

Задание: формирование траектории для движения простого геометрического объекта

Пусть гипоциклоида задается параметрическим уравнением:

$$\begin{cases} x = r(k-1) \left(\cos t + \frac{\cos((k-1)t)}{k-1} \right) \\ y = r(k-1) \left(\sin t - \frac{\sin((k-1)t)}{k-1} \right) \end{cases}$$

Обеспечить формирование изображения траектории и анимацию движения катящейся окружности.

Для обеспечения остановки по времени необходимо подключить библиотеку:

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

Для прорисовки окружностей и линий (область прорисовки для pictureBox1) необходимо в класс добавить функции:

```
private: void Paint_Circle(int cX,int cY,int centX,int centY,int radius,int x,int y)
{ Graphics^ Графика=pictureBox1->CreateGraphics();
  Графика->DrawEllipse(Pens::Black,centX+cX-radius,cY-radius-centY,radius*2,radius*2);
  Графика->DrawLine(Pens::Black,centX+cX,cY-centY,cX+x,cY+y);
  // прорисовка радиуса большей окружности
}
```

```
private: void Paint_Graphic(int cX,int cY,int r2,int x,int y,array<Point> ^p)
{ Graphics^ Графика=pictureBox1->CreateGraphics();
  Графика->Clear(BackColor);
  Paint_Circle(cX, cY, 0, 0, r2, x, y);
  Графика->DrawLines(Pens::Red, p); // траектория
}
```

```
double InitT=0, LastT=6.3; // оборот в 360 градусов (6,28 радиан)
```

```
double Step=0.1, angle=InitT;
```

```
double x, y, x1, y1;
```

```
int cX=120, cY=120; // центр большой окружности
```

```
int R2=90; // радиус большой окружности
```

```
int k=20; // число областей на траектории
```

```
int R1=int(R2/k); // радиус меньшей (движущейся) окружности
```

```
int i=0; // количество точек прорисовки
```

```
array<Point> ^p;
```

```
p=gcnew array<Point>(64); // точки для прорисовки (LastT/Step)
```

```
while(angle<=LastT)
```

```
{ x=R1*(k-1)*(cos(angle)+cos((k-1)*angle)/(k-1));
```

```
  y=R1*(k-1)*(sin(angle)-sin((k-1)*angle)/(k-1));
```

```
  p[i]=Drawing::Point(cX+int(x), cY+int(y)); // расчет очередной точки траектории
```

```
  Paint_Graphic(cX,cY,R2,x,y,p);
```

```
  x1=(R2-R1)*sin(angle+1.57);
```

```
  y1=(R2-R1)*cos(angle+1.57);
```

```
  Paint_Circle(cX, cY, int(x1), int(y1), R1, x, y);
```

```
  angle+=Step;
```

```
  ::Sleep(40); //время приостановки прорисовки
```

```
  i++;
```

```
}
```