**Министерство науки и высшего образования Российской Федерации**

**ФГАОУ ВО «УрФУ имени первого Президента России Б.Н. Ельцина»**

Кафедра «школы бакалавриата (школа)»

Оценка работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель от УрФУ Желонкина Н.И.

Тема задания на практику

Механика и математическое моделирование

ОТЧЕТ

Вид практики Учебная практика

Тип практики Учебная практика, Научно-исследовательская работа (получение первичных навыков научно-исследовательской работы)

Руководитель практики от предприятия (организации) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ФИО руководителя Подпись**

Студент Бренчук Я.В.

**ФИО студента**

Специальность (направление подготовки) 01.03.03 Механика и математическое моделирование

Группа МЕН-290705

Екатеринбург 2021

Оглавление

[**Вступление** 3](#_Toc75075021)

[**Решение поставленных задачи** 4](#_Toc75075022)

[**Условие задачи** 4](#_Toc75075023)

[**Решение** 5](#_Toc75075024)

[***Участок АВ:*** 5](#_Toc75075025)

[**Участок ВС:** 8](#_Toc75075026)

[**Участок СE:** 10](#_Toc75075027)

[**Описание кода программы** 12](#_Toc75075028)

[**Приложение** 17](#_Toc75075029)

[**Файл MainWindow.xaml.сs** 17](#_Toc75075030)

[**Файл MainWindow.xaml:** 22](#_Toc75075031)

[**Подведение итогов работы** 37](#_Toc75075032)

[**Список литературы** 38](#_Toc75075033)

# **Вступление**

Данная курсовая работа посвящена изучению механики, математическому моделированию и построению компьютерной модели. В отличие от работы по теоретической механике эта работа отличается тем, что мы не только сможем мысленно представить, как движется тело, но и сможем увидеть траекторию тела и поменять показатели, чтобы выстроенная траектория изменялась.

Были поставлены следующие цели и задачи.

Цель – создание модели полета тела по полученным уравнениям движения

Задачи:

1. Ознакомиться с порядком обрамления и написания курсовой работы
2. Изучить теоретическую часть для решения задачи
3. Решить и объяснить решение поставленной задачи
4. Написать и объяснить программу:
   1. изучить язык программирования C# и язык разметки XAML
   2. спроектировать интерес с помощью языка разметки XAML
   3. добавить ввод и вывод данных с помощью языков XAML и C#
   4. с помощью полученных уравнений движений нарисовать траектории в окне с помощью языков XAML и C#
   5. анимировать нарисованную траекторию
5. Сделать выводы о проделанной работе
6. Отчитаться за проделанную работу в письменной и устной форме.

# **Решение поставленных задачи**

## **Условие задачи**

Тело массой *т*, получив в точке *А* начальную скорость V*0,* движется по поверхности *АВС*, расположенной в вертикальной плоскости. В точке *С* тело покидает поверхность и в точке *Е* падает на берег рва. Движение тела разбито на три участка.

1. На **участке *АВ*** на тело, кроме силы тяжести, действуют движущая сила *Q,* и сила сопротивления среды *R.* Трением тела о поверхность на участке *АВ* пренебречь.

2. В точке *В* тело, не изменяя величины своей скорости движения, переходит на **участок *ВС***, где на него действуют сила тяжести, сила трения (коэффициент трения скольжения груза о поверхность *f =*0,2) и переменная сила *F.* Время движения груза по участку *ВС* составляет *t =* 4 *с*.

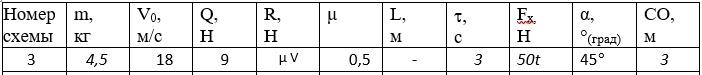
3. Движение тела на **участке *СЕ*** происходит под действием силы тяжести; сопротивлением воздуха пренебречь.

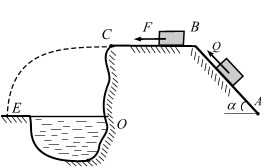
Требуется, считая тело материальной точкой и зная расстояние *AB= L*,

или время движения тела на участке АВ (*τ),* вычислить:

1. скорость движения тела в точке *В* – *V* B;
2. уравнение движения тела на участке *ВС* – *x= f* (*t*);
3. скорость движения тела в точке *С* (*V*C) в момент времени *4 c*;
4. уравнение траектории движения тела на участке *СЕ*, т. е. *y3= f3(x).*

Начальные условия

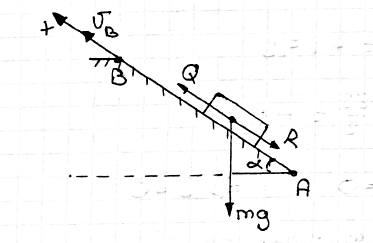


Схема графика

## Решение

### Участок АВ:

При решении задачи мы пренебрегаем трением на этом участке. Принимая тело за материальную точку, покажем действующие на него силы: вес , нормальную реакцию , движущую силу и силу сопротивления среды . Возьмем начало координат в точке А, ось направим вдоль прямой AB.



Cоставим дифференциальное уравнение движения тела на участке АВ:

Разделим правую и левую часть на :

По условию сила сопротивления среды равна или тоже . Таким образом:

Обозначим за , а за . Получим:

Перегруппируем:

Решаем дифференциальное уравнение с постоянными коэффициентами:

Вынесем , получим:

Решая уравнение, получаем:

Тогда общее решение получается:

Найдем теперь частное решение:

Тогда решением уравнения получается:

Найдем интегрируемые постоянные, для этого продифференцируем полученное уравнение:

Подставляем начальные данные, т. е. . Получаем

Тогда можно найти при :

Найдем теперь уравнение движения:

Подставляя начальные условия , получаем:

Тогда уравнение движения будет иметь вид:

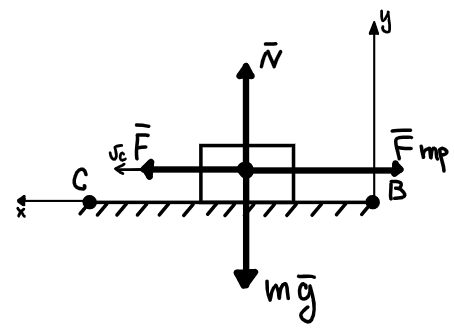
Чтобы нарисовать мой график в программе, нужно повернуть его на , для этого воспользуемся формулой поворота координат:

В нашем случае координата «у» не изменяется, т. е. равна нулю, тогда получаем:

Или

### Участок ВС:

Рассмотрим движение камня от точки В до точки С. Покажем силу тяжести G, действующую на тело, силу трения и силу F. Составим дифференциальное уравнение его движения:



*, где*

Разделим правую и левую часть равенства на :

Проинтегрируем, т. к. F зависит от t, то обозначим F как :

Найдем постоянные интегрирования:

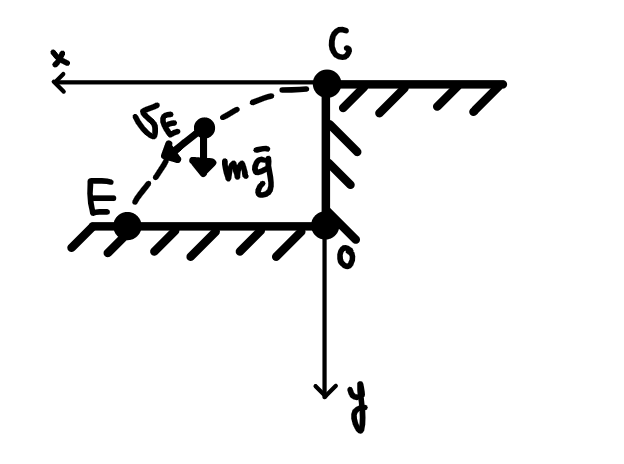
Подставив начальные условия ,

Найдем скорость точки в точке C

Тогда уравнение движения принимает следующий вид:

### Участок СE:

Рассмотрим движение камня от точки C до точки E. Покажем силу тяжести G и скорость . Возьмём начало системы отсчета в точке С и направим ось Оу вниз. Составим дифференциальное уравнение его движения:



По оси X:

Т. к. на тела не действуют никакие силы. Разделим правую и левую часть на :

Найдем постоянную интегрирования , подставим начальные данные, :

При

Получаем уравнение движения по оси Ox:

По оси Oy:

Разделим правую и левую часть на m:

*=>*

Получаем уравнения движения:

# **Описание кода программы**

При запуске программы мы можем главное окно программы как показано на рисунке 1.



Рисунок 1 Главное окно программы

При запуске программы вызывется метод MainWindow, в котором вызывается стандартный метод InitializeComponent, также у переменной timer появляется обработчик событий, который будет вызываться по истечению времени, дальше таймеру присваиваем интервал, равный 20 миллисекундам, который имеет вид:

public MainWindow()  
{  
 InitializeComponent();  
 timer.Tick += new EventHandler(OnTimer);  
 timer.Interval = new TimeSpan(0, 0, 0, 0, 20);  
}

Дальше ожидается нажатие на кнопку, и, когда нажали кнопку “Start”, вызывается метод ButtonStart, который имеет вид:

Clear();  
InputData();   
if (!statusInput) return;  
ConstantIntegrationsAB();

timer.Start();   
ellipse.Fill = new SolidColorBrush(Color.FromRgb(157, 129, 186));  
Speed();   
StartDataCanvas();

В этом методе сначала вызывается метод Clear(), который осуществляет очистку данных, таких как траектория polyline. Переменным присваиваем ноль и также очищаем скорости в точках. Данный метод имеет вид:

private void Clear()  
{  
 polyline.Points.Clear();  
 time = xAB = xCE = xСВ = yCE = yСВ = 0;  
 speedC.Text = "........";  
 speedB.Text = "........";  
 speedA.Text = "........";  
}

Дальше вызывается метод по вводу данных пользователем InputData(), также этот метод проверяет на «глупости» пользователя, например, как:

1. Отрицательная скорость
2. Не все заполненные поля ввода
3. Невозможные углы

Также с помощью методов IsTextAllowed и TextBox\_PreviewTextInput мы запрещаем пользователю вводить любые значения, кроме цифр от нуля до 9 и запятой.

private static readonly Regex *regex* = new Regex("[^0-9,]");  
private static bool IsTextAllowed(string text)  
{  
 return !*regex*.IsMatch(text);  
}  
private void TextBox\_PreviewTextInput(object sender, TextCompositionEventArgs e)  
{  
 e.Handled = !IsTextAllowed(e.Text);  
}

Метод InputData() имеет вид:

private void InputData()  
{  
 if (body\_mass.Text == "" && angle.Text == "" &&   
 coefficient\_μ.Text == "" && driving\_force.Text == "" &&   
 initial\_speed.Text == "" &&   
 time\_τ.Text == "" && height.Text == "")  
 {  
 MessageBox.Show("Не все поля были заполнены ", "🔥Ошибка🔥",  
 MessageBoxButton.OK, MessageBoxImage.Error);  
 statusInput = false;  
 }  
 else if (body\_mass.Text == "0" && angle.Text == "0" &&  
 coefficient\_μ.Text == "0" && driving\_force.Text == "0" &&  
 initial\_speed.Text == "0" &&  
 time\_τ.Text == "0" && height.Text == "0")  
 {  
 MessageBox.Show("Нулевые данные", "🔥Ошибка🔥",   
 MessageBoxButton.OK, MessageBoxImage.Error);  
 statusInput = false;  
 }  
 else if (Convert.ToDouble(angle.Text) > 90 || Convert.ToDouble(angle.Text) == 0)  
 {  
 MessageBox.Show("Невозможные углы", "🔥Ошибка🔥",  
 MessageBoxButton.OK, MessageBoxImage.Error);  
 statusInput = false;  
 }  
 else if(statusInput)  
 {  
 \_body\_mass = Convert.ToDouble(body\_mass.Text);  
 \_angle = Convert.ToDouble(angle.Text) \* Math.PI / 180;  
 \_coefficient\_μ = Convert.ToDouble(coefficient\_μ.Text);  
 \_driving\_force = Convert.ToDouble(driving\_force.Text);  
 \_initial\_speed = Convert.ToDouble(initial\_speed.Text);  
 \_time\_τ = Convert.ToDouble(time\_τ.Text);  
 \_coefficient\_f = Convert.ToDouble(coefficient\_f.Text);  
 \_force\_F = Convert.ToDouble(force\_F.Text);  
 \_travel\_time = Convert.ToDouble((travel\_time.Text));  
 \_height = Convert.ToDouble(height.Text);

// для краткости  
 b = \_driving\_force / \_body\_mass - G \* Math.Sin(\_angle);  
 m = \_coefficient\_μ / \_body\_mass;  
 }  
}

Дальше мы проверяем, все ли данные были введены, и продолжаем выполнение программы, в противном случае выдаём ошибку о неправильном вводе данных. Если программа продолжила выполняться, то вызывается метод ConstantIntegrationsAB() по вычислению констант интегрирования на первом участке, этот метод имеет вид:

private void ConstantIntegrationsAВ()  
{  
 \_cAB\_1 = (\_initial\_speed - b/m)/-m;  
 \_cAB\_2 = -\_cAB\_1;  
}

Дальше мы запускаем таймер и красим эллипс в цвет, отличный от цвета фона программы. Затем вызывается метод Speed () по вычислению скоростей в отдельных точках, а также проверяем на отрицательную скорость, этот метод имеет вид:

private void Speed()  
{  
 \_speedB = b / m - m \* \_cAB\_1 \* Math.Exp(-m \* \_time\_τ);  
 \_speedС = \_force\_F \* \_travel\_time \* \_travel\_time / (2 \* \_body\_mass) - \_coefficient\_f \* G \* \_travel\_time +  
 \_speedB;  
 if (\_initial\_speed <= 0 || \_speedB <= 0 || \_speedС <= 0)  
 {  
 timer.Stop();  
 Clear();  
 ellipse.Fill = new SolidColorBrush(Color.FromRgb(235, 235, 235));  
 MessageBox.Show("Нулевая скорость", "🔥Ошибка🔥",  
 MessageBoxButton.OK, MessageBoxImage.Error);  
 }  
 else  
 {  
 speedC.Text = Math.Round(\_speedС, 1) + "м/с";  
 speedB.Text = Math.Round(\_speedB, 1) + "м/с";  
 speedA.Text = Math.Round(\_initial\_speed, 1) + "м/с";  
 }  
}

Дальше вызывается метод StartDataCanvas() по масштабированию графика, который вычисляет текущий размер окна с графиком и максимальный размер графика, потом с помощью математических операций вычисляются коэффициенты масштабирования, для того чтобы получить график по размеру окна. Метод StartDataCanvas() имеет вид:

private void StartDataCanvas()  
{  
 aW = canvas.ActualWidth;  
 aH = canvas.ActualHeight / 2;  
 maxAB = \_cAB\_1 / Math.Exp(m \* \_time\_τ) - b \* \_time\_τ / m + \_cAB\_2;  
 maxBC = \_force\_F \* \_travel\_time \* \_travel\_time \* \_travel\_time / (6 \* \_body\_mass) -  
 \_coefficient\_f \* G \* \_travel\_time \* \_travel\_time / 2 - \_speedB \* \_travel\_time;  
 maxCE = 1000;  
 maxWidth = maxAB + maxBC + maxCE;  
 maxHeight = \_cAB\_1 / Math.Exp(\_coefficient\_μ / \_body\_mass \* \_time\_τ) - \_driving\_force / \_body\_mass + \_cAB\_2 + 100;  
 coffWidth = -aW / maxWidth;  
 coffHeight = aH / maxHeight;  
}

После всех этих операций начинает работать метод OnTimer(), отвечающий за таймер, которой по полученным уравнениям движения, зависящих от времени, считает координаты тела и добавляет из в класс polyline, и при достижении графика на третьем участке земли таймер заканчивает работу. Метод OnTimer имеет вид:

private void OnTimer(object sender, EventArgs e)  
{  
 time += 0.1;  
 if (time <= \_time\_τ)  
 {  
 xAB = \_cAB\_1 / Math.Exp(m \* time) - b \* time / m + \_cAB\_2;  
 flipXAB = aW + coffWidth \* xAB \* Math.Cos(\_angle);  
 flipYAB = aH + coffHeight \* (- xAB) \* Math.Sin(\_angle);  
 polyline.Points.Add(new Point(flipXAB, flipYAB));  
 }  
 else if(time - \_time\_τ <= \_travel\_time)  
 {  
 xСВ = \_force\_F \* time \* time \* time / (6 \* \_body\_mass) - \_coefficient\_f \* G \* time \* time / 2 -  
 \_speedB \* time;  
 polyline.Points.Add(new Point(flipXAB + coffWidth \* xСВ, flipYAB));  
 }  
 else if(flipYAB + coffHeight \* yCE <= aH)  
 {  
 xCE = \_speedС \* (time - \_time\_τ - \_travel\_time);  
 yCE = G \* (time - \_time\_τ - \_travel\_time) \* (time - \_time\_τ - \_travel\_time) / 2 ;  
 polyline.Points.Add(new Point(flipXAB + coffWidth \* (xCE +xСВ) , flipYAB + coffHeight \* yCE));  
 }  
 else   
 timer.Stop();  
 Canvas.SetLeft(ellipse, polyline.Points.Last().X - ellipse.Width / 2.0);  
 Canvas.SetTop(ellipse, polyline.Points.Last().Y - ellipse.Height / 2.0);   
}

В результате мы должны получить график как на рисунке 2

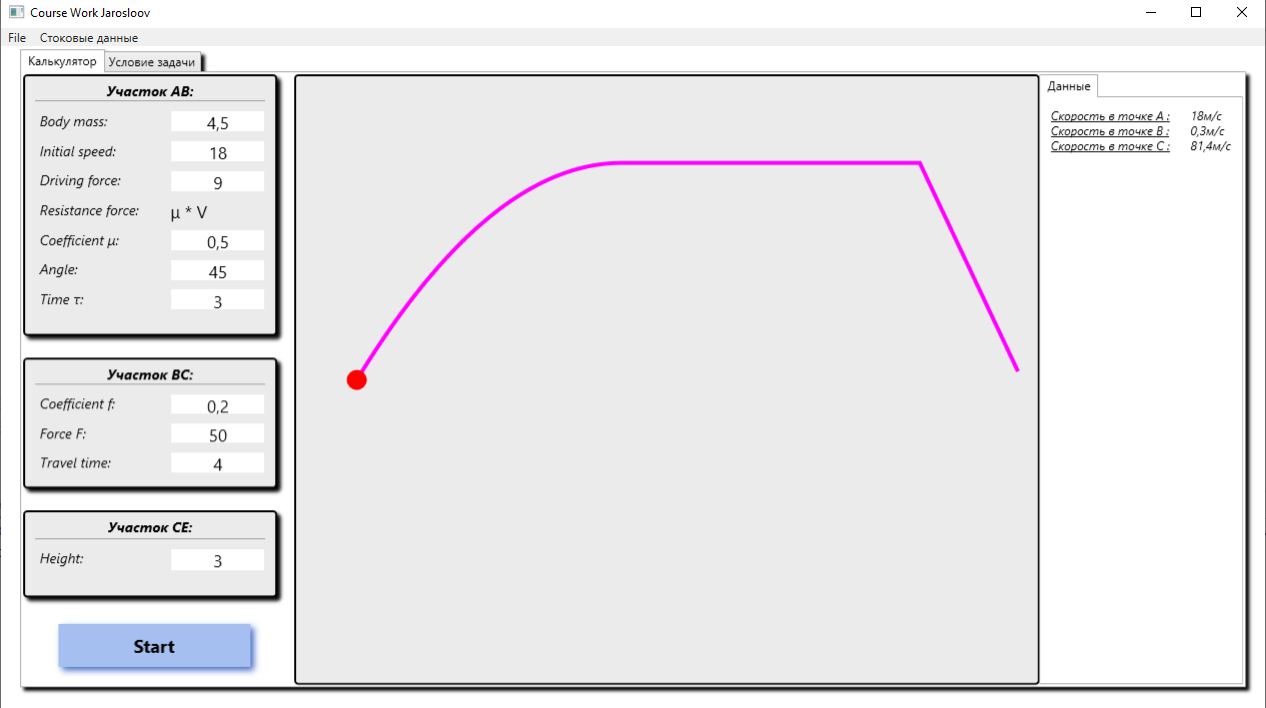


Рисунок 2 Результат выполнения программы

Также можно посмотреть условие задачи, как показано на рисунке 3

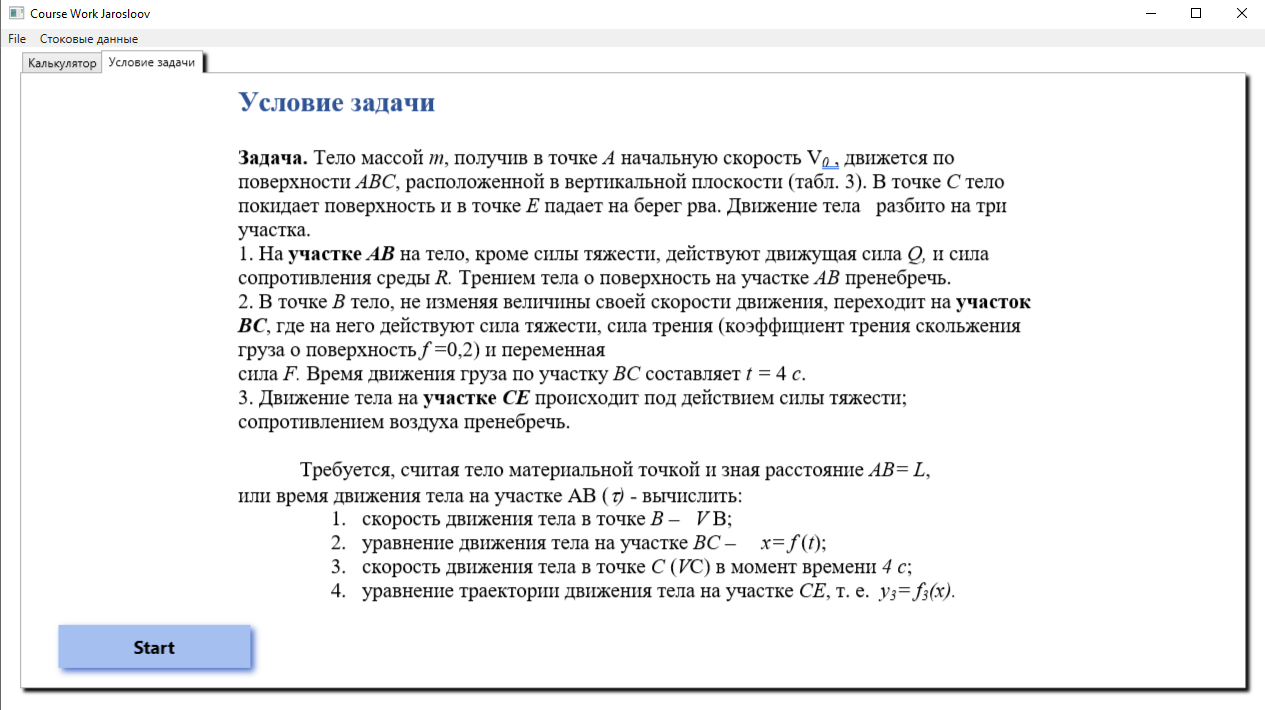


Рисунок 3 Окно с условием задачи

Весь код можно посмотреть в пункте Приложение.

# **Приложение**

## **Файл MainWindow.xaml.сs**

using System;  
using System.Diagnostics;  
using System.Linq;  
using System.Text.RegularExpressions;  
using System.Windows;  
using System.Windows.Controls;  
using System.Windows.Input;  
using System.Windows.Media;  
using System.Windows.Threading;  
  
  
namespace FirstCourseWork  
{  
 public partial class MainWindow : Window  
 {  
 private DispatcherTimer timer = new DispatcherTimer();  
 private static readonly Regex \_regex = new Regex("[^0-9.]");  
 private double time;  
 const double G = 9.80665;  
  
 // Положение графика   
 private double aW;  
 private double aH;  
 private double maxWidth;  
 private double maxHeight;  
 private double coffWidth;  
 private double coffHeight;  
 private double maxAB;  
 private double maxBC;  
 private double maxCE;  
  
 // Участок АВ  
 private double \_body\_mass;  
 private double \_initial\_speed;  
 private double \_driving\_force; //Q  
 private double \_coefficient\_μ;  
 private double \_angle;  
 private double \_time\_τ;  
  
 private double xAB;  
 private double flipXAB;  
 private double flipYAB;  
 private double \_speedB;  
  
 private double \_cAB\_1;  
 private double \_cAB\_2;  
  
 // Участок СВ  
 private double \_coefficient\_f;  
 private double \_force\_F;  
 private double \_travel\_time;  
  
 private double xСВ;  
 private double yСВ;  
 private double \_speedС;  
   
 // Участок СЕ  
 private double \_height;  
 private double xCE;  
 private double yCE;  
  
 // Проверки (защита от дурака)  
  
 private bool statusInput = true;  
  
 // Для сокращения  
  
 private double b;  
 private double m;  
   
 public MainWindow()  
 {  
 InitializeComponent();  
 timer.Tick += new EventHandler(OnTimer);  
 timer.Interval = new TimeSpan(0, 0, 0, 0, 20);  
 }  
  
 private static readonly Regex *regex* = new Regex("[^0-9,]");  
 private static bool IsTextAllowed(string text)  
 {  
 return !*regex*.IsMatch(text);  
 }  
 private void TextBox\_PreviewTextInput(object sender, TextCompositionEventArgs e)  
 {  
 e.Handled = !IsTextAllowed(e.Text);  
 }  
 private void Speed()  
 {  
 \_speedB = b / m - m \* \_cAB\_1 \* Math.Exp(-m \* \_time\_τ);  
 \_speedС = \_force\_F \* \_travel\_time \* \_travel\_time / (2 \* \_body\_mass) - \_coefficient\_f \* G \* \_travel\_time +  
 \_speedB;  
 if (\_initial\_speed <= 0 || \_speedB <= 0 || \_speedС <= 0)  
 {  
 timer.Stop();  
 Clear();  
 ellipse.Fill = new SolidColorBrush(Color.FromRgb(235, 235, 235));  
 MessageBox.Show("Нулевая скорость", "🔥Ошибка🔥",  
 MessageBoxButton.OK, MessageBoxImage.Error);  
 }  
 else  
 {  
 speedC.Text = Math.Round(\_speedС, 1) + "м/с";  
 speedB.Text = Math.Round(\_speedB, 1) + "м/с";  
 speedA.Text = Math.Round(\_initial\_speed, 1) + "м/с";  
 }  
 }  
 private void StartDataCanvas()  
 {  
 aW = canvas.ActualWidth;  
 aH = canvas.ActualHeight / 2;  
 maxAB = \_cAB\_1 / Math.Exp(m \* \_time\_τ) - b \* \_time\_τ / m + \_cAB\_2;  
 maxBC = \_force\_F \* \_travel\_time \* \_travel\_time \* \_travel\_time / (6 \* \_body\_mass) -  
 \_coefficient\_f \* G \* \_travel\_time \* \_travel\_time / 2 - \_speedB \* \_travel\_time;  
 maxCE = 1000;  
 maxWidth = maxAB + maxBC + maxCE;  
 maxHeight = \_cAB\_1 / Math.Exp(\_coefficient\_μ / \_body\_mass \* \_time\_τ) - \_driving\_force / \_body\_mass + \_cAB\_2 + 100;  
 coffWidth = -aW / maxWidth;  
 coffHeight = aH / maxHeight;  
 }  
 private void ConstantIntegrationsAB()  
 {  
 \_cAB\_1 = (\_initial\_speed - b/m)/-m;  
 \_cAB\_2 = -\_cAB\_1;  
 }  
  
 private void OnTimer(object sender, EventArgs e)  
 {  
 time += 0.1;  
 if (time <= \_time\_τ)  
 {  
 xAB = \_cAB\_1 / Math.Exp(m \* time) - b \* time / m + \_cAB\_2;  
 flipXAB = aW + coffWidth \* xAB \* Math.Cos(\_angle);  
 flipYAB = aH + coffHeight \* (- xAB) \* Math.Sin(\_angle);  
 polyline.Points.Add(new Point(flipXAB, flipYAB));  
 }  
 else if(time - \_time\_τ <= \_travel\_time)  
 {  
 xСВ = \_force\_F \* time \* time \* time / (6 \* \_body\_mass) - \_coefficient\_f \* G \* time \* time / 2 -  
 \_speedB \* time;  
 polyline.Points.Add(new Point(flipXAB + coffWidth \* xСВ, flipYAB));  
 }  
 else if(flipYAB + coffHeight \* yCE <= aH)  
 {  
 xCE = \_speedС \* (time - \_time\_τ - \_travel\_time);  
 yCE = G \* (time - \_time\_τ - \_travel\_time) \* (time - \_time\_τ - \_travel\_time) / 2 ;  
 polyline.Points.Add(new Point(flipXAB + coffWidth \* (xCE +xСВ) , flipYAB + coffHeight \* yCE));  
 }  
 else   
 timer.Stop();  
 Canvas.SetLeft(ellipse, polyline.Points.Last().X - ellipse.Width / 2.0);  
 Canvas.SetTop(ellipse, polyline.Points.Last().Y - ellipse.Height / 2.0);   
 }  
  
 private void InputData()  
 {  
 if (body\_mass.Text == "" && angle.Text == "" &&   
 coefficient\_μ.Text == "" && driving\_force.Text == "" &&   
 initial\_speed.Text == "" &&   
 time\_τ.Text == "" && height.Text == "")  
 {  
 MessageBox.Show("Не все поля были заполнены ", "🔥Ошибка🔥",  
 MessageBoxButton.OK, MessageBoxImage.Error);  
 statusInput = false;  
 }  
 else if (body\_mass.Text == "0" && angle.Text == "0" &&  
 coefficient\_μ.Text == "0" && driving\_force.Text == "0" &&  
 initial\_speed.Text == "0" &&  
 time\_τ.Text == "0" && height.Text == "0")  
 {  
 MessageBox.Show("Нулевые данные", "🔥Ошибка🔥",   
 MessageBoxButton.OK, MessageBoxImage.Error);  
 statusInput = false;  
 }  
 else if (Convert.ToDouble(angle.Text) > 90 || Convert.ToDouble(angle.Text) == 0)  
 {  
 MessageBox.Show("Невозможные углы", "🔥Ошибка🔥",  
 MessageBoxButton.OK, MessageBoxImage.Error);  
 statusInput = false;  
 }  
 else if(statusInput)  
 {  
 \_body\_mass = Convert.ToDouble(body\_mass.Text);  
 \_angle = Convert.ToDouble(angle.Text) \* Math.PI / 180;  
 \_coefficient\_μ = Convert.ToDouble(coefficient\_μ.Text);  
 \_driving\_force = Convert.ToDouble(driving\_force.Text);  
 \_initial\_speed = Convert.ToDouble(initial\_speed.Text);  
 \_time\_τ = Convert.ToDouble(time\_τ.Text);  
 \_coefficient\_f = Convert.ToDouble(coefficient\_f.Text);  
 \_force\_F = Convert.ToDouble(force\_F.Text);  
 \_travel\_time = Convert.ToDouble((travel\_time.Text));  
 \_height = Convert.ToDouble(height.Text);  
 // для краткости  
 b = \_driving\_force / \_body\_mass - G \* Math.Sin(\_angle);  
 m = \_coefficient\_μ / \_body\_mass;  
 }  
   
 }  
 private void LinkToProject(object sender, RoutedEventArgs e)  
 {  
 Process.Start("https://github.com/jarosloov/WPF2021");  
 }  
  
 private void LinkToMyProfile(object sender, RoutedEventArgs e)  
 {  
 Process.Start("https://github.com/jarosloov");  
 }  
 // Выход из программы  
 private void Exit(object sender, RoutedEventArgs e)  
 {  
 System.Windows.Application.Current.Shutdown();  
 }  
   
 private void ButtonStart(object sender, RoutedEventArgs e)  
 {  
 Clear();  
 InputData();   
 if (!statusInput) return;  
 ConstantIntegrationsAB();

ellipse.Fill = new SolidColorBrush(Color.FromRgb(157, 129, 186));  
 timer.Start();  
 Speed();   
 StartDataCanvas();

}  
  
 private void Clear()  
 {  
 polyline.Points.Clear();  
 time = xAB = xCE = xСВ = yCE = yСВ = 0;  
 speedC.Text = "........";  
 speedB.Text = "........";  
 speedA.Text = "........";  
 }  
  
 private void StokData(object sender, RoutedEventArgs e)  
 {  
 body\_mass.Text = "4,5";  
 angle.Text = "45";  
 initial\_speed.Text = "18";  
 coefficient\_μ.Text = "0,5";  
 driving\_force.Text = "9";  
 time\_τ.Text = "3";  
 resistance\_force.Text = "μ \* V";  
 coefficient\_f.Text = "0,2";  
 force\_F.Text = "50";  
 travel\_time.Text = "4";  
 height.Text = "3";  
 }  
 }  
  
}

## **Файл MainWindow.xaml:**

<Window  
 xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
 xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
 xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  
 xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"  
 xmlns:local="clr-namespace:FirstCourseWork"  
 xmlns:Properties="clr-namespace:FirstCourseWork.Properties" x:Class="FirstCourseWork.MainWindow"  
 mc:Ignorable="d"  
 x:Name="window" Title="Course Work Jarosloov" Height="720" Width="1280" Background="White">  
 <Grid>  
 <Grid.ColumnDefinitions>  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 <ColumnDefinition />  
 </Grid.ColumnDefinitions>  
 <Grid.RowDefinitions>  
  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
 <RowDefinition/>  
  
 </Grid.RowDefinitions>  
  
 <Menu VerticalAlignment="Top" Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="66">  
 <MenuItem Header="File">  
 <MenuItem Header="Ссылки">  
 <MenuItem Header="GitHub">  
 <MenuItem Click="LinkToProject" Header="Проект"></MenuItem>  
 <MenuItem Click="LinkToMyProfile" Header="Автор"></MenuItem>  
 </MenuItem>  
  
 </MenuItem>  
 <Separator></Separator>  
 <MenuItem Header="Exit" Click="Exit" VerticalAlignment="Top"></MenuItem>  
 </MenuItem>  
 <MenuItem Header="Стоковые данные" Click="StokData"></MenuItem>  
 </Menu>  
  
 <TabControl x:Name="tabControl1" Grid.Row="1" Grid.Column="1" Grid.ColumnSpan="64" Grid.RowSpan="30">  
 <TabControl.Effect>  
 <DropShadowEffect></DropShadowEffect>  
 </TabControl.Effect>  
 <TabItem>  
 <TabItem.Header >  
 <StackPanel>  
 <TextBlock Text="Калькулятор"></TextBlock>  
 </StackPanel>  
 </TabItem.Header>  
 <Grid>  
 <Grid.RowDefinitions>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
  
 </Grid.RowDefinitions>  
 <Grid.ColumnDefinitions>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 </Grid.ColumnDefinitions>  
  
 <Border x:Name="regionAB" Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="15" Grid.RowSpan="12" BorderThickness="2" CornerRadius="3" BorderBrush="Black" Background="#FFEBEBEB">  
 <Border.Effect>  
 <DropShadowEffect></DropShadowEffect>  
 </Border.Effect>  
 <Grid >  
 <Grid.RowDefinitions>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 </Grid.RowDefinitions>  
 <Grid.ColumnDefinitions>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 </Grid.ColumnDefinitions>  
 <Label Content="Участок АВ:" Grid.Column="1" Grid.ColumnSpan="24" Grid.Row="0" Grid.RowSpan="3" HorizontalAlignment="Center" FontSize="14" FontStyle="Italic" FontWeight="Bold"/>  
 <Separator Grid.Row="2" Grid.Column="1" Grid.ColumnSpan="24"></Separator>  
 <Label Content="Body mass:" Grid.Column="1" Grid.ColumnSpan="12" Grid.Row="3" Grid.RowSpan="3" HorizontalAlignment="Left" FontSize="14" FontStyle="Italic"/>  
 <Label Content="Initial speed:" Grid.Column="1" Grid.ColumnSpan="12" Grid.Row="6" Grid.RowSpan="3" HorizontalAlignment="Left" FontSize="14" FontStyle="Italic"/>  
 <Label Content="Driving force:" Grid.Column="1" Grid.ColumnSpan="12" Grid.Row="9" Grid.RowSpan="3" HorizontalAlignment="Left" FontSize="14" FontStyle="Italic"/>  
 <Label Content="Resistance force:" Grid.Column="1" Grid.ColumnSpan="12" Grid.Row="12" Grid.RowSpan="3" HorizontalAlignment="Left" FontSize="14" FontStyle="Italic"/>  
 <Label Content="Сoefficient μ:" Grid.Column="1" Grid.ColumnSpan="12" Grid.Row="15" Grid.RowSpan="3" HorizontalAlignment="Left" FontSize="14" FontStyle="Italic"/>  
 <Label Content="Angle:" Grid.Column="1" Grid.ColumnSpan="12" Grid.Row="18" Grid.RowSpan="3" HorizontalAlignment="Left" FontSize="14" FontStyle="Italic"/>  
 <Label Content="Time τ:" Grid.Column="1" Grid.ColumnSpan="12" Grid.Row="21" Grid.RowSpan="3" HorizontalAlignment="Left" FontSize="14" FontStyle="Italic"/>  
  
 <TextBox x:Name="body\_mass" Grid.Column="15" Grid.Row="3" Grid.RowSpan="3" BorderBrush="#FFEBEBEB" MaxLength="10" MaxLines="10" Background="White" FontSize="17" HorizontalContentAlignment="Center" Grid.ColumnSpan="10" Margin="1,4.333,0,3.333" PreviewTextInput="TextBox\_PreviewTextInput" />  
 <TextBox x:Name="initial\_speed" Grid.Column="15" Grid.Row="6" Grid.RowSpan="3" BorderBrush="#FFEBEBEB" MaxLength="10" MaxLines="10" Background="White" FontSize="17" HorizontalContentAlignment="Center" Grid.ColumnSpan="10" Margin="1,4.333,0,3.333" PreviewTextInput="TextBox\_PreviewTextInput" />  
 <TextBox x:Name="driving\_force" Grid.Column="15" Grid.Row="9" Grid.RowSpan="3" BorderBrush="#FFEBEBEB" MaxLength="10" MaxLines="10" Background="White" FontSize="17" HorizontalContentAlignment="Center" Grid.ColumnSpan="10" Margin="1,4.333,0,3.333" PreviewTextInput="TextBox\_PreviewTextInput" />  
 <TextBlock x:Name="resistance\_force" Grid.Column="15" Grid.Row="12" Grid.RowSpan="3" FontSize="17" Grid.ColumnSpan="10" Margin="1,4.333,0,3.333" />  
 <TextBox x:Name="coefficient\_μ" Grid.Column="15" Grid.Row="15" Grid.RowSpan="3" BorderBrush="#FFEBEBEB" MaxLength="10" MaxLines="10" Background="White" FontSize="17" HorizontalContentAlignment="Center" Grid.ColumnSpan="10" Margin="1,4.333,0,3.333" PreviewTextInput="TextBox\_PreviewTextInput" />  
 <TextBox x:Name="angle" Grid.Column="15" Grid.Row="18" Grid.RowSpan="3" BorderBrush="#FFEBEBEB" MaxLength="10" MaxLines="10" Background="White" FontSize="17" HorizontalContentAlignment="Center" Grid.ColumnSpan="10" Margin="1,4.333,0,3.333" PreviewTextInput="TextBox\_PreviewTextInput" />  
 <TextBox x:Name="time\_τ" Grid.Column="15" Grid.Row="21" Grid.RowSpan="3" BorderBrush="#FFEBEBEB" MaxLength="10" MaxLines="10" Background="White" FontSize="17" HorizontalContentAlignment="Center" Grid.ColumnSpan="10" Margin="1,4.333,0,3.333" PreviewTextInput="TextBox\_PreviewTextInput" />  
  
  
 </Grid>  
  
 </Border>  
  
 <Border x:Name="regionCB" Grid.Row="13" Grid.Column="0" Grid.ColumnSpan="15" Grid.RowSpan="6" BorderThickness="2" CornerRadius="3" BorderBrush="Black" Background="#FFEBEBEB">  
 <Border.Effect>  
 <DropShadowEffect></DropShadowEffect>  
 </Border.Effect>  
 <Grid >  
 <Grid.RowDefinitions>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 </Grid.RowDefinitions>  
 <Grid.ColumnDefinitions>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 </Grid.ColumnDefinitions>  
   
 <Label Content="Участок BC:" Grid.Column="1" Grid.ColumnSpan="24" Grid.Row="0" Grid.RowSpan="3" HorizontalAlignment="Center" FontSize="14" FontStyle="Italic" FontWeight="Bold"/>  
 <Separator Grid.Row="2" Grid.Column="1" Grid.ColumnSpan="24"></Separator>  
 <Label Content="Coefficient f:" Grid.Column="1" Grid.ColumnSpan="12" Grid.Row="3" Grid.RowSpan="3" HorizontalAlignment="Left" FontSize="14" FontStyle="Italic"/>  
 <Label Content="Force F:" Grid.Column="1" Grid.ColumnSpan="12" Grid.Row="6" Grid.RowSpan="3" HorizontalAlignment="Left" FontSize="14" FontStyle="Italic"/>  
 <Label Content="Travel time:" Grid.Column="1" Grid.ColumnSpan="12" Grid.Row="9" Grid.RowSpan="3" HorizontalAlignment="Left" FontSize="14" FontStyle="Italic"/>  
  
 <TextBox x:Name="coefficient\_f" Grid.Column="15" Grid.Row="3" Grid.RowSpan="3" BorderBrush="#FFEBEBEB" MaxLength="10" MaxLines="10" Background="White" FontSize="17" HorizontalContentAlignment="Center" Grid.ColumnSpan="10" Margin="1,4.333,0,3.333" PreviewTextInput="TextBox\_PreviewTextInput" />  
 <TextBox x:Name="force\_F" Grid.Column="15" Grid.Row="6" Grid.RowSpan="3" BorderBrush="#FFEBEBEB" MaxLength="10" MaxLines="10" Background="White" FontSize="17" HorizontalContentAlignment="Center" Grid.ColumnSpan="10" Margin="1,4.333,0,3.333" PreviewTextInput="TextBox\_PreviewTextInput" />  
 <TextBox x:Name="travel\_time" Grid.Column="15" Grid.Row="9" Grid.RowSpan="3" BorderBrush="#FFEBEBEB" MaxLength="10" MaxLines="10" Background="White" FontSize="17" HorizontalContentAlignment="Center" Grid.ColumnSpan="10" Margin="1,4.333,0,3.333" PreviewTextInput="TextBox\_PreviewTextInput" />   
 </Grid>  
 </Border>  
 <Border x:Name="regionCE" Grid.Row="20" Grid.Column="0" Grid.ColumnSpan="15" Grid.RowSpan="4" BorderThickness="2" CornerRadius="3" BorderBrush="Black" Background="#FFEBEBEB">  
 <Border.Effect>  
 <DropShadowEffect></DropShadowEffect>  
 </Border.Effect>  
 <Grid >  
 <Grid.RowDefinitions>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 </Grid.RowDefinitions>  
 <Grid.ColumnDefinitions>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 </Grid.ColumnDefinitions>  
 <Label Content="Участок CE:" Grid.Column="1" Grid.ColumnSpan="24" Grid.Row="0" Grid.RowSpan="3" HorizontalAlignment="Center" FontSize="14" FontStyle="Italic" FontWeight="Bold"/>  
 <Separator Grid.Row="2" Grid.Column="1" Grid.ColumnSpan="24"></Separator>  
 <Label Content="Height:" Grid.Column="1" Grid.ColumnSpan="12" Grid.Row="3" Grid.RowSpan="3" HorizontalAlignment="Left" FontSize="14" FontStyle="Italic"/>  
  
 <TextBox x:Name="height" Grid.Column="15" Grid.Row="3" Grid.RowSpan="3" BorderBrush="#FFEBEBEB" MaxLength="10" MaxLines="10" Background="White" FontSize="17" HorizontalContentAlignment="Center" Grid.ColumnSpan="10" Margin="1,4.333,0,3.333" PreviewTextInput="TextBox\_PreviewTextInput" />  
   
  
 </Grid>  
  
 </Border>  
  
  
 <Border Grid.Row="0" Grid.RowSpan="28" Grid.Column="16" Grid.ColumnSpan="44" BorderThickness="2" CornerRadius="3" BorderBrush="Black" Background="#FFEBEBEB">  
 <Grid >  
 <Grid.RowDefinitions>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 </Grid.RowDefinitions>  
 <Grid.ColumnDefinitions>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 </Grid.ColumnDefinitions>  
 <Canvas x:Name="canvas" Grid.Row="1" Grid.Column="1" Grid.ColumnSpan="45" Grid.RowSpan="22">  
 <Polyline x:Name="polyline" Stroke="MediumAquamarine" StrokeThickness="4" ></Polyline>  
 <Ellipse x:Name="ellipse" Fill="#FFEBEBEB" Width="20" Height="20"></Ellipse>  
 </Canvas>  
   
 </Grid>  
  
 </Border>  
   
 <TabControl Grid.Row="0" Grid.RowSpan="28" Grid.Column="60" Grid.ColumnSpan="12" >  
 <TabItem>  
 <TabItem.Header>  
 <TextBlock Text="Данные"></TextBlock>  
 </TabItem.Header>  
  
 <Grid>  
 <Grid.ColumnDefinitions>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 <ColumnDefinition></ColumnDefinition>  
 </Grid.ColumnDefinitions>  
 <Grid.RowDefinitions>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 <RowDefinition></RowDefinition>  
 </Grid.RowDefinitions>  
 <TextBlock x:Name="speedA" Grid.Row="1" Grid.Column="18" Grid.ColumnSpan="5" Grid.RowSpan="2" Text="................" FontStyle="Italic" />  
 <TextBlock Grid.Row="1" Grid.Column="1" Grid.ColumnSpan="15" Grid.RowSpan="2" Text="Скорость в точке A :" FontStyle="Italic" TextDecorations="Underline"/>  
  
 <TextBlock x:Name="speedB" Grid.Row="3" Grid.Column="18" Grid.ColumnSpan="5" Grid.RowSpan="2" Text="................" FontStyle="Italic" />  
 <TextBlock Grid.Row="3" Grid.Column="1" Grid.ColumnSpan="15" Grid.RowSpan="2" Text="Скорость в точке B :" FontStyle="Italic" TextDecorations="Underline"/>  
  
 <TextBlock x:Name="speedC" Grid.Row="5" Grid.Column="18" Grid.ColumnSpan="5" Grid.RowSpan="2" Text="................" FontStyle="Italic" />  
 <TextBlock Grid.Row="5" Grid.Column="1" Grid.ColumnSpan="15" Grid.RowSpan="2" Text="Скорость в точке C :" FontStyle="Italic" TextDecorations="Underline"/>  
  
 </Grid>  
 </TabItem>  
 </TabControl>  
   
 </Grid>  
   
 </TabItem>  
 <TabItem>  
 <TabItem.Header>  
 <StackPanel>  
 <TextBlock Text="Условие задачи"></TextBlock>  
 </StackPanel>  
 </TabItem.Header>  
 <InkCanvas>  
 <Image Source="https://user-images.githubusercontent.com/60881838/122439608-3c24b180-cfb5-11eb-8e79-f31cafa154d3.png" Margin="200,10,-256,-108"/>  
 </InkCanvas>  
   
   
 </TabItem>  
 </TabControl>  
   
   
 <Button x:Name="buttonStart" Grid.Row="28" Grid.RowSpan="2" Grid.Column="3" Grid.ColumnSpan="10" Content="Start" FontWeight="Bold" FontSize="18" BorderBrush="#FFA5C0F0" Background="#FFA5C0F0" Click="ButtonStart" >  
 <Button.Effect>  
 <DropShadowEffect BlurRadius="10" Color="#FF91AFE4"/>  
 </Button.Effect>  
   
 </Button>  
   
  
 </Grid>  
</Window>

# **Подведение итогов работы**

Работа над построением компьютерной модели графика оказалась непростой, но очень интересной. Благодаря задаче, поставленной передо мной, я улучшил свои знания в «Теоретической механике» и произвел дополнительную подготовку к экзамену по двум дисциплинам. Также мной было освоено написание самой работы, а именно изучена работа с формулами в приложении «Microsoft Office Word».

# **Список литературы**

1. Бухгольц Н.Н. Основной курс теоретической механики.
2. Ч. Петцольд «Microsoft Windows Presentation Foundation: базовый курс»
3. Макконнелл С. «Совершенный код»