Одеський національний університет імені І.І.Мечникова
(повне найменування вищого навчального закладу)

Інститут математики, економіки і механіки
(повне найменування інституту/факультету)

Кафедра теоретичної механіки
(повна назва кафедри)

Дипломна робота

бакалавр
(освітньо-кваліфікаційний рівень)

на тему: **«Дослідження руху механічної системи**

з двома ступенями вільності»

«Investigation of the mechanical system
with two degrees of freedom»

Виконала: студентка денної форми навчання
Маці юк А.О.
напряму підготовки
6.040202 Механіка
Керівник доцент Косой М.Б. _____

Рецензент доцент Рачинська А.Л. _____

Рекомендовано до захисту:
Протокол засідання кафедри
№ від 2017 р.

Завідувач кафедри д.ф.-м. наук, проф. Асланов С.К.
(підпис)

Захищено на засіданні ЕК №
протокол № від 2017 р.
Оцінка _____ / _____ / _____
(за національною шкалою, шкалою ECTS, бали)

Голова ЕК д. ф.-м..наук, проф. Лещенко Д.Д.
(підпис)

Одеса – 2017

ЗМІСТ

ВСТУП……...3

§1. Рівняння Лагранжу 2-го роду.....4

§2. Застосування рівняння Лагранжу до дослідження руху механічної системи з двома ступенями свободи.....7

§3. Чисельний метод Рунге-Кутта.....12

§4. Використання графічних можливостей мови C#.....16

§5. Процес комп'ютерного моделювання руху механічної системи з двома ступенями свободи.....25

ВИСНОВКИ……....31

ПЕРЕЛІК ПОСИЛАНЬ..... 32

Додаток 1. Зовнішній вигляд форми Form1…….....33

Додаток 2. Зовнішній вигляд форми Form2…….....34

Додаток 3. Зовнішній вигляд форми Form3…….....35

Додаток 4. Механізм в початковий момент часу.....36

Додаток 5. Механізм в поточний момент часу.....37

Додаток 6. Графік функції x=x(t).....38

Додаток 7. Графік функції x'=x'(t).....39

Додаток 8. Графік функції φ=φ(t).....40

Додаток 9. Графік функції φ'=φ'(t).....41

Додаток 10. Графік функції M=M(t).....42

Додаток 11. Графік функції P=P(t).....43

Додаток 12. Клас форми Form1…….....44

Додаток 13. Клас форми Form2….....52

Додаток 14. Клас форми Form3…….54

ВСТУП

Необхідно розробити **Windows-**додаток, який дозволить провести дослідження задачі руху механічної системи з двома ступенями свободи.

Користувач цим програмним продуктом, повинен мати можливість задавати різні динамічні характеристики тіл системи та задавати різні закони діючих силових факторів, що здійснюють рух системи.

Повинна існувати можливість моделювати руху механічної системи з двома ступенями свободи, при цьому користувач може повертати систему навколо осі вертикалі для більш детального дослідження.

Розрахунок закону руху системи може проводитися при різних початкових умовах, які задає користувач.

Можна моделювати різний вплив силових факторів на закони рух тіл системи за допомогою вибору різних законів силових факторів.

Розрахунок системи рівнянь руху механічної системи з двома ступенями свободи проводиться чисельно за допомогою методу Рунге-Кутта.

Крім того, додаток повинен мати функціональну можливість побудови графіків, що дозволить користувачу цим програмним продуктом, зробити необхідний аналіз та порівняння законів руху системи з різними вхідними параметрами задачі.

ВИСНОВКИ

У даній роботі розроблено **Windows-**додаток, який моделює рух системи твердих тіл з двома ступеня свободи. Для системи змодельована **3-D** анімація з використанням бібліотеки **DirectX9.0**. Модель конструювалася з готових **Mesh-**об'єктів. Для реальності картини колеса будувалися у каркасному вигляді, інші частини системи у вигляді суцільних об'єктів.

Користувач розробленого додатку може задавати різні значення для геометричних параметрів задачі: радіуси колес; динамічні характеристики: маси тіл, та діючий момент, який приводить колесо зі стрічкою транспортера у рух, а також діючу силу, що приводить у рух візок.

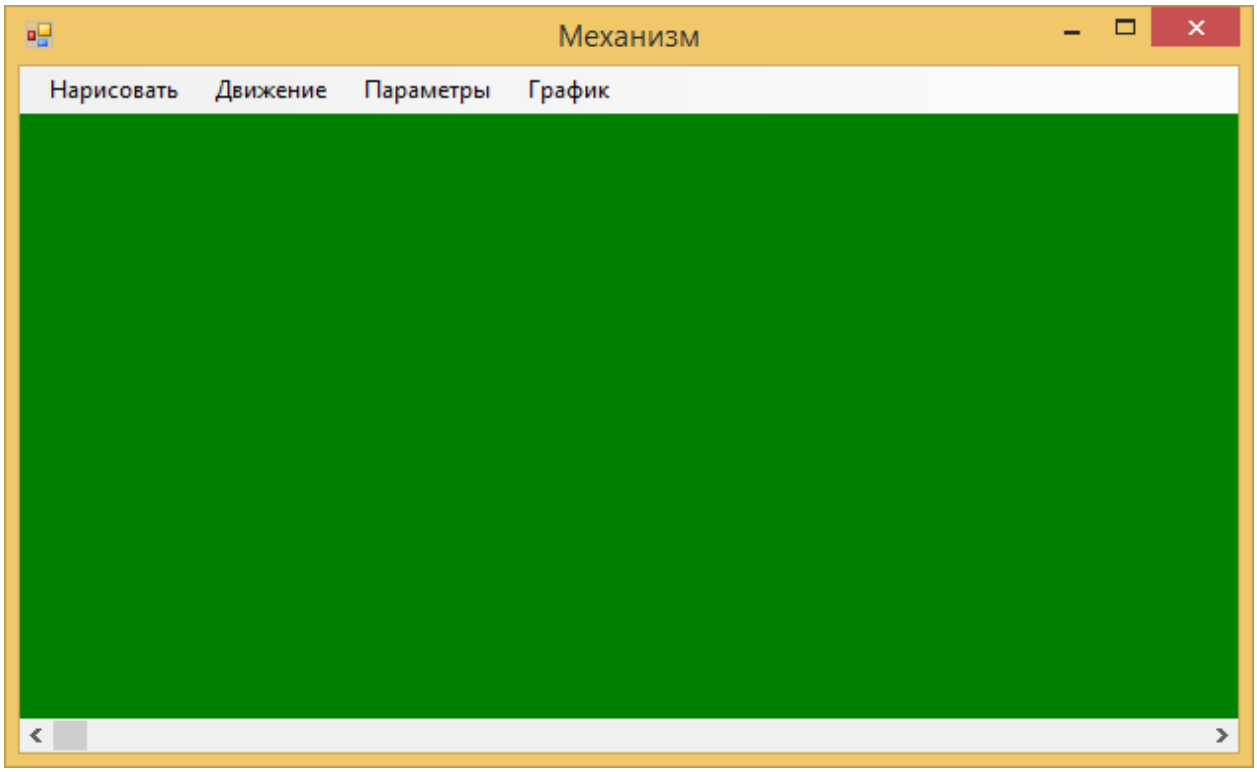
Змодельовано рух системи матеріальних точок з двома степенями свободи під дією моментів та сил різного характеру: постійної величини, періодичної функції часу, лінійної функції часу або квадратичної функції часу . За допомогою **3-D** анімації у побудованій віртуальній лабораторії дипломного додатку демонструється обчислений рух. Для повноти картини аналізу основних характеристик руху системи, існує функціональна можливість побудови графіків змін кута повороту колеса, кутової швидкості колес, відносного переміщення візка по стрічці транспортера та відносної швидкості руху візка. Крім того існує можливість побудови графіків діючих моментів та сил.

Windows-додаток має інтуїтивно понятий та функціонально зручний інтерфейс, що дозволяє його використовувати студентам під час вивчання курсу теоретичної механіки.

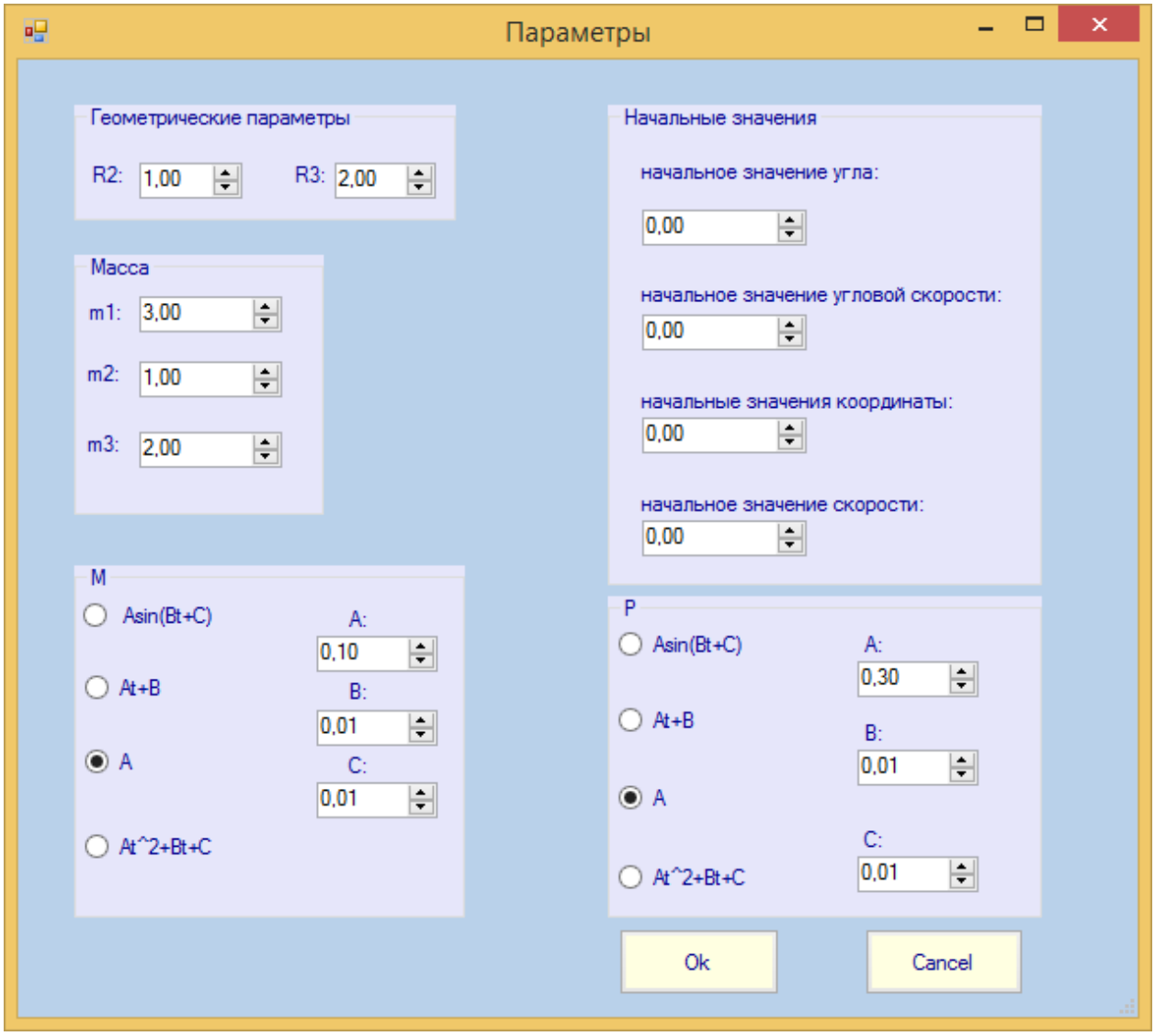
Перелік посилань

1. Том Миллер. DirectX 9 с управляемым кодом. Программирование игр и графика./ М.: Издательский дом «КомБук», 2005. – 395 с.
2. Шилдт Г. Полный справочник по C#. /. М.: Издаетльский дом «Вильямс». 2004 – 740 с.
3. Джесс Либерти. Программирование на C#./ 2-е издание.
4. Эндрю Троелсен. C# и платформа .NET. / Библиотека программиста/ Москва., Санкт-Петербург: Издательство «Питер». – 2004. – 785 с.
5. Культин Н.Б. C# в задачах и примерах. – БХВ.-Перетбург, 2007. – 240с. : с иллюстрациями.
6. Бухгольц Н.Н. Основной курс теоретической механики. Часть 1. Динамика системы материальных точек./Издание 4, переработанное и дополненное Таргом С.М// - М..:.Наука, 1965. – с. 183 -200
7. Яблонский А.А. Курс теоретической механики./ Издательство «Высшая школа» - 1966. – 520 с.
8. Калиткин Н.Н. Численные методы./ Математика/ - М. : Наука. 1978. – 512с.

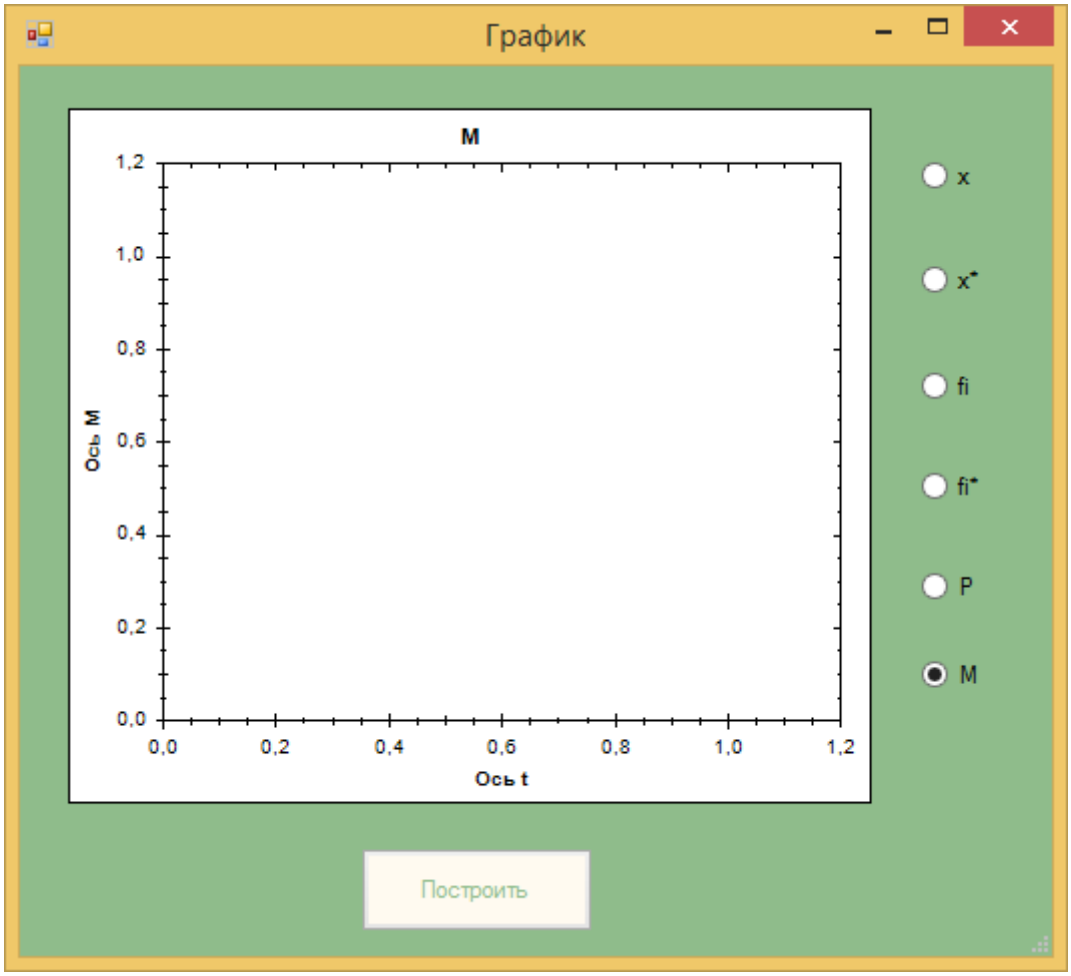
Додаток 1. Зовнішній вигляд форми Form1



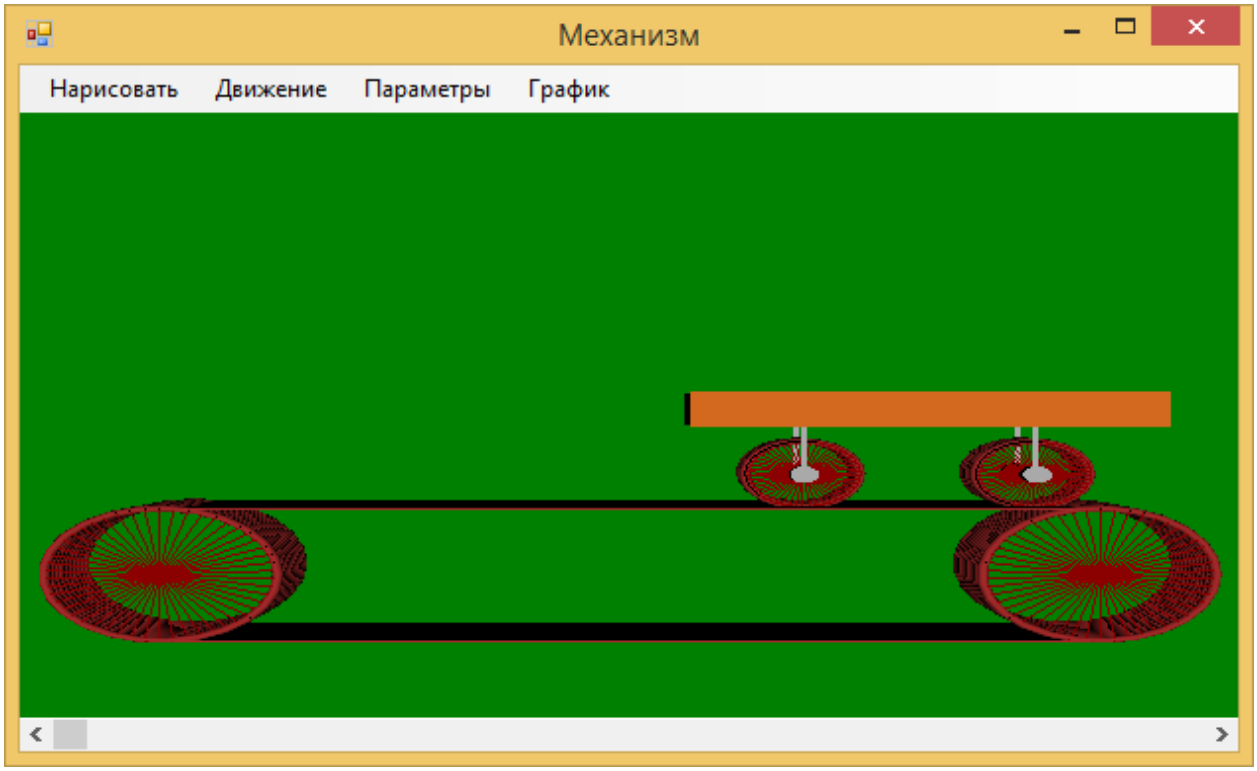
Додаток 2. Зовнішній вигляд форми Form2



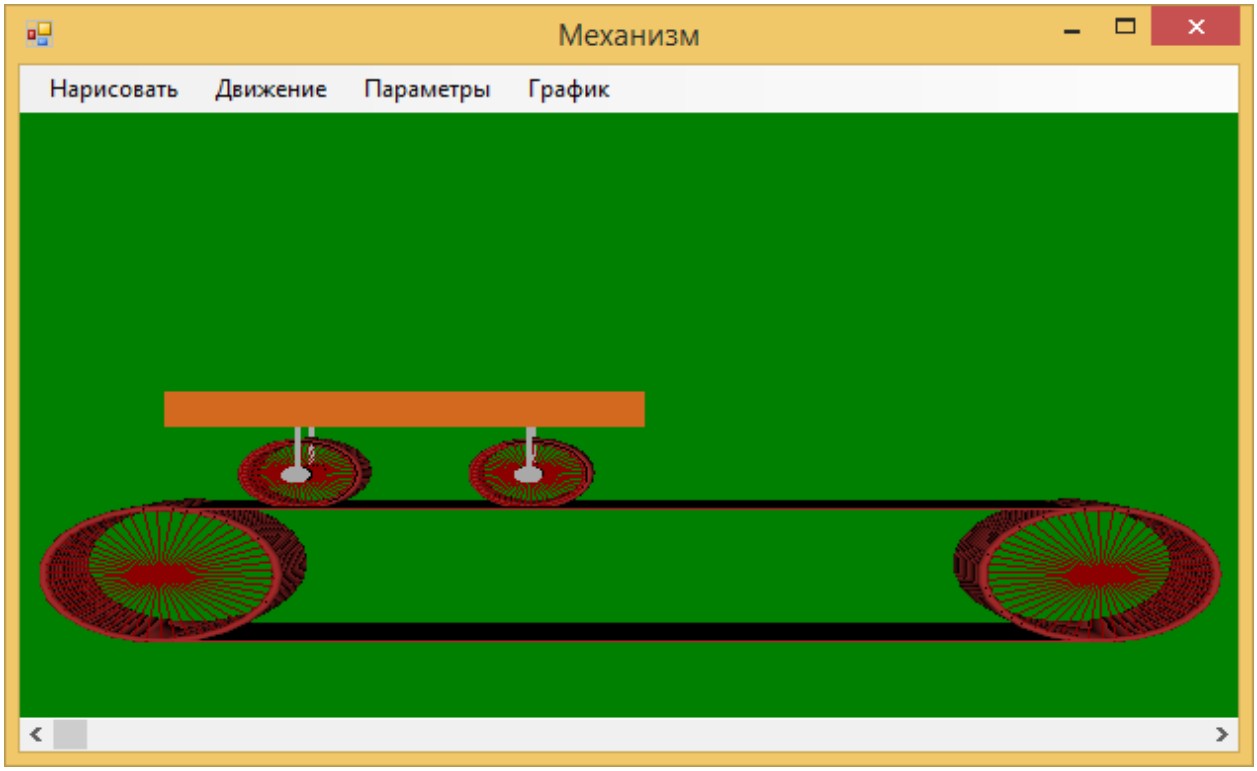
Додаток 3. Зовнішній вигляд форми Form3



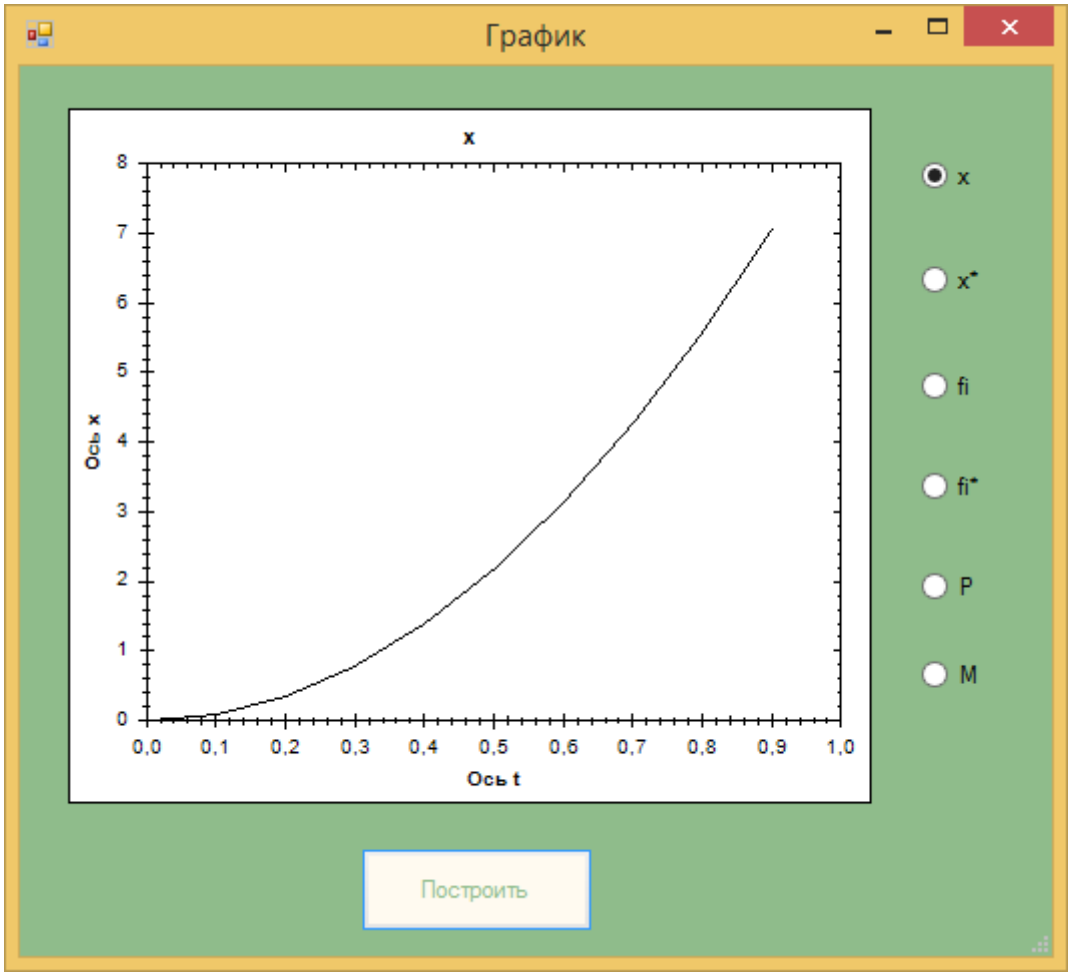
Додаток 4. Механізм в початковий момент часу



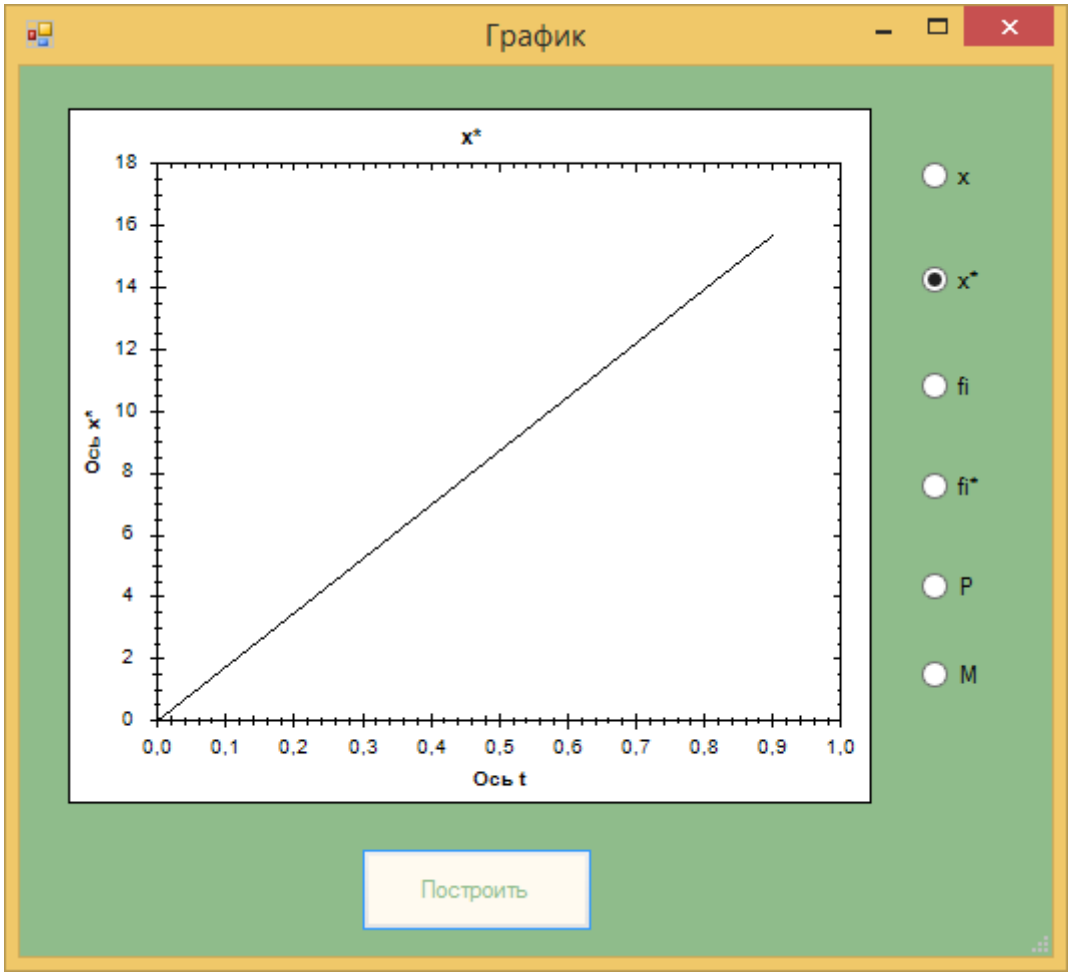
Додаток 5. Механізм в поточний момент часу



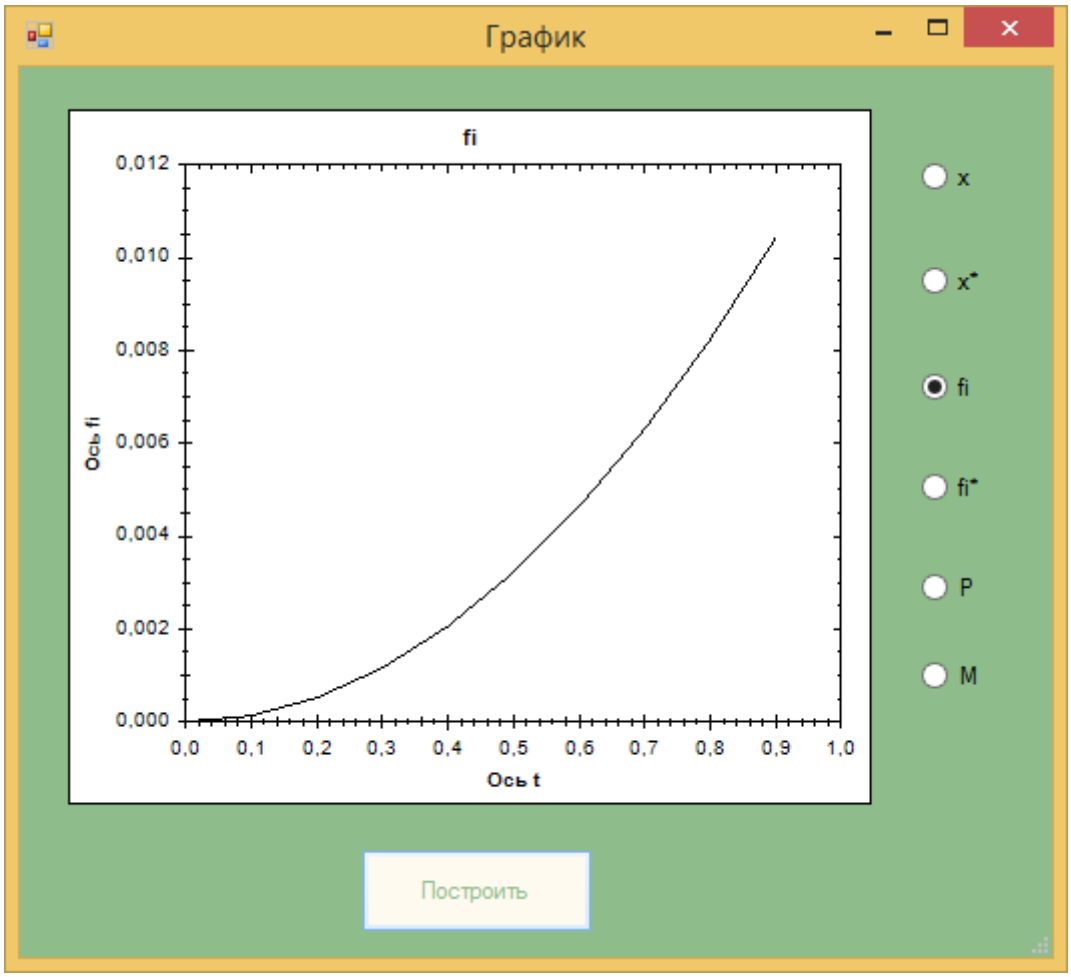
Додаток 6. Графік функції x=x(t).



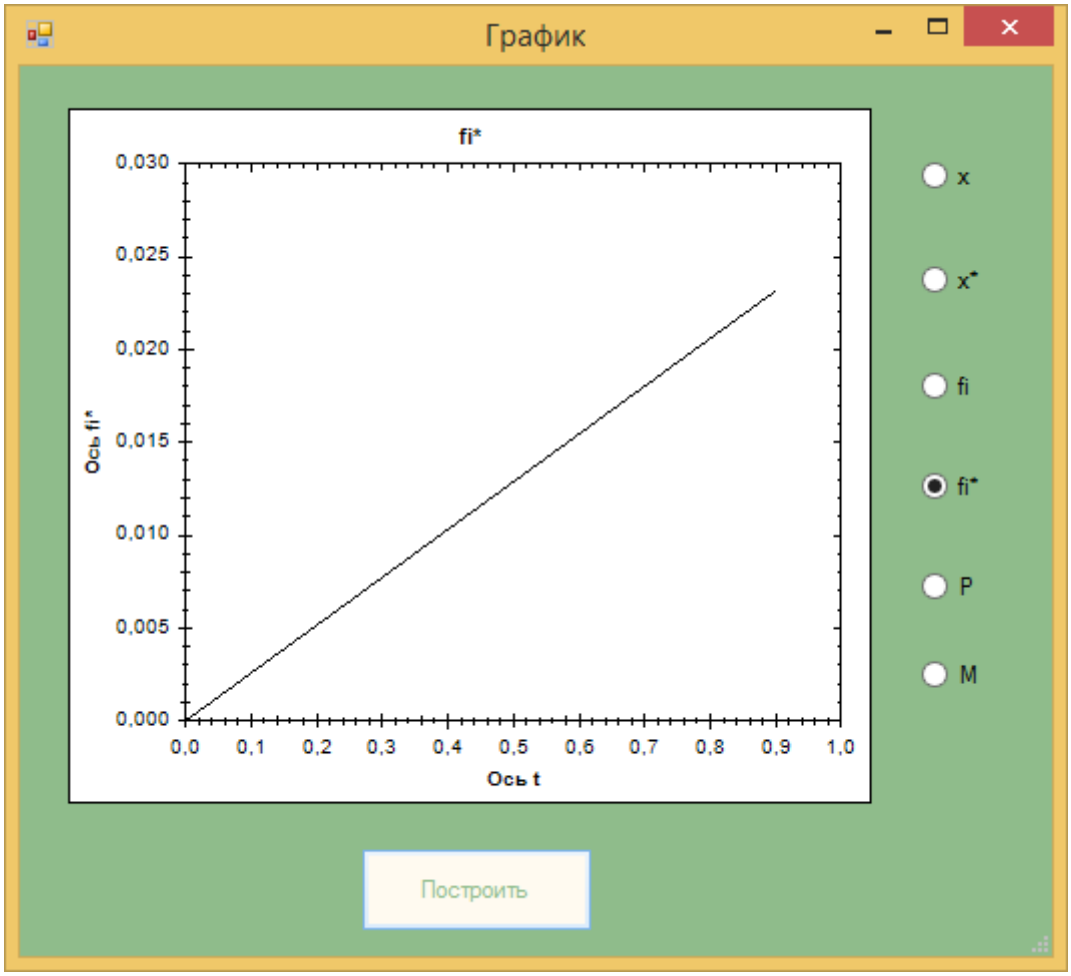
Додаток 7. Графік функції x’=x’(t).



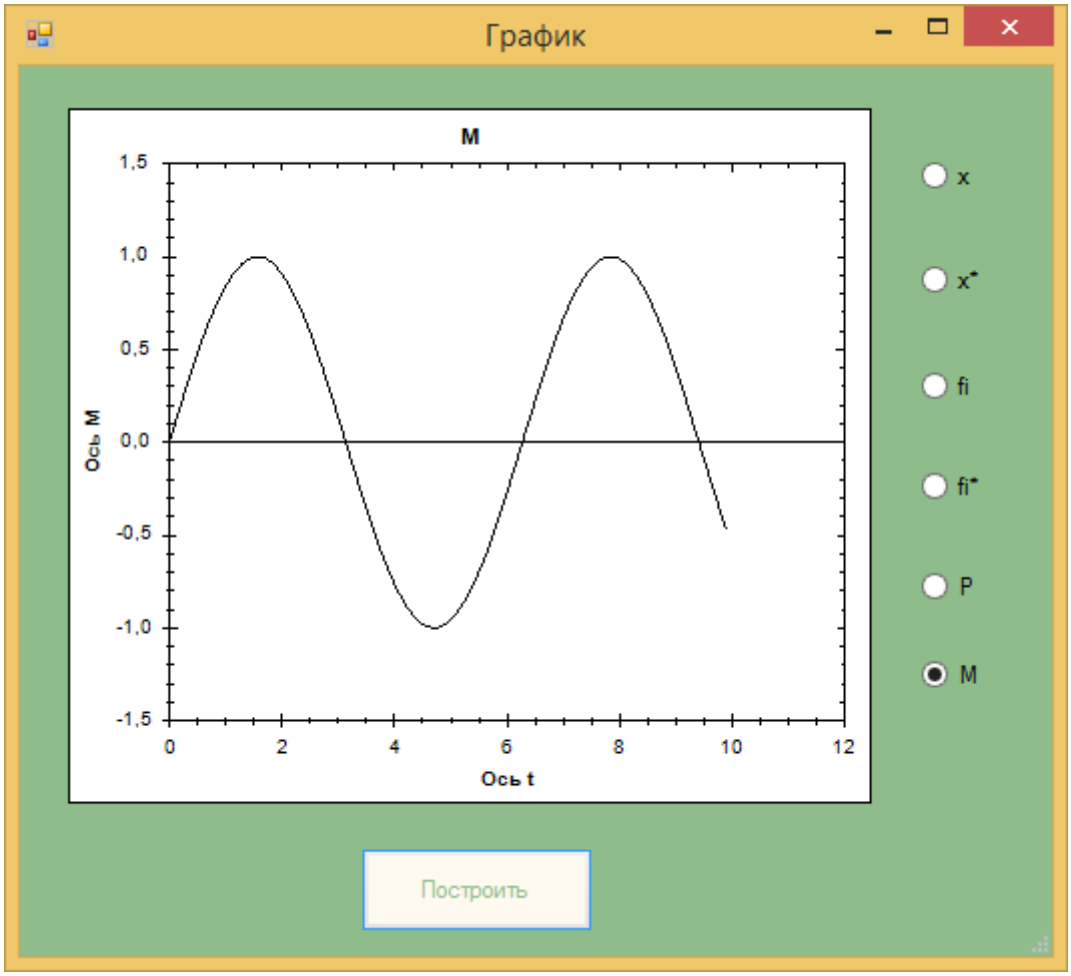
Додаток 8. Графік функції φ=φ(t).



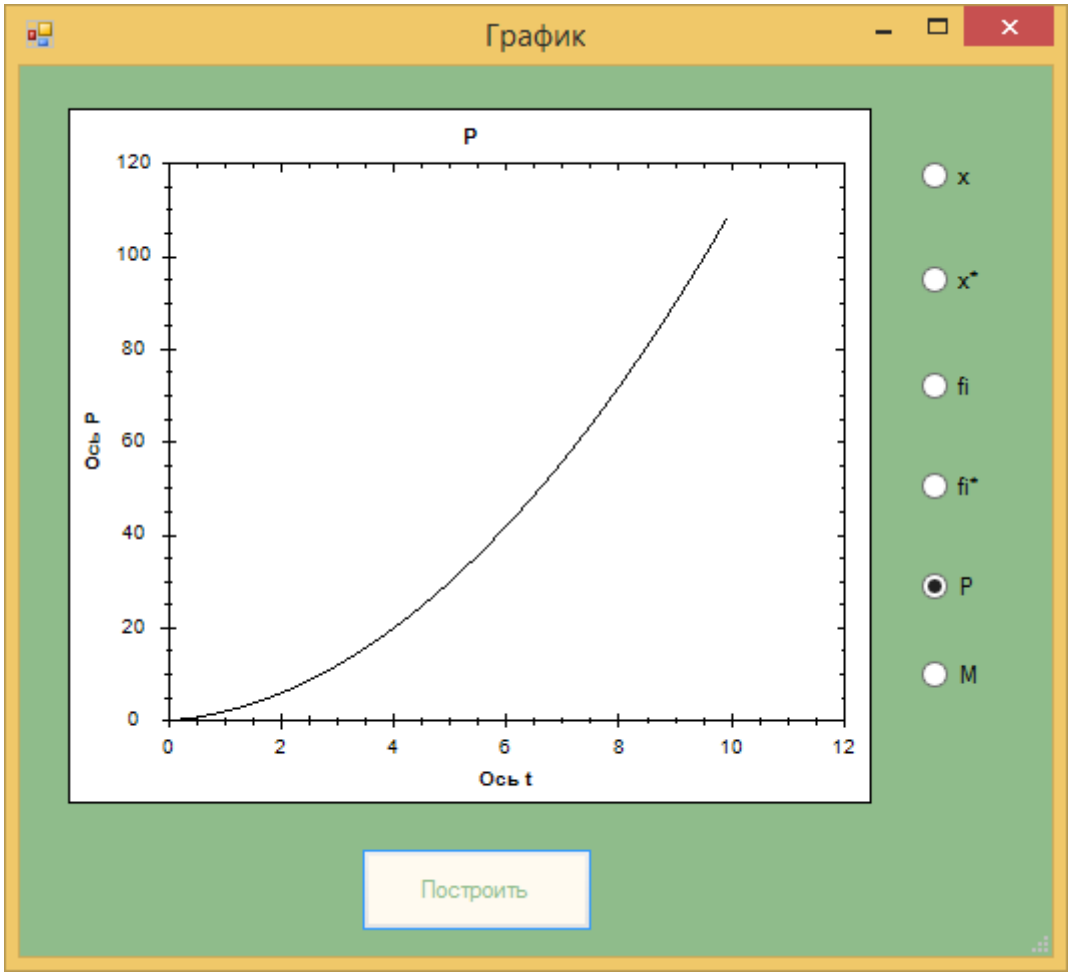
Додаток 9. Графік функції φ’=φ’(t).



Додаток 10. Графік функції M=M(t).



Додаток 11. Графік функції P=P(t).



Додаток 12. Клас форми Form1.

```
public partial class Form1 : Form
    {
        bool flag = false;
        Device d3d;
        Mesh koleso2, koleso3;
        Mesh podshipnik;
        Mesh telegka;
        Mesh lenta;
        Mesh shuna;
        Mesh kriplenie;
        Mesh[] mas_obod;
        Mesh[] mas_obod2;

        static public float R2 = 1f;
        float H_R2 = 1f;
        static public float R3 = 2f;
        float H_R3 = 2f;
        float L_lenta = 16f;
        float H_lenta = 0.05f;
        float W_lenta = 2f;
        float L_telegka = 3f;
        float H_telegka = 1f;
        float W_telegka = 8f;
        float R_podshipnik = 0.05f;
        float l_kriplenie = 1f;
        float h_kriplenie = 0.05f;
        float w_kriplenie = 0.1f;// розмеры крипления(от тележки до колесика)
        float Z = 26f;//здвиг в глубину сцены
        float Y = -2f;//здвиг по вертикали
        float l_k = 4f;
        float h_t = 0.2f;
        int n = 10;
        int n2 = 5;

        static public float m1 = 3f;
        static public float m2 = 1f;
        static public float m3 = 2f;

        static public float y0 = 0f;
        static public float y1 = 0f;
        static public float y2 = 0f;
        static public float y3 = 0f;

        static public float A = 0.1f;
        static public float B = 0.01f;
        static public float C = 0.01f;

        static public float A1 = 0.3f;
        static public float B1 = 0.01f;
        static public float C1 = 0.01f;

        static public List<float> tt;
        static public List<float> xx;
        static public List<float> xt;
        static public List<float> ffi;
        static public List<float> fit;


        float fi = 0.1f;
        float x = 0.1f;
        float time = 0;
        float h = 0.1f;
```

```
delegate float myF(float x, float[] y, int n);

float[] y;
myF[] F;



Material kolesoMaterial;
Material podshipnikMaterial;
Material telegkaMaterial;
Material lentaMaterial;
Material kriplenieMaterial;
Material shunaMaterial;

public Form1()
{
    InitializeComponent();
    d3d = null;
    tt = new List<float>();
    xx = new List<float>();
    xt = new List<float>();
    ffi = new List<float>();
    fit = new List<float>();
}

private void Form1_Load(object sender, EventArgs e)
{

    try
    {
        // Устанавливаем режим отображения трехмерной графики
        PresentParameters d3dpp = new PresentParameters();
        d3dpp.BackBufferCount = 1;
        d3dpp.SwapEffect = SwapEffect.Discard;
        d3dpp.Windowed = true; // Выводим графику в окно
        d3dpp.MultiSample = MultiSampleType.None; // Выключаем антиалиасинг
        d3dpp.EnableAutoDepthStencil = true; // Разрешаем создание z-буфера
        d3dpp.AutoDepthStencilFormat = DepthFormat.D16; // Z-буфер в 16 бит
         d3d = new Device(0, // D3D_ADAPTER_DEFAULT - видеоадаптер по
            // умолчанию
              DeviceType.Hardware, // Тип устройства - аппаратный ускоритель
              this, // Окно для вывода графики
              CreateFlags.SoftwareVertexProcessing, // Геометрию обрабатывает CPU
              d3dpp);

    }
    catch (Exception exc)
    {
        MessageBox.Show(this, exc.Message, "Ошибка инициализации");
        Close(); // Закрываем окно
    }

    koleso2 = Mesh.Cylinder(d3d, R2, R2, H_R2, 50, 10);
    koleso3 = Mesh.Cylinder(d3d, R3, R3,H_R3, 50, 10);
    kolesoMaterial = new Material();
    kolesoMaterial.Diffuse = Color.DarkRed;
    kolesoMaterial.Specular = Color.White;

    lenta = Mesh.Box(d3d, L_lenta, H_lenta, W_lenta);
    lentaMaterial = new Material();
    lentaMaterial.Diffuse = Color.Brown;
    lentaMaterial.Specular = Color.White;
```

```
            telegka=Mesh.Box(d3d,W_telegka,H_telegka,L_telegka);
            telegkaMaterial.Diffuse = Color.Chocolate;
            lentaMaterial.Specular = Color.White;

            podshipnik = Mesh.Cylinder(d3d,0.25f*R2,0.25f*R2,1.5f*H_R2,100,20);
            podshipnikMaterial.Diffuse = Color.DarkGray;
            podshipnikMaterial.Specular = Color.White;

            kriplenie = Mesh.Box(d3d,w_kriplenie ,R2+l_kriplenie, h_kriplenie);
            kriplenieMaterial.Diffuse = Color.DarkGray;
            kriplenieMaterial.Specular = Color.White;

            mas_obod = new Mesh[n];
            for (int i = 0; i < n; i++)
                mas_obod[i] = Mesh.Torus(d3d, 0.5f*h_t,R3, 25, 25);

            mas_obod2 = new Mesh[n];
            for (int i = 0; i < n2; i++)
                mas_obod2[i] = Mesh.Torus(d3d, 0.5f * h_t, R2, 25, 25);

                }

        private void Form1_Paint(object sender, PaintEventArgs e)
        {
            // Очищаем буфер глубины и дублирующий буфер
            d3d.Clear(ClearFlags.Target | ClearFlags.ZBuffer, Color.Green, 1.0f, 0);
            d3d.BeginScene();
            SetupProekcii();

            if (flag)
            {

                //код механизма
                d3d.RenderState.FillMode = FillMode.Solid;
                d3d.Material = lentaMaterial;
                d3d.Transform.World = Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value
* Math.PI / 180)) * Matrix.Translation(0, -R2 + Y, Z);
                lenta.DrawSubset(0);// лента

                d3d.RenderState.FillMode = FillMode.WireFrame;
                d3d.Material = kolesoMaterial;
                d3d.Transform.World = Matrix.RotationZ(y[0]) * Matrix.Translation(0.5f *
L_lenta, 0, 0) * Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value * Math.PI / 180)) *
Matrix.Translation(0, -R2 - R3 + Y, Z);
                koleso3.DrawSubset(0);//колесо_большое

                d3d.Transform.World = Matrix.RotationZ(y[0]) * Matrix.Translation(-0.5f *
L_lenta, 0, 0) * Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value * Math.PI / 180)) *
Matrix.Translation(0, -R2 - R3 + Y, Z);
                koleso3.DrawSubset(0);//колесо_большое

                d3d.RenderState.FillMode = FillMode.Solid;
                d3d.Material = lentaMaterial;
                d3d.Transform.World = Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value
* Math.PI / 180)) * Matrix.Translation(0, -R2 - 2 * R3 + Y, Z);
                lenta.DrawSubset(0);// лента

                for (int i = 0; i < n; i++)
                {
                    d3d.Transform.World = Matrix.Translation(0.5f * L_lenta, 0, -0.5f *
H_R3 - 0.5f * h_t + h_t * (i + 1)) * Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value
* Math.PI / 180)) * Matrix.Translation(0, -R2 - R3 + Y, Z);
                    mas_obod[i].DrawSubset(0);
                }
```

```
                for (int i = 0; i < n; i++)
                {
                    d3d.Transform.World = Matrix.Translation(-0.5f * L_lenta, 0, -0.5f *
H_R3 - 0.5f * h_t + h_t * (i + 1)) * Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value
* Math.PI / 180)) * Matrix.Translation(0, -R2 - R3 + Y, Z);
                    mas_obod[i].DrawSubset(0);
                }

                for (int i = 0; i < n2; i++)
                {
                    d3d.Transform.World = Matrix.Translation(0.5f * L_lenta - R2 - y[2] -
y[0] * R2, 3*R2, -0.5f * H_R2 - 0.5f * h_t + h_t * (i + 1)) *
Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value * Math.PI / 180)) *
Matrix.Translation(0, -R2 - R3 + Y, Z);
                    mas_obod2[i].DrawSubset(0);
                }

                for (int i = 0; i < n2; i++)
                {
                    d3d.Transform.World = Matrix.Translation( 3*R2 - y[2] - y[0] * R2, 3
* R2, -0.5f * H_R2 - 0.5f * h_t + h_t * (i + 1)) *
Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value * Math.PI / 180)) *
Matrix.Translation(0, -R2 - R3 + Y, Z);
                    mas_obod2[i].DrawSubset(0);
                }

                    x = 0.1f * time;

                d3d.RenderState.FillMode = FillMode.WireFrame;
                d3d.Material = kolesoMaterial;
                d3d.Transform.World = Matrix.RotationZ(y[2] / R2) *
Matrix.Translation(0.5f * L_lenta - R2 - y[2] - y[0] * R3, 0, 0) *
Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value * Math.PI / 180)) *
Matrix.Translation(0, 0 + Y, Z);
                koleso2.DrawSubset(0);//колесо_маленькое

                d3d.Transform.World = Matrix.RotationZ(y[2] / R2) *
Matrix.Translation(0.5f * L_lenta - R2 - l_k - y[2] - y[0] * R3, 0, 0) *
Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value * Math.PI / 180)) *
Matrix.Translation(0, 0 + Y, Z);
                koleso2.DrawSubset(0);

                //тележка(коробочка)
                d3d.RenderState.FillMode = FillMode.Solid;
                d3d.Material = telegkaMaterial;

                d3d.Transform.World = Matrix.Translation(0.5f * L_lenta - R2 - 0.5f * l_k
- y[2] - y[0] * R3, 0, 0) * Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value * Math.PI
/ 180)) * Matrix.Translation(0, R2+l_kriplenie + Y, Z);
                telegka.DrawSubset(0);

                //подшипник
                d3d.RenderState.FillMode = FillMode.Solid;
                d3d.Material = podshipnikMaterial;
                d3d.Transform.World = Matrix.Translation(0.5f * L_lenta - R2 - l_k - y[2]
- y[0] * R3, 0, 0) * Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value * Math.PI /
180)) * Matrix.Translation(0, 0 + Y, Z);
                podshipnik.DrawSubset(0);

                d3d.Transform.World = Matrix.Translation(0.5f * L_lenta - R2 - y[2] -
y[0] * R3, 0, 0) * Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value * Math.PI / 180))
* Matrix.Translation(0, 0 + Y, Z);
                podshipnik.DrawSubset(0);

                //крипления
```

```
                d3d.RenderState.FillMode = FillMode.Solid;
                d3d.Material = kriplenieMaterial;
                d3d.Transform.World = Matrix.Translation(0.5f * L_lenta - R2 - l_k - y[2]
- y[0] * R3, 0, -0.5f * H_R2 - 0.07f) *
Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value * Math.PI / 180)) *
Matrix.Translation(0, 0 + Y + 0.5f*(R2+l_kriplenie), Z);
                kriplenie.DrawSubset(0);

                d3d.RenderState.FillMode = FillMode.Solid;
                d3d.Material = kriplenieMaterial;
                d3d.Transform.World = Matrix.Translation(0.5f * L_lenta - R2 - y[2] -
y[0] * R3, 0, -0.5f * H_R2 - 0.07f) * Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value
* Math.PI / 180)) * Matrix.Translation(0, 0 + Y + 0.5f * (R2 + l_kriplenie), Z);
                kriplenie.DrawSubset(0);

                d3d.RenderState.FillMode = FillMode.Solid;
                d3d.Material = kriplenieMaterial;
                d3d.Transform.World = Matrix.Translation(0.5f * L_lenta - R2 - l_k - y[2]
- y[0] * R3, 0, 0.5f * H_R2 + 0.07f) *
Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value * Math.PI / 180)) *
Matrix.Translation(0, 0 + Y + 0.5f * (R2 + l_kriplenie), Z);
                kriplenie.DrawSubset(0);

                d3d.RenderState.FillMode = FillMode.Solid;
                d3d.Material = kriplenieMaterial;
                d3d.Transform.World = Matrix.Translation(0.5f * L_lenta - R2 - y[2] -
y[0] * R3, 0, 0.5f * H_R2 + 0.07f) * Matrix.RotationY(Convert.ToSingle(hScrollBar1.Value
* Math.PI / 180)) * Matrix.Translation(0, 0 + Y + 0.5f * (R2 + l_kriplenie), Z);
                kriplenie.DrawSubset(0);


            }


            d3d.EndScene();
            //Обновление состояния дисплея
            d3d.Present();


        }

        private void SetupProekcii()
        {
            // Устанавливаем параметры источника освещения
            // Устанавливаем параметры источника освещения
            d3d.Lights[0].Enabled = true;   // Включаем нулевой источник освещения
            d3d.Lights[0].Diffuse = Color.White;         // Цвет источника освещения
            d3d.Lights[0].Position = new Vector3(0, 0, 0); // Задаем координаты
            d3d.Transform.Projection = Matrix.PerspectiveFovLH((float)Math.PI / 4,
 this.Width / this.Height, 1.0f, 50.0f);
        }

        private void hScrollBar1_Scroll(object sender, ScrollEventArgs e)
        {
            this.Invalidate();
        }

        private void параметрыToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Form2 f2 = new Form2();
            if (f2.ShowDialog() == DialogResult.OK)
            {
                time = 0;
                koleso2.Dispose();
```

```
            koleso2 = Mesh.Cylinder(d3d, R2, R2, H_R2, 50, 10);
            koleso3.Dispose();
            koleso3 = Mesh.Cylinder(d3d, R3, R3, H_R3, 50, 10);
            for (int i = 0; i < n; i++)
                mas_obod[i].Dispose();


            for (int i = 0; i < n; i++)
                mas_obod[i] = Mesh.Torus(d3d, 0.5f * h_t, R3, 25, 25);

            kriplenie.Dispose();
            kriplenie = Mesh.Box(d3d, w_kriplenie, R2 + l_kriplenie, h_kriplenie);


            нарисоватьToolStripMenuItem_Click(sender, new EventArgs());
        }

    }

    private void нарисоватьToolStripMenuItem_Click(object sender, EventArgs e)
    {
        y = new float[4];
        y[0] = y0;
        y[1] = y1;
        y[2] = y2;
        y[3] = y3;

        F = new myF[4];
        F[0] = F0;
        F[1] = F1;
        F[2] = F2;
        F[3] = F3;
        flag = true;

        начатьToolStripMenuItem.Enabled = true;

        this.Invalidate();
    }


    private void начатьToolStripMenuItem_Click(object sender, EventArgs e)
    {
        timer1.Enabled = true;

        tt.Clear();
        xx.Clear();
        xt.Clear();
        ffi.Clear();
        fit.Clear();
        tt.Add(time);
        xx.Add(y[2]);
        xt.Add(y[3]);
        ffi.Add(y[0]);
        fit.Add(y[1]);


        y = Runge(time, y, F, 4, h);
        time = time + h;
        начатьToolStripMenuItem.Enabled = false;
        остановитьToolStripMenuItem.Enabled = true;
        параметрыToolStripMenuItem.Enabled = false;
    }


    private void timer1_Tick(object sender, EventArgs e)
```

```
        {
            this.Invalidate();
            tt.Add(time);
            xx.Add(y[2]);
            xt.Add(y[3]);
            ffi.Add(y[0]);
            fit.Add(y[1]);
            y = Runge(time, y, F, 4, h);
            time = time + h;
        }


        float[] Runge(float x0, float[] y, myF[] f, int n, float h)// метод Рунге-Кутта
        {
            int j, i;
            float[] y_new = new float[n];
            float[] k1 = new float[n];
            float[] k2 = new float[n];
            float[] k3 = new float[n];
            float[] k4 = new float[n];

            float x = x0;
            for (j = 0; j < n; j++)
                k1[j] = h * f[j](x, y, n);
            for (i = 0; i < n; i++)
                y_new[i] = y[i] + 0.5f * k1[i];
            for (j = 0; j < n; j++)
                k2[j] = h * f[j](x + 0.5f * h, y_new, n);
            for (i = 0; i < n; i++)
                y_new[i] = y[i] + 0.5f * k2[i];
            for (j = 0; j < n; j++)
                k3[j] = h * f[j](x + 0.5f * h, y_new, n);
            for (i = 0; i < n; i++)
                y_new[i] = y[i] + k3[i];
            for (j = 0; j < n; j++)
                k4[j] = h * f[j](x + h, y_new, n);
            for (i = 0; i < n; i++)
                y_new[i] = y[i] + (k1[i] + 2 * k2[i] + 2 * k3[i] + k4[i]) / 6;

            return y_new;
        }

        float F0(float x, float[] y, int n)
        {

            return y[1];
        }

        float F1(float x, float[] y, int n)
        {
            float A = (m1*m2+5*m2*m2+m1*m3+3*m2*m3)*R3*R3;
            float B = 2 * P(x) * R3 * m2 + M(x) * m1 + 3 * M(x) * m2;
            return B / A;
        }

        float F2(float x, float[] y, int n)
       {

           return y[3];
        }

       float F3(float x, float[] y, int n)
       {
            float C = (m1*m2+2*m2*m2+m1*m3+3*m2*m3)*R3;
            float D = P(x)*R3*m3-M(x)*m1-2*M(x)*m2;
```

```
        return C / D;
    }

    private void остановитьToolStripMenuItem_Click(object sender, EventArgs e)
    {
        timer1.Enabled = false;
        начатьToolStripMenuItem.Enabled = true;
        остановитьToolStripMenuItem.Enabled = false;
        параметрыToolStripMenuItem.Enabled = true;
    }

    static public int fM = 3;
    static public int fP = 3;

    static public float M (float t)
    {
        float mom = 0;
        switch (fM)
        {
            case 1: mom = Convert.ToSingle(A * Math.Sin(B * t + C));
                break;
            case 2: mom = A * t + B;
                break;
            case 3: mom = A;
                break;
            case 4: mom = A * t * t + B * t + C;
                break;

        }
        return mom;
    }

    static public float P(float t)
    {
        float mom = 0;
        switch (fP)
        {
            case 1: mom = Convert.ToSingle(A1 * Math.Sin(B1 * t + C1));
                break;
            case 2: mom = A1 * t + B1;
                break;
            case 3: mom = A1;
                break;
            case 4: mom = A1 * t * t + B1 * t + C1;
                break;

        }
        return mom;

    }

    private void графикToolStripMenuItem_Click(object sender, EventArgs e)
    {
        Form3 f3 = new Form3();
        f3.ShowDialog();
    }
```

Додаток 13. Клас форми Form2.

```
public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();

        }

        private void numericUpDown6_ValueChanged(object sender, EventArgs e)
        {

        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form1.R2 = Convert.ToSingle(numericUpDown1.Value);
            Form1.R3 = Convert.ToSingle(numericUpDown2.Value);
            Form1.m1 = Convert.ToSingle(numericUpDown3.Value);
            Form1.m2 = Convert.ToSingle(numericUpDown4.Value);
            Form1.m3 = Convert.ToSingle(numericUpDown5.Value);
            Form1.y0 = Convert.ToSingle(numericUpDown6.Value);
            Form1.y1 = Convert.ToSingle(numericUpDown7.Value);
            Form1.y2 = Convert.ToSingle(numericUpDown8.Value);
            Form1.y3 = Convert.ToSingle(numericUpDown9.Value);
            Form1.A  = Convert.ToSingle(numericUpDown10.Value);
            Form1.B =  Convert.ToSingle(numericUpDown11.Value);
            Form1.C =  Convert.ToSingle(numericUpDown12.Value);
            Form1.A1=  Convert.ToSingle(numericUpDown13.Value);
            Form1.B1 = Convert.ToSingle(numericUpDown14.Value);
            Form1.C1 = Convert.ToSingle(numericUpDown15.Value);
            if (radioButton1.Checked == true)
                Form1.fM = 1;
            if (radioButton2.Checked == true)
                Form1.fM = 2;
            if (radioButton3.Checked == true)
                Form1.fM = 3;
            if (radioButton4.Checked == true)
                Form1.fM = 4;
            if (radioButton5.Checked == true)
                Form1.fP = 1;
            if (radioButton6.Checked == true)
                Form1.fP = 2;
            if (radioButton7.Checked == true)
                Form1.fP = 3;
            if (radioButton8.Checked == true)
                Form1.fP = 4;

        }

        private void groupBox5_Enter(object sender, EventArgs e)
        {

        }

        private void Form2_Load(object sender, EventArgs e)
        {
            numericUpDown1.Value = Convert.ToDecimal(Form1.R2);
            numericUpDown2.Value = Convert.ToDecimal(Form1.R3);
            numericUpDown3.Value = Convert.ToDecimal(Form1.m1);
            numericUpDown4.Value = Convert.ToDecimal(Form1.m2);
```

```
        numericUpDown5.Value = Convert.ToDecimal(Form1.m3);
        numericUpDown6.Value = Convert.ToDecimal(Form1.y0);
        numericUpDown7.Value = Convert.ToDecimal(Form1.y1);
        numericUpDown8.Value = Convert.ToDecimal(Form1.y2);
        numericUpDown9.Value = Convert.ToDecimal(Form1.y3);
        numericUpDown10.Value = Convert.ToDecimal(Form1.A);
        numericUpDown11.Value = Convert.ToDecimal(Form1.B);
        numericUpDown12.Value = Convert.ToDecimal(Form1.C);
        numericUpDown13.Value = Convert.ToDecimal(Form1.A1);
        numericUpDown14.Value = Convert.ToDecimal(Form1.B1);
        numericUpDown15.Value = Convert.ToDecimal(Form1.C1);

        numericUpDown1.Maximum = Convert.ToDecimal(1.8);

        numericUpDown1.Increment = Convert.ToDecimal(0.01);
        numericUpDown2.Increment = Convert.ToDecimal(0.01);
        numericUpDown3.Increment = Convert.ToDecimal(0.01);
        numericUpDown4.Increment = Convert.ToDecimal(0.01);
        numericUpDown5.Increment = Convert.ToDecimal(0.01);
        numericUpDown6.Increment = Convert.ToDecimal(0.01);
        numericUpDown7.Increment = Convert.ToDecimal(0.01);
        numericUpDown8.Increment = Convert.ToDecimal(0.01);
        numericUpDown9.Increment = Convert.ToDecimal(0.01);
        numericUpDown10.Increment = Convert.ToDecimal(0.01);
        numericUpDown11.Increment = Convert.ToDecimal(0.01);
        numericUpDown12.Increment = Convert.ToDecimal(0.01);
        numericUpDown13.Increment = Convert.ToDecimal(0.01);
        numericUpDown14.Increment = Convert.ToDecimal(0.01);
        numericUpDown15.Increment = Convert.ToDecimal(0.01);
    }
```

Додаток 14. Клас форми Form3.

```
public partial class Form3 : Form
 {
     GraphPane myPane = null;

     public Form3()
     {
         InitializeComponent();
     }

     private void Form3_Load(object sender, EventArgs e)
     {
         myPane = zedGraphControl1.GraphPane;
     }

     private void button1_Click(object sender, EventArgs e)
     {
         myPane.CurveList.Clear();
         // Создадим список точек
         PointPairList list = new PointPairList();

         if (radioButton6.Checked)
         {
             for (float t = 0; t < 10; t += 0.1f)
             {
                 list.Add(t, Form1.M(t));
             }
         }

         if (radioButton5.Checked)
         {
             for (float t = 0; t < 10; t += 0.1f)
             {
                 list.Add(t, Form1.P(t));
             }
         }
         if (radioButton1.Checked)
         {
             for (int i = 0; i < Form1.tt.Count; i++)
             {
                 list.Add(Form1.tt[i], Form1.xx[i]);
             }
         }
         if (radioButton2.Checked)
         {
             for (int i = 0; i < Form1.tt.Count; i++)
             {
                 list.Add(Form1.tt[i], Form1.xt[i]);
             }
         }
         if (radioButton3.Checked)
         {
             for (int i = 0; i < Form1.tt.Count; i++)
             {
                 list.Add(Form1.tt[i], Form1.ffi[i]);
             }
         }
         if (radioButton4.Checked)
         {
             for (int i = 0; i < Form1.tt.Count; i++)
             {
                 list.Add(Form1.tt[i], Form1.fit[i]);
             }
```

```
            }

            LineItem myCurve = myPane.AddCurve("", list, Color.Black);
            //в узловых точках не должно быть никаких фигурок, нужна математическая
кривая
            myCurve.Symbol.IsVisible = false;
            //что-то вроде автомасштабирования осей
            zedGraphControl1.AxisChange();
            // Обновляем график
            zedGraphControl1.Invalidate();

        }

        private void Form3_Shown(object sender, EventArgs e)
        {
            myPane.Title.Text = "M";
            myPane.XAxis.Title.Text = "Ось t";
            myPane.YAxis.Title.Text = "Ось M";

            zedGraphControl1.Invalidate();

        }


        private void radioButton1_Click(object sender, EventArgs e)
        {
            myPane.Title.Text = ((RadioButton)sender).Text;
            myPane.XAxis.Title.Text = "Ось t";
            myPane.YAxis.Title.Text = "Ось " + ((RadioButton)sender).Text;
            myPane.CurveList.Clear();
            zedGraphControl1.Invalidate();
        }


    }
```