**Лабораторная работа № 3.2**

**Тема:** Изучение аффинных преобразований на плоскости на примере объекта «Планеты»

I. Реализовать функции аффинных преобразований и добавить их в библиотеку Libgraph

**CMatrix CreateTranslate2D (double dx, double dy)**

// Формирует матрицу для преобразования координат **объекта** при его смещении

// на dx по оси X и на dy по оси Y в фиксированной системе координат

// --- ИЛИ ---

// Формирует матрицу для преобразования координат объекта при смещении начала

// системы координат на -dx оси X и на -dy по оси Y при фиксированном положении объекта

**CMatrix CreateRotate2D (double fi)**

// Формирует матрицу для преобразования координат **объект**а при его повороте

// на угол fi (при fi>0 против часовой стрелки) в фиксированной системе координат

// --- ИЛИ ---

// Формирует матрицу для преобразования координат объекта при повороте начала

// системы координат на угол -fi при фиксированном положении объекта

// fi - угол в градусах

II. Реализовать объект «Планеты» реализовать как класс CSunSystem.

class CSunSystem

{

CRect Sun; // Прямоугольник Солнца

CRect Earth; // Прямоугольник Земли

CRect Moon; // Прямоугольник Луны

CRect EarthOrbit; // Прямоугольник, описанный вокруг орбиты Земли

CRect MoonOrbit; // Прямоугольник, описанный вокруг орбиты Луны

CMatrix ECoords; // Текущие координаты Земли в СК Солнца ( x,y,1)

CMatrix MCoords; // Текущие координаты Луны в СК Земли ( x,y,1)

CMatrix MCoords1; // Текущие координаты Луны в СК Солнца ( x,y,1)

CMatrix PM; // Матрица поворота для луны

CMatrix PE; // Матрица поворота для Земли

CRect RW; // Прямоугольник в окне

CRectD RS; // Прямоугольник области в МСК

double wEarth; // Угловая скорость Земли относительно Солнца, град./сек.

double wMoon; // Угловая скорость Земли относительно Солнца, град./сек.

double fiE; // Угловое положение Земли в системе кординат Солнца, град

double dfiE; // Угол поворота Земли за время dt.

double fiM; // Угловое положение Луны в системе кординат Земли, град

double dfiM; // Угол поворота Земли за время dt

double dt; // Интервал дискретизации, сек.

public:

CSunSystem();

void SetDT(double dtx){dt=dtx;}; // Установка интервала дискретизации

void SetNewCoords(); // Вычисляет новые координаты планет

void GetRS(CRectD& RSX); // Возвращает область графика в мировой СК

CRect GetRW(){return RW;}; // Возвращает область графика в окне

void Draw(CDC& dc); // Рисование без самостоятельного пересчета координат

};

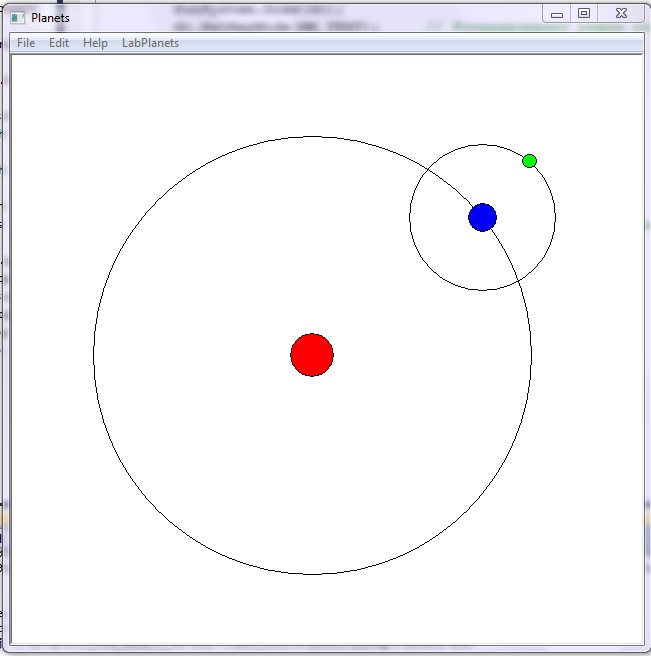
Добавить в класс CSunSystem планету Mars, орбита которой находится между Солнцем и Землей. Новая планета вращается в направлении, противоположном вращению Земли.

Рисование выполнять в режиме MM**\_ANISOTROPIC**. Толщина линии – **1пиксел**.

**Сценарий работы**

* После запуска приложения на экране появляется пустое окно.
* После выбора пункта меню «LabPlanets►Planets» на экране появляется статичное изображение планет, размер и положение которых определяются в соответствии с параметрами, заданными в конструкторе по умолчанию.
* После двойного щелчка ЛКМ (левая клавиша мыши) в области окна планеты приходят в движение.
* После двойного щелчка ПКМ (правая клавиша мыши) в области окна движение планет прекращается.

Размеры окна устанавливаются **по габаритам системы планет и** **не должны изменяться.**



Mars