо <hpen1>, стиль – сплошной,
 // толщина 3 пикселя, цвет – ярко-желтый
   hpen1 = CreatePen(PS_SOLID,3,RGB(255,255,0));
   //выбираем перо <hpen1> в контекст устройства <hdc>,
   //запоминаем дескриптор старого пера <hpenOld>
   hpenOld = (HPEN)SelectObject(hdc,hpen1);

   //рисуем прямоугольник с границей
   Rectangle(hdc,nLeft,mBottom,nRight,mTop);
```

Описание функции Line_Paint()

```
//создаем перо <hpen2>, стиль – сплошной,
// толщина 1 пиксель, цвет – ярко-голубой
  hpen2 = CreatePen(PS_SOLID,1,RGB(0,255,255));
  //выбираем перо <hpen2> в контекст устройства <hdc>
  SelectObject(hdc,hpen2);

  int nb, ne, mb, me;
  POINT pt;

  //рисуем ось OX
  nb = xn(xLeft); mb = ym(0);
  MoveToEx(hdc, nb, mb, &pt);
  ne = xn(xRight); me = ym(0);
  LineTo(hdc,ne,me);

  //рисуем ось OY
  nb = xn(0); mb = ym(yBottom);
  MoveToEx(hdc, nb, mb, &pt);
  ne = xn(0); me = ym(yTop);
  LineTo(hdc,ne,me);
```

```
  //рисуем график по двум массивам

  //создаем перо <hpen3>, стиль – сплошной,
// толщина 2 пикселя, цвет – ярко-красный
  hpen3 = CreatePen(PS_SOLID,2,RGB(255,0,0));
  //выбираем перо <hpen3> в контекст устройства <hdc>
  SelectObject(hdc,hpen3);

  nb = xn(corX[0]); mb = ym(corY[0]);
  MoveToEx(hdc, nb, mb, &pt);
  for( int i=1; i<N; i++)
  {
    nb = xn(corX[i]); mb = ym(corY[i]);
    LineTo(hdc,nb,mb);
  }

//выбираем старую кисть <hbrushOld> в контекст
устройства <hdc>
  SelectObject(hdc,hbrushOld);
  DeleteObject(hbrush); //уничтожаем кисть <hbrush>

//выбираем старое перо <hpenOld> в контекст устройства
<hdc>
  SelectObject(hdc,hpenOld);
  DeleteObject(hpen1); //уничтожаем перо <hpen1>
  DeleteObject(hpen2); //уничтожаем перо <hpen2>
  DeleteObject(hpen3); //уничтожаем перо <hpen3>

//освобождаем контекст устройства <hdc> в окне <hwnd>
  EndPaint(hwnd, &ps);
}
```

Комментарий

Переход от мировых координат к экранным осуществляется с помощью вспомогательных функций xn() и ym(), их описание дается в конце файла программы.

Результат выполнения программы