

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Приладобудівний факультет

(повна назва інституту/факультету)

Кафедра приладобудування

(повна назва кафедри)

«На правах рукопису»

УДК 681.121

«До захисту допущено»

Завідувач кафедри

_____ М. Д. Гераїмчук
(підпис) (ініціали, прізвище)

“ ___ ” _____ 20__ р.

**Магістерська дисертація
на здобуття освітнього ступеня магістр**

зі спеціальності (спеціалізації) 151 Автоматизація та комп'ютерно-інтегровані
(код і назва спеціальності)

технології (Комп'ютерно-інтегровані технології та системи точної механіки)

на тему: Дослідження динамічних характеристик вимірювальних

перетворювачів витрати рідини

Виконав: студент 2 курсу, групи ПМ-61м
(шифр групи)

Місяць Андрій Сергійович

(прізвище, ім'я, по батькові)

_____ (підпис)

Науковий керівник доцент, к.т.н. Писарець А. В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант _____

(назва розділу)

_____ (науковий ступінь, вчене звання, прізвище, ініціали)

_____ (підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018 року

**Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»**

Факультет (інститут) _____ Приладобудівний _____
(повна назва)

Кафедра _____ Приладобудування _____
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-науковою програмою

Спеціальність (спеціалізація) 151 Автоматизація та комп'ютерно-інтегровані технології (Комп'ютерно-інтегровані технології та системи точної механіки) _____
(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ М. Д. Гераїмчук
(підпис) (ініціали, прізвище)

« ____ » _____ 2018 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Місяцю Андрію Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема дисертації *Дослідження динамічних характеристик вимірвальних перетворювачів витрати рідини*
науковий керівник дисертації _____ Писарець А. В., к.т.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « ____ » _____ 20__ р. № _____

2. Строк подання студентом дисертації _____ 10 травня 2018 р.

3. Об'єкт дослідження процес визначення динамічної характеристики витратоміра

4. Предмет дослідження (вихідні дані для магістерської дисертації за освітньо-професійною програмою) _____ гідродинамічний витратомір

5. Перелік завдань, які потрібно розробити Вступ. Огляд літератури за темою, вибір та обґрунтування напрямів та методів дослідження. Розробка

математичної моделі перетворювача витрати. Визначення динамічних характеристик перетворювача витрати. Моделювання роботи витратовимірювальної установки. Стендові дослідження витратоміра. Аналіз і узагальнення результатів досліджень

6. Орієнтовний перелік ілюстративного (графічного) матеріалу: презентаційний аркуш ф. А1; матеріали досліджень 6 арк. Ф. А1; презентація у Microsoft Power Point

7. Орієнтовний перелік публікацій 2 доповіді на науково-технічних конференціях, 1 стаття у фаховому виданні

–

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 14.03.2018 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Огляд літератури за темою	14.03.18 – 29.03.18	
2	Розробка математичної моделі перетворювача витрати	30.03.18 – 16.04.18	
3	Визначення динамічних характеристик перетворювача витрати	10.04.18 – 20.04.18	
4	Розробка вимірювальної схеми для проведення напівнатурних досліджень	10.04.18 – 21.04.18	
5	Моделювання роботи витратовимірювальної установки	16.04.18 – 26.04.18	
6	Стендові дослідження витратоміра	23.04.18 – 26.04.18	
7	Аналіз отриманих результатів	27.04.18 – 06.05.18	
8	Оформлення МД та її графічної частини	07.05.18	

9	Передача МД на перевірку науковому керівнику	08.05.18	
10	Передача матеріалів МД на перевірку виявлення збігів/схожості текстів	09.05.18	
11	Представити МД на рецензію	10.05.18	
12	Представити МД на затвердження зав. кафедри	11.05.18	
13	Передача електронної версії МД о бібліотеки	17.05.18	
14	Представити МД до екзаменаційної комісії НТУУ «КПІ ім. Ігоря Сікорського»	18.05.18	
15			

Студент

(підпис)

А. С. Місяць

(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

А. В. Писарець

(ініціали, прізвище)

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Хансуваров К.И., Цейтлин В.Г. Техника измерения давления, расхода, количества и уровня жидкости, газа и пара: Учебное пособие для техникумов. – М.: Издательство стандартов, 1990. 287 с.
2. Бирюков Б. В., Данилов М. А., Кивилис С. С. Испытания расходомеров. – М.: Издательство стандартов, 1987. – 240 с., с илл.
3. Бирюков Б. В., Данилов М. А., Кивилис С. С. Средства испытаний расходомеров. – М.: Энергоатомиздат, 1983. – 112 с., ил. – (Б-ка по автоматике; Вып. 638).
4. Бирюков Б. В., Данилов М. А., Кивилис С. С. Испытательные расходомерные установки. – М.: Энергия, 1976. – 145с.
5. Бирюков Б. В., Данилов М. А., Кивилис С. С. Точные измерения расхода жидкостей. – М.: Энергия, 1977. – 144с.
6. Данилов М. А. О Возможном методе испытания расходомеров большой производительности. – В кн.: Метрологические исследования в области измерения расхода и количества веществ. Труды метрологических институтов СССР, 1972, вып. 135 (195), с. 40-45.
7. Коробко І. В. Вплив гаусової кривизни поверхні чутливого елементу і гідродинамічних характеристик потоку на метрологічні показники перетворювачів витрати / І. В. Коробко, А. В. Писарець // Наукові нотатки. Міжвузівський збірник (за галузями знань “Машинобудування та металообробка”, “Інженерна механіка”, “Металургія та матеріалознавство”). – 2015. – №48. – С.116 – 120.
8. Валиев Э. Р. Исследование динамических характеристик систем измерения расхода природного газа на базе стандартной диафрагмы в условиях эксплуатации / Э. Р. Валиев, Н. А. Николаев // Вестник Казанского Технологического Университета. – 2012. – № 9. – С. 246-248.

9. Коробко І. В. Наукові принципи творення засобів вимірювання витрати плинних потоків технологічних мереж : дис. ... д-ра техн. наук. : 05.11.01 – прилади та методи вимірювання механічних величин / І. В. Коробко. - К., 2014. - 413 ст.
10. Коробко І.В., Сергеев С.П., Нікітін О.К. Науковий звіт по темі "Розробка і виготовлення первинних перетворювачів технологічних параметрів енергетичних установок". Київський Політехнічний інститут. 1984.
11. Коробко І.В., Сергеев С.П., Нікітін О.К. Звіт по темі "Удосконалення експлуатаційних і метрологічних характеристик, виготовлення та поставка вимірювальних перетворювачів технологічних параметрів енергетичних установок". Київський Політехнічний інститут. 1987.
12. Коробко І.В., Сергеев С.П., Нікітін О.К. Звіт по темі "Удосконалення експлуатаційних і метрологічних характеристик, виготовлення та поставка вимірювальних перетворювачів технологічних параметрів енергетичних установок". Київський Політехнічний інститут. 1988.
13. Коробко І.В., Сергеев С.П., Барковська Н.І. Визначення модуля пружності чутливих елементів гідродинамічних вимірювальних перетворювачів витрати. Вісник Київського Політехнічного інституту. Приладобудування, вип.. 16, 1986.
14. Коробко І.В., Сергеев С.П. До питання визначення конструкції пружного елемента гідродинамічного витратоміра. Депон. ЦНІТЕІ приладобудування, №3167, м. Москва, 1986р.
15. Коробко І.В., Барковська Н.І., Нікітін О.К., Сергеев С.П. Вибір форми пружного елемента імпульсного вимірювального перетворювача витрати гідродинамічного типу. Вісник Київського Політехнічного інституту. Приладобудування, вип. 17, 1987.
16. Коробко І.В., Сергеев С.П., Нікітін О.К. Тези доповіді "Гідродинамічні витратоміри". Зб. тез МНТК "Удосконалення засобів витрати рідини, газу, пари", Санкт-Петербург 1992р.

- 17.Коробко І.В. Визначення коливальної маси чутливого елемента вимірювального перетворювача витрат гідродинамічного типу. Праці Таврійської державної агротехнічної академії. - Вип.4.Том 22, м.Мелітополь,-2003,-С.87-93.
- 18.Коробко І. В. Моделювання роботи гідродинамічних вимірювальних перетворювачів витрати та кількості паливно-енергетичних ресурсів / І. В. Коробко, А. В. Писарець // XIII Міжнародна науково-технічна конференція “Прогресивна техніка та технологія”. Севастополь, 11 – 15 червня 2012 р. – Севастополь.
- 19.ARDUINO UNO REV3 [Електронний ресурс] – Режим доступу до ресурсу: <https://store.arduino.cc/usa/arduino-uno-rev3>.
- 20.Uno Плати Ардуино [Електронний ресурс] – Режим доступу до ресурсу: <https://doc.arduino.ua/ru/hardware/Uno>.
- 21.Реле модуль подключение к Arduino [Електронний ресурс] – Режим доступу до ресурсу: <http://zelectro.cc/relayModule>.
- 22.HX711 Двухканальный модуль датчиков веса (тензодатчиков) Источник <https://arduino.ua/prod1147-hx711-dual-channel-weighing-sensor-module> [Електронний ресурс] – Режим доступу до ресурсу: <https://arduino.ua/prod1147-hx711-dual-channel-weighing-sensor-module>.
- 23.Бабаков Н. А. Теория автоматического управления: Учеб. для вузов по спец. «Автоматика и телемеханика». В 2-х ч. Ч. I. Теория линейных систем автоматического управления / Н. А. Бабаков, А. А. Воронов, А. А. Воронова и др.; Под ред. А. А., Воронова.—2-е изд., перераб. и доп. — М.: Высш. шк., 1986. — 367 е., ил.
- 24.Писарець А. В. Дослідження динамічних характеристик витратомірів / А. В. Писарець, А. С. Місяць // Збірник тез доповідей XVII Міжнародної науково-технічної конференції "ПРИЛАДОБУДУВАННЯ: стан і перспективи", 15 – 16 травня 2018 р., м. Київ, ПФФ, НТУУ «КПІ». – 2018. – С. 195 – 196.

25. Місяць А. С. Автоматизація досліджень динамічних характеристик витратомірів / А. С. Місяць, А. В. Писарець // Збірник статей XI науково-практичної конференції студентів та аспірантів "ПОГЛЯД У МАЙБУТНЄ ПРИЛАДОБУДУВАННЯ", 15 – 16 травня 2018 р., м. Київ, ПБФ, НТУУ «КПІ». – 2018. – С. 516 – 519.

ДОДАТКИ

ДОДАТОК А

Лістинг коду для мікроконтролеру

```
#include <Q2HX711.h>
Q2HX711 hx711(2, 3);
unsigned long previousMillis = 0;
unsigned long currentMillis = 0;
unsigned long am = 0;
byte state_13      = LOW;
byte state_7       = HIGH;
byte pin13         = 13;
byte pin7          = 7;
String mode        = "";
String response    = "";
int impulse        = 0;
int pause          = 0;
void setup()
{
  Serial.begin(38400);
  pinMode(pin13, OUTPUT);
  pinMode(pin7, OUTPUT);
  digitalWrite(pin7, HIGH);
}
void loop()
{
  if (Serial.available() > 0)
  {
```

```

String input = Serial.readString();
byte index    = input.indexOf(';');
mode         = input.substring(0, index);
byte impulseIndex = input.indexOf(';', index + 1);
impulse      = input.substring(index + 1, impulseIndex).toInt();
pause       = input.substring(impulseIndex + 1).toInt();
}
currentMillis = millis();
if (mode == "impulse")
{
  if (am == 0)
  {
    am = millis();
    while(500 > millis() - am)
    {
      response = (String) hx711.read() + ";" + (String) millis() + ";" + state_7;
      Serial.println(response);
    }
  }
}
currentMillis = millis();
if (currentMillis - previousMillis >= impulse)
{
  previousMillis = currentMillis;
  if (state_7 == LOW)
  {
    response = (String) hx711.read() + ";" + (String) millis() + ";" + state_7;
    Serial.println(response);
    state_13 = LOW;
    state_7  = HIGH;
    response = (String) hx711.read() + ";" + (String) millis() + ";" + state_7;

```

```

        Serial.println(response);
        digitalWrite(pin13, state_13);
        digitalWrite(pin7, state_7);
        unsigned long pm = millis();
        while(2000 > millis() - pm)
        {
            response = (String) hx711.read() + ";" + (String) millis() + ";" +
state_7;
            Serial.println(response);
        }
        mode = "";
        impulse = 0;
        am = 0;
        return;
    }
    response = (String) hx711.read() + ";" + (String) millis() + ";" + state_7;
    Serial.println(response);
    state_13 = HIGH;
    state_7 = LOW;
    response = (String) hx711.read() + ";" + (String) millis() + ";" + state_7;
    Serial.println(response);
    digitalWrite(pin13, state_13);
    digitalWrite(pin7, state_7);
}
else
{
    response = (String) hx711.read() + ";" + (String) millis() + ";" + state_7;
    Serial.println(response);
}
}

```

```

if (mode == "serie")
{
  if (currentMillis - previousMillis >= impulse && state_7 == LOW)
  {
    previousMillis = currentMillis;
    response = (String) hx711.read() + ";" + (String) millis() + ";" + state_7;
    Serial.println(response);
    state_13 = LOW;
    state_7 = HIGH;
    response = (String) hx711.read() + ";" + (String) millis() + ";" + state_7;
    Serial.println(response);
    digitalWrite(pin13, state_13);
    digitalWrite(pin7, state_7);
  }
  if (currentMillis - previousMillis >= pause && state_7 == HIGH)
  {
    previousMillis = currentMillis;
    response = (String) hx711.read() + ";" + (String) millis() + ";" + state_7;
    Serial.println(response);
    state_13 = HIGH;
    state_7 = LOW;
    response = (String) hx711.read() + ";" + (String) millis() + ";" + state_7;
    Serial.println(response);
    digitalWrite(pin13, state_13);
    digitalWrite(pin7, state_7);
  }
  else
  {
    response = (String) hx711.read() + ";" + (String) millis() + ";" + state_7;
    Serial.println(response);
  }
}

```

```

    }
}
if (mode == "stationary")
{
    response = (String) hx711.read() + ";" + (String) millis() + ";" + state_7;
    Serial.println(response);
    state_13 = HIGH;
    state_7 = LOW;
    response = (String) hx711.read() + ";" + (String) millis() + ";" + state_7;
    Serial.println(response);
    digitalWrite(pin13, state_13);
    digitalWrite(pin7, state_7);
}
if (mode == "stop")
{
    state_13 = LOW;
    state_7 = HIGH;
    digitalWrite(pin13, state_13);
    digitalWrite(pin7, state_7);
    impulse = 0;
    pause = 0;
    mode = "";
    return;
}
}

```

ДОДАТОК Б

Лістинг комп'ютерного додадку

Головний клас Program – точка входу в додаток:

```
using System;
using System.Windows.Forms;
namespace Master
{
    static class Program
    {
        /// <summary>
        /// Головна точка входу в додаток.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new MainForm());
        }
    }
}
```

Клас MainForm(). Клас обробки даних

```
using System;
using System.Windows.Forms;
using System.IO.Ports;
```

```

using System.Collections.Generic;
using System.Linq;
using System.IO;
using System.Text;
namespace Master
{
    public partial class MainForm : Form
    {
        public SerialPort port;
        List<int> kgs = new List<int>();
        List<int> times = new List<int>();
        List<int> state = new List<int>();
        List<double> filterKgs = new List<double>();
        double startKG = 0;
        int startTime = 0;
        double koeff = 0.1;
        public MainForm()
        {
            InitializeComponent();
            CheckRadioButtons();
            ConnectToPort();
        }
        public void ConnectToPort()
        {
            port = new SerialPort
            {
                PortName = "COM4",
                Parity = Parity.None,
                StopBits = StopBits.One,
                DataBits = 8,
            }
        }
    }
}

```



```

        BaudRate = 38400
    };
    port.DataReceived += new
SerialDataReceivedEventHandler(serialDataReceivedEventHandler);
    try
    {
        port.Open();
    }
    catch (System.IO.IOException e)
    {
        MessageBox.Show(e.Message.ToString());
        this.Close();
    }
}
private void serialDataReceivedEventHandler(object sender,
SerialDataReceivedEventArgs e)
{
    SerialPort sData = sender as SerialPort;
    string recData = sData.ReadLine();
    int dataKG;
    int dataTIME;
    int dataOPEN;
    string[] subStrings = recData.Split(';');
    if (Int32.TryParse(subStrings[0], out dataKG))
    {
        kgs.Add(dataKG);
    }
    if (Int32.TryParse(subStrings[1], out dataTIME))
    {
        times.Add(dataTIME);
    }
}

```

```

    }
    if (Int32.TryParse(subStrings[2], out dataOPEN))
    {
        state.Add(dataOPEN);
    }
    if (startTime == 0)
        startTime = dataTIME;
    if (startKG == 0)
        startKG = dataKG;
    startKG = startKG * (1 - koeff) + dataKG * koeff;
    filterKgs.Add(startKG);
    chart.Invoke((MethodInvoker)delegate {
chart.Series[0].Points.AddXY(dataTIME - startTime, startKG); });
    if (dataOPEN == 1)
        chartValve.Invoke((MethodInvoker)delegate {
chartValve.Series[0].Points.AddXY(dataTIME - startTime, 0); });
    else
        chartValve.Invoke((MethodInvoker)delegate {
chartValve.Series[0].Points.AddXY(dataTIME - startTime, 1); });
    chart.Invoke((MethodInvoker)delegate
    {
        chart.ChartAreas[0].AxisY.Maximum = filterKgs.Max() + 2000;
        chart.ChartAreas[0].AxisY.Minimum = filterKgs.Min() - 2000;
    });
}
private void CheckRadioButtons()
{
    if (seriesOfImpulsesRB.Checked)
    {
        adjustmentDurationOfImpulseNUD.Enabled = true;

```

```

        adjustmentDurationOfPauseNUD.Enabled = true;
    }
    if (singleImpulseRB.Checked)
    {
        adjustmentDurationOfImpulseNUD.Enabled = true;
        adjustmentDurationOfPauseNUD.Enabled = false;
    }
    if (stationaryRB.Checked)
    {
        adjustmentDurationOfImpulseNUD.Enabled = false;
        adjustmentDurationOfPauseNUD.Enabled = false;
    }
}
private void startButton_Click(object sender, EventArgs e)
{
    decimal                adjustmentDurationOfImpulse           =
adjustmentDurationOfImpulseNUD.Value;
    decimal                adjustentDurationOfPause              =
adjustmentDurationOfPauseNUD.Value;
    string modeOfOperation = "";
    if (stationaryRB.Checked)
        modeOfOperation = stationaryRB.Tag.ToString();
    if (singleImpulseRB.Checked)
        modeOfOperation = singleImpulseRB.Tag.ToString();
    if (seriesOfImpulsesRB.Checked)
        modeOfOperation = seriesOfImpulsesRB.Tag.ToString();
    port.Write(
        modeOfOperation + ";" +
        adjustmentDurationOfImpulse + ";" +
        adjustentDurationOfPause

```

```

    );
}
private void stopButton_Click(object sender, EventArgs e)
{
    port.Write("stop;");
}
private void stationaryRB_CheckedChanged(object sender, EventArgs e)
{
    if (stationaryRB.Checked)
    {
        adjustmentDurationOfImpulseNUD.Enabled = false;
        adjustmentDurationOfPauseNUD.Enabled = false;
    }
}
private void singleImpulseRB_CheckedChanged(object sender, EventArgs e)
{
    if (singleImpulseRB.Checked)
    {
        adjustmentDurationOfImpulseNUD.Enabled = true;
        adjustmentDurationOfPauseNUD.Enabled = false;
    }
}
private void seriesOfImpulsesRB_CheckedChanged(object sender, EventArgs
e)
{
    if (seriesOfImpulsesRB.Checked)
    {
        adjustmentDurationOfImpulseNUD.Enabled = true;
        adjustmentDurationOfPauseNUD.Enabled = true;
    }
}

```

```

}
private void clearLists_Click(object sender, EventArgs e)
{
    times.Clear();
    kgs.Clear();
    state.Clear();
    filterKgs.Clear();
    startTime = 0;
    startKG = 0;
    foreach (var series in chart.Series)
    {
        series.Points.Clear();
    }
    foreach (var series in chartValve.Series)
    {
        series.Points.Clear();
    }
}
private void kNUD_ValueChanged(object sender, EventArgs e)
{
    koeff = (double) kNUD.Value;
}
private void qualityControlButton_Click(object sender, EventArgs e)
{
    double h_yst = filterKgs[filterKgs.Count - 1];
    double minus_delta = h_yst - ((h_yst * (double)faultNUD.Value) / 100);
    double plus_delta = h_yst + ((h_yst * (double)faultNUD.Value) / 100);
    double x_minus = 0;
    double y_minus = 0;
    double x_plus = 0;
}

```

```

double y_plus = 0;
double pp_time = 0;
double narost_time = 0;
chart.Series[1].Points.Clear();
chart.Series[2].Points.Clear();
chart.Series[3].Points.Clear();
chart.Series[4].Points.Clear();
foreach (var time in times)
{
    chart.Series[1].Points.AddXY(time - startTime, minus_delta);
    chart.Series[2].Points.AddXY(time - startTime, plus_delta);
    chart.Series[4].Points.AddXY(time - startTime, h_yst);
}
for (var i = times.Count - 1; i > 0; i--)
{
    double A1 = filterKgs[i] - filterKgs[i - 1];
    double B1 = (times[i - 1] - startTime) - (times[i] - startTime);
    double A2 = plus_delta - plus_delta;
    double B2 = (times[i - 1] - startTime) - (times[i] - startTime);
    double delta = (A1 * B2) - (A2 * B1);
    if (delta != 0 && filterKgs[i - 1] > plus_delta && filterKgs[i] <
plus_delta)
    {
        double C1 = (A1 * (times[i - 1] - startTime)) + (B1 * filterKgs[i - 1]);
        double C2 = (A2 * (times[i - 1] - startTime)) + (B2 * plus_delta);
        x_plus = (B2 * C1 - B1 * C2) / delta;
        y_plus = (A1 * C2 - A2 * C1) / delta;
        break;
    }
}
}

```

```

for (var i = times.Count - 1; i > 0; i--)
{
    double A1 = filterKgs[i] - filterKgs[i - 1];
    double B1 = (times[i - 1] - startTime) - (times[i] - startTime);
    double A2 = minus_delta - minus_delta;
    double B2 = (times[i - 1] - startTime) - (times[i] - startTime);
    double delta = (A1 * B2) - (A2 * B1);

    if (delta != 0 && filterKgs[i - 1] < minus_delta && filterKgs[i] >
minus_delta)
    {
        double C1 = (A1 * (times[i - 1] - startTime)) + (B1 * filterKgs[i - 1]);
        double C2 = (A2 * (times[i - 1] - startTime)) + (B2 * minus_delta);
        x_minus = (B2 * C1 - B1 * C2) / delta;
        y_minus = (A1 * C2 - A2 * C1) / delta;
        break;
    }
}

for (var i = 0; i < times.Count - 1; i++)
{
    double A1 = filterKgs[i + 1] - filterKgs[i];
    double B1 = (times[i] - startTime) - (times[i + 1] - startTime);
    double A2 = h_yst - h_yst;
    double B2 = (times[i] - startTime) - (times[i + 1] - startTime);
    double delta = (A1 * B2) - (A2 * B1);
    if (delta != 0 && ((filterKgs[i] < h_yst && filterKgs[i + 1] > h_yst) ||
(filterKgs[i] > h_yst && filterKgs[i + 1] < h_yst)))
    {
        double C1 = (A1 * (times[i] - startTime)) + (B1 * filterKgs[i]);
        double C2 = (A2 * (times[i] - startTime)) + (B2 * h_yst);

```

```

        double x_per = (B2 * C1 - B1 * C2) / delta;
        double y_per = (A1 * C2 - A2 * C1) / delta;
        narost_time = x_per;
        break;
    }
}
if (x_minus > x_plus)
{
    chart.Series[3].Points.AddXY(x_minus, y_minus);
    pp_time = x_minus;
}
else
{
    chart.Series[3].Points.AddXY(x_plus, y_plus);
    pp_time = x_plus;
}
double perereg = ((filterKgs.Max() - h_yst) / filterKgs.Max()) * 100;
int maxTimeIndex = filterKgs.IndexOf(filterKgs.Max());
MessageBox.Show(
    "Час перехідного процесу: " + (pp_time).ToString() + " мс." + "\n" +
    "Перерегулювання: " + (perereg).ToString() + " %" + "\n" +
    "Час досягнення максимуму: " + (times[maxTimeIndex] -
startTime).ToString() + " мс." + "\n" +
    "Час наростання ПП: " + (narost_time).ToString() + " мс.",
    "Оцінка якості перехідного процесу",
    MessageBoxButtons.OK
);
}
private void dataToFileButton_Click(object sender, EventArgs e)

```



```

    {
        string n = string.Format("research-{0:yyyy-MM-dd_hh-mm-ss}.txt",
DateTime.Now);
        string path = @"e:\research\" + n;
        try
        {
            // Delete the file if it exists.
            if (File.Exists(path))
            {
                File.Delete(path);
            }
            // Create the file.
            using (StreamWriter file = new StreamWriter(path))
            {
                for (var i = 0; i < times.Count; i++)
                {
                    if (state[i] == 0)
                    {
                        file.WriteLine((times[i] - startTime) + ";" + filterKgs[i] + ";" + 1
+ ";"");
                    }
                    else
                    {
                        file.WriteLine((times[i] - startTime) + ";" + filterKgs[i] + ";" + 0
+ ";"");
                    }
                }
            }
        }
        catch (Exception ex)

```

```

        {
            Console.WriteLine(ex.ToString());
        }
    }
}
}
}

```

Ініціалізація форми

```
namespace Master
```

```

{
    partial class MainForm
    {
        /// <summary>
        /// Обязательная переменная конструктора.
        /// </summary>
        private System.ComponentModel.IContainer components = null;
        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        /// <param name="disposing">истинно, если управляемый ресурс должен
        быть удален; иначе ложно.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
        #region Код, автоматически созданный конструктором форм Windows
        /// <summary>

```

```

/// Требуемый метод для поддержки конструктора — не изменяйте
/// содержимое этого метода с помощью редактора кода.
/// </summary>
private void InitializeComponent()
{
    System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea1
= new System.Windows.Forms.DataVisualization.Charting.ChartArea();
    System.Windows.Forms.DataVisualization.Charting.Legend legend1 =
new System.Windows.Forms.DataVisualization.Charting.Legend();
    System.Windows.Forms.DataVisualization.Charting.Series series1 = new
System.Windows.Forms.DataVisualization.Charting.Series();
    System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea2
= new System.Windows.Forms.DataVisualization.Charting.ChartArea();
    System.Windows.Forms.DataVisualization.Charting.Series series2 = new
System.Windows.Forms.DataVisualization.Charting.Series();
    this.label2 = new System.Windows.Forms.Label();
    this.adjustmentDurationOfImpulseNUD = new
System.Windows.Forms.NumericUpDown();
    this.adjustmentDurationOfPauseNUD = new
System.Windows.Forms.NumericUpDown();
    this.label3 = new System.Windows.Forms.Label();
    this.label4 = new System.Windows.Forms.Label();
    this.startButton = new System.Windows.Forms.Button();
    this.stationaryRB = new System.Windows.Forms.RadioButton();
    this.singleImpulseRB = new System.Windows.Forms.RadioButton();
    this.seriesOfImpulsesRB = new System.Windows.Forms.RadioButton();
    this.stopButton = new System.Windows.Forms.Button();
    this.label1 = new System.Windows.Forms.Label();
    this.label5 = new System.Windows.Forms.Label();
    this.label6 = new System.Windows.Forms.Label();

```

```

        this.chart = new
System.Windows.Forms.DataVisualization.Charting.Chart();
        this.chartValve = new
System.Windows.Forms.DataVisualization.Charting.Chart();
        this.serialData = new System.Windows.Forms.RichTextBox();
        this.clearLists = new System.Windows.Forms.Button();
        this.label7 = new System.Windows.Forms.Label();
        this.kNUD = new System.Windows.Forms.NumericUpDown();
        ((System.ComponentModel.ISupportInitialize)(this.adjustmentDurationOfImpulse
NUD)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.adjustmentDurationOfPauseN
UD)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.chart)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.chartValve)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.kNUD)).BeginInit();
        this.SuspendLayout();
        //
        // label2
        //
        this.label2.AutoSize = true;
        this.label2.Font = new System.Drawing.Font("Times New Roman", 16.2F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte)(204)));
        this.label2.ForeColor =
System.Drawing.Color.FromArgb(((int)((byte)(81))), ((int)((byte)(86))),
((int)((byte)(89))));
        this.label2.Location = new System.Drawing.Point(24, 195);
        this.label2.Name = "label2";
        this.label2.Size = new System.Drawing.Size(337, 32);
        this.label2.TabIndex = 5;

```

```

this.label2.Text = "Регулювання тривалістю";
//
// adjustmentDurationOfImpulseNUD
//
this.adjustmentDurationOfImpulseNUD.Font = new
System.Drawing.Font("Times New Roman", 13.8F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
(byte)(204));
this.adjustmentDurationOfImpulseNUD.ForeColor =
System.Drawing.Color.FromArgb(((int)((byte)(81))), ((int)((byte)(86))),
(int)((byte)(89)));
this.adjustmentDurationOfImpulseNUD.Location = new
System.Drawing.Point(134, 245);
this.adjustmentDurationOfImpulseNUD.Maximum = new decimal(new
int[] {
2000,
0,
0,
0});
this.adjustmentDurationOfImpulseNUD.Minimum = new decimal(new
int[] {
10,
0,
0,
0});
this.adjustmentDurationOfImpulseNUD.Name =
"adjustmentDurationOfImpulseNUD";
this.adjustmentDurationOfImpulseNUD.Size = new
System.Drawing.Size(183, 34);
this.adjustmentDurationOfImpulseNUD.TabIndex = 6;

```

```

this.adjustmentDurationOfImpulseNUD.Value = new decimal(new int[] {
    10,
    0,
    0,
    0});
//
// adjustmentDurationOfPauseNUD
//
this.adjustmentDurationOfPauseNUD.Font = new
System.Drawing.Font("Times New Roman", 13.8F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
(byte)(204));
this.adjustmentDurationOfPauseNUD.ForeColor =
System.Drawing.Color.FromArgb(((int)((byte)(81))), ((int)((byte)(86))),
(int)((byte)(89))));
this.adjustmentDurationOfPauseNUD.Location = new
System.Drawing.Point(134, 295);
this.adjustmentDurationOfPauseNUD.Maximum = new decimal(new int[]
{
    20000,
    0,
    0,
    0});
this.adjustmentDurationOfPauseNUD.Minimum = new decimal(new int[] {
    10,
    0,
    0,
    0});
this.adjustmentDurationOfPauseNUD.Name =
"adjustmentDurationOfPauseNUD";

```

```

        this.adjustmentDurationOfPauseNUD.Size = new
System.Drawing.Size(183, 34);
        this.adjustmentDurationOfPauseNUD.TabIndex = 7;
        this.adjustmentDurationOfPauseNUD.Value = new decimal(new int[] {
10,
0,
0,
0});
//
// label3
//
        this.label3.AutoSize = true;
        this.label3.Font = new System.Drawing.Font("Times New Roman", 13.8F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)(204)));
        this.label3.ForeColor =
System.Drawing.Color.FromArgb(((int)(((byte)(81)))), ((int)(((byte)(86)))),
((int)(((byte)(89)))));
        this.label3.Location = new System.Drawing.Point(25, 245);
        this.label3.Name = "label3";
        this.label3.Size = new System.Drawing.Size(104, 27);
        this.label3.TabIndex = 8;
        this.label3.Text = "Імпульса";
//
// label4
//
        this.label4.AutoSize = true;
        this.label4.Font = new System.Drawing.Font("Times New Roman", 13.8F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)(204)));

```

```

        this.label4.ForeColor =
System.Drawing.Color.FromArgb(((int)(((byte)(81)))), ((int)(((byte)(86)))),
((int)(((byte)(89)))));
        this.label4.Location = new System.Drawing.Point(25, 297);
        this.label4.Name = "label4";
        this.label4.Size = new System.Drawing.Size(73, 27);
        this.label4.TabIndex = 9;
        this.label4.Text = "Паузи";
        //
        // startButton
        //
        this.startButton.BackColor = System.Drawing.Color.DodgerBlue;
        this.startButton.FlatAppearance.BorderSize = 0;
        this.startButton.Font = new System.Drawing.Font("Times New Roman",
13.8F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)(204)));
        this.startButton.ForeColor = System.Drawing.Color.White;
        this.startButton.Location = new System.Drawing.Point(23, 653);
        this.startButton.Name = "startButton";
        this.startButton.Size = new System.Drawing.Size(168, 61);
        this.startButton.TabIndex = 27;
        this.startButton.Text = "ПУСК";
        this.startButton.UseVisualStyleBackColor = false;
        this.startButton.Click +=
new
System.EventHandler(this.startButton_Click);
        //
        // stationaryRB
        //
        this.stationaryRB.AutoSize = true;
        this.stationaryRB.Checked = true;

```



```

        this.stationaryRB.Font = new System.Drawing.Font("Times New Roman",
13.8F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)(204)));
        this.stationaryRB.ForeColor =
System.Drawing.Color.FromArgb(((int)(((byte)(81)))), ((int)(((byte)(86)))),
((int)(((byte)(89)))));
        this.stationaryRB.Location = new System.Drawing.Point(29, 72);
        this.stationaryRB.Name = "stationaryRB";
        this.stationaryRB.Size = new System.Drawing.Size(176, 31);
        this.stationaryRB.TabIndex = 28;
        this.stationaryRB.TabStop = true;
        this.stationaryRB.Tag = "stationary";
        this.stationaryRB.Text = "Стационарний";
        this.stationaryRB.TextAlign =
System.Drawing.ContentAlignment.TopLeft;
        this.stationaryRB.UseVisualStyleBackColor = true;
        this.stationaryRB.CheckedChanged += new
System.EventHandler(this.stationaryRB_CheckedChanged);
        //
        // singleImpulseRB
        //
        this.singleImpulseRB.AutoSize = true;
        this.singleImpulseRB.Font = new System.Drawing.Font("Times New
Roman", 13.8F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.singleImpulseRB.ForeColor =
System.Drawing.Color.FromArgb(((int)(((byte)(81)))), ((int)(((byte)(86)))),
((int)(((byte)(89)))));
        this.singleImpulseRB.Location = new System.Drawing.Point(29, 109);
        this.singleImpulseRB.Name = "singleImpulseRB";

```

```

this.singleImpulseRB.Size = new System.Drawing.Size(235, 31);
this.singleImpulseRB.TabIndex = 29;
this.singleImpulseRB.Tag = "impulse";
this.singleImpulseRB.Text = "Одиночний імпульс";
this.singleImpulseRB.UseVisualStyleBackColor = true;
this.singleImpulseRB.CheckedChanged += new
System.EventHandler(this.singleImpulseRB_CheckedChanged);
//
// seriesOfImpulsesRB
//
this.seriesOfImpulsesRB.AutoSize = true;
this.seriesOfImpulsesRB.Font = new System.Drawing.Font("Times New
Roman", 13.8F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.seriesOfImpulsesRB.ForeColor =
System.Drawing.Color.FromArgb(((int)(((byte)(81)))), ((int)(((byte)(86)))),
((int)(((byte)(89)))));
this.seriesOfImpulsesRB.Location = new System.Drawing.Point(27, 146);
this.seriesOfImpulsesRB.Name = "seriesOfImpulsesRB";
this.seriesOfImpulsesRB.Size = new System.Drawing.Size(191, 31);
this.seriesOfImpulsesRB.TabIndex = 30;
this.seriesOfImpulsesRB.Tag = "serie";
this.seriesOfImpulsesRB.Text = "Серія імпульсів";
this.seriesOfImpulsesRB.UseVisualStyleBackColor = true;
this.seriesOfImpulsesRB.CheckedChanged += new
System.EventHandler(this.seriesOfImpulsesRB_CheckedChanged);
//
// stopButton
//
this.stopButton.BackColor = System.Drawing.Color.DodgerBlue;

```

```

        this.stopButton.FlatAppearance.BorderColor =
System.Drawing.Color.SeaShell;
        this.stopButton.FlatAppearance.BorderSize = 5;
        this.stopButton.FlatAppearance.MouseDownBackColor =
System.Drawing.Color.Black;
        this.stopButton.FlatAppearance.MouseOverBackColor =
System.Drawing.Color.Black;
        this.stopButton.Font = new System.Drawing.Font("Times New Roman",
13.8F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
(byte)(204));
        this.stopButton.ForeColor = System.Drawing.Color.White;
        this.stopButton.Location = new System.Drawing.Point(205, 653);
        this.stopButton.Name = "stopButton";
        this.stopButton.Size = new System.Drawing.Size(156, 61);
        this.stopButton.TabIndex = 32;
        this.stopButton.Text = "СТОП";
        this.stopButton.UseVisualStyleBackColor = false;
        this.stopButton.Click += new
System.EventHandler(this.stopButton_Click);
//
// label1
//
        this.label1.AutoSize = true;
        this.label1.Font = new System.Drawing.Font("Times New Roman", 13.8F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
(byte)(204));
        this.label1.ForeColor =
System.Drawing.Color.FromArgb(((int)((byte)(81))), ((int)((byte)(86))),
(int)((byte)(89))));
        this.label1.Location = new System.Drawing.Point(323, 247);

```

```

this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(38, 27);
this.label1.TabIndex = 33;
this.label1.Text = "mc";
//
// label5
//
this.label5.AutoSize = true;
this.label5.Font = new System.Drawing.Font("Times New Roman", 13.8F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
(byte)(204));
this.label5.ForeColor =
System.Drawing.Color.FromArgb(((int)(((byte)(81)))), ((int)(((byte)(86)))),
(int)(((byte)(89)))));
this.label5.Location = new System.Drawing.Point(323, 297);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(38, 27);
this.label5.TabIndex = 34;
this.label5.Text = "mc";
//
// label6
//
this.label6.AutoSize = true;
this.label6.Font = new System.Drawing.Font("Times New Roman", 16.2F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
(byte)(204));
this.label6.ForeColor =
System.Drawing.Color.FromArgb(((int)(((byte)(81)))), ((int)(((byte)(86)))),
(int)(((byte)(89)))));
this.label6.Location = new System.Drawing.Point(23, 20);

```

```

this.label6.Margin = new System.Windows.Forms.Padding(10, 0, 10, 0);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(195, 32);
this.label6.TabIndex = 35;
this.label6.Text = "Режим работы";
//
// chart
//
chartArea1.AxisX.ScrollBar.ButtonStyle =
System.Windows.Forms.DataVisualization.Charting.ScrollBarButtonStyles.None;
chartArea1.AxisX.Title = "t, мс";
chartArea1.AxisX.TitleAlignment = System.Drawing.StringAlignment.Far;
chartArea1.AxisY.TextOrientation =
System.Windows.Forms.DataVisualization.Charting.TextOrientation.Horizontal;
chartArea1.AxisY.Title = "U, В";
chartArea1.AxisY.TitleAlignment = System.Drawing.StringAlignment.Far;
chartArea1.BackColor = System.Drawing.Color.White;
chartArea1.Name = "ChartArea1";
chartArea1.Position.Auto = false;
chartArea1.Position.Height = 94F;
chartArea1.Position.Width = 94F;
chartArea1.Position.Y = 3F;
this.chart.ChartAreas.Add(chartArea1);
legend1.Enabled = false;
legend1.Name = "Legend1";
this.chart.Legends.Add(legend1);
this.chart.Location = new System.Drawing.Point(358, 20);
this.chart.Name = "chart";
series1.BorderWidth = 3;
series1.ChartArea = "ChartArea1";

```

```

series1.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
series1.Legend = "Legend1";
series1.MarkerSize = 2;
series1.Name = "Series1";
this.chart.Series.Add(series1);
this.chart.Size = new System.Drawing.Size(1059, 350);
this.chart.TabIndex = 36;
this.chart.Text = "chart";
//
// chartValve
//
chartArea2.AxisX.Title = "t, mc";
chartArea2.AxisX.TitleAlignment = System.Drawing.StringAlignment.Far;
chartArea2.AxisX2.Title = "t, mc";
chartArea2.AxisX2.TitleAlignment =
System.Drawing.StringAlignment.Far;
chartArea2.AxisY.MajorGrid.IntervalOffsetType =
System.Windows.Forms.DataVisualization.Charting.DateTimeIntervalType.Auto;
chartArea2.AxisY.TextOrientation =
System.Windows.Forms.DataVisualization.Charting.TextOrientation.Horizontal;
chartArea2.AxisY.Title = "N";
chartArea2.AxisY.TitleAlignment = System.Drawing.StringAlignment.Far;
chartArea2.BackColor = System.Drawing.Color.White;
chartArea2.Name = "ChartArea1";
this.chartValve.ChartAreas.Add(chartArea2);
this.chartValve.Location = new System.Drawing.Point(420, 376);
this.chartValve.Name = "chartValve";
series2.BorderWidth = 3;
series2.ChartArea = "ChartArea1";

```

```

series2.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
series2.Name = "Series1";
this.chartValve.Series.Add(series2);
this.chartValve.Size = new System.Drawing.Size(966, 338);
this.chartValve.TabIndex = 37;
this.chartValve.Text = "chartValve";
//
// serialData
//
this.serialData.Location = new System.Drawing.Point(29, 462);
this.serialData.Name = "serialData";
this.serialData.Size = new System.Drawing.Size(338, 75);
this.serialData.TabIndex = 38;
this.serialData.Text = "";
this.serialData.TextChanged += new
System.EventHandler(this.serialData_TextChanged);
//
// clearLists
//
this.clearLists.BackColor = System.Drawing.Color.DodgerBlue;
this.clearLists.Font = new System.Drawing.Font("Times New Roman",
13.2F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)204));
this.clearLists.ForeColor = System.Drawing.Color.White;
this.clearLists.Location = new System.Drawing.Point(23, 563);
this.clearLists.Name = "clearLists";
this.clearLists.Size = new System.Drawing.Size(337, 61);
this.clearLists.TabIndex = 40;
this.clearLists.Text = "ОЧИСТИТИ ДАНІ";

```

```

this.clearLists.UseVisualStyleBackColor = false;
this.clearLists.Click += new System.EventHandler(this.clearLists_Click);
//
// label7
//
this.label7.AutoSize = true;
this.label7.Font = new System.Drawing.Font("Times New Roman", 13.2F,
System.Drawing.FontStyle.Regular,          System.Drawing.GraphicsUnit.Point,
(byte)(204));
this.label7.Location = new System.Drawing.Point(25, 376);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(201, 25);
this.label7.TabIndex = 41;
this.label7.Text = "Коефіцієнт фільтру:";
//
// kNUD
//
this.kNUD.DecimalPlaces = 1;
this.kNUD.Font = new System.Drawing.Font("Times New Roman", 13.2F,
System.Drawing.FontStyle.Regular,          System.Drawing.GraphicsUnit.Point,
(byte)(204));
this.kNUD.Increment = new decimal(new int[] {
1,
0,
0,
65536});
this.kNUD.Location = new System.Drawing.Point(29, 405);
this.kNUD.Maximum = new decimal(new int[] {
5,
0,
0,
65536});

```



```

0,
65536});
this.kNUD.Minimum = new decimal(new int[] {
1,
0,
0,
65536});
this.kNUD.Name = "kNUD";
this.kNUD.Size = new System.Drawing.Size(183, 33);
this.kNUD.TabIndex = 42;
this.kNUD.Value = new decimal(new int[] {
1,
0,
0,
65536});
this.kNUD.ValueChanged += new
System.EventHandler(this.kNUD_ValueChanged);
//
// MainForm
//
this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.BackColor = System.Drawing.Color.White;
this.ClientSize = new System.Drawing.Size(1440, 736);
this.Controls.Add(this.kNUD);
this.Controls.Add(this.label7);
this.Controls.Add(this.clearLists);
this.Controls.Add(this.serialData);
this.Controls.Add(this.chartValve);
this.Controls.Add(this.chart);

```

```

this.Controls.Add(this.label6);
this.Controls.Add(this.stationaryRB);
this.Controls.Add(this.seriesOfImpulsesRB);
this.Controls.Add(this.label5);
this.Controls.Add(this.singleImpulseRB);
this.Controls.Add(this.label1);
this.Controls.Add(this.stopButton);
this.Controls.Add(this.startButton);
this.Controls.Add(this.label4);
this.Controls.Add(this.label3);
this.Controls.Add(this.adjustmentDurationOfPauseNUD);
this.Controls.Add(this.adjustmentDurationOfImpulseNUD);
this.Controls.Add(this.label2);
this.ForeColor = System.Drawing.SystemColors.ActiveCaptionText;
this.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedDialog;
this.Name = "MainForm";
this.Padding = new System.Windows.Forms.Padding(20);
this.ShowIcon = false;
this.TransparencyKey = System.Drawing.Color.Black;

((System.ComponentModel.ISupportInitialize)(this.adjustmentDurationOfImpulse
NUD)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.adjustmentDurationOfPauseN
UD)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.chart)).EndInit();
((System.ComponentModel.ISupportInitialize)(this.chartValve)).EndInit();
((System.ComponentModel.ISupportInitialize)(this.kNUD)).EndInit();
this.ResumeLayout(false);

```

```

        this.PerformLayout();

    }

    #endregion

    private System.Windows.Forms.Label label2;
    private System.Windows.Forms.NumericUpDown
adjustmentDurationOfImpulseNUD;
    private System.Windows.Forms.NumericUpDown
adjustmentDurationOfPauseNUD;
    private System.Windows.Forms.Label label3;
    private System.Windows.Forms.Label label4;
    private System.Windows.Forms.Button startButton;
    private System.Windows.Forms.RadioButton stationaryRB;
    private System.Windows.Forms.RadioButton singleImpulseRB;
    private System.Windows.Forms.RadioButton seriesOfImpulsesRB;
    private System.Windows.Forms.Button stopButton;
    private System.Windows.Forms.Label label1;
    private System.Windows.Forms.Label label5;
    private System.Windows.Forms.Label label6;
    private System.Windows.Forms.DataVisualization.Charting.Chart chart;
    private System.Windows.Forms.DataVisualization.Charting.Chart
chartValve;
    private System.Windows.Forms.RichTextBox serialData;
    private System.Windows.Forms.Button clearLists;
    private System.Windows.Forms.Label label7;
    private System.Windows.Forms.NumericUpDown kNUD;
}
}

```

