**Залікове заняття. Кодогенерація та візуалізація кодів**

**Хід роботи:**

1. Для визначеного варіанту завдання реалізувати діаграми USE CASE, Activity, Class, Components у відповідності до обраної мови програмування та середовища розробки.

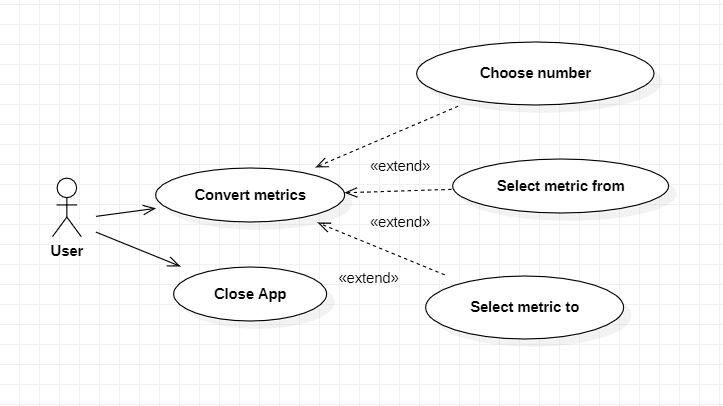


Рисунок 1. UseCase діаграма

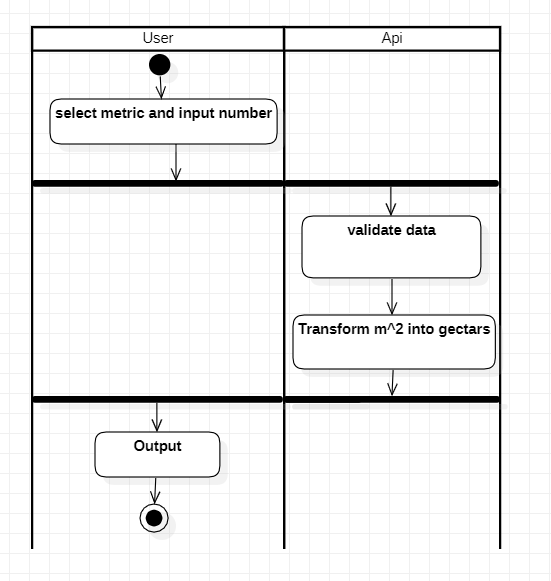


Рисунок 2. Activity діаграма

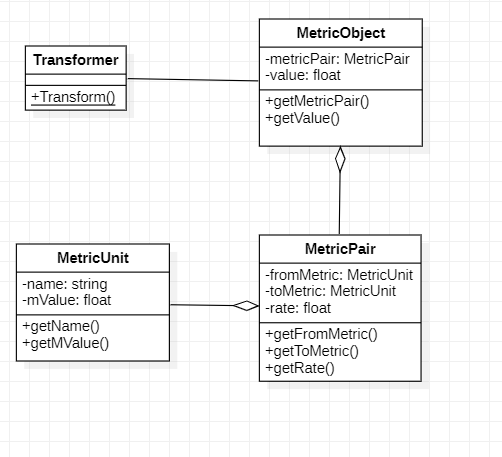


Рисунок 3. Model діаграма

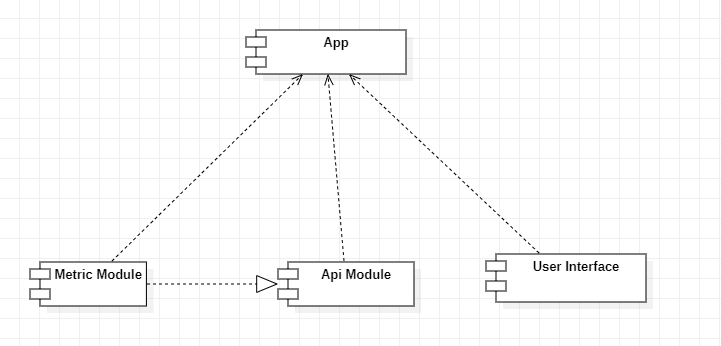


Рисунок 4. Component діаграма

1. Виконати кодогенерацію діаграми класів в середовище розробки та реалізувати програмний продукт.

**Лістинг**

public class MetricUnit {

public MetricUnit(string name, double mValue) {

this.name = name;

this.mValue = mValue;

}

private string name;

private double mValue;

public string getName() {

return name;

}

public double getMValue() {

return mValue;

}

}

public class MetricPair

{

public MetricPair(MetricUnit fromMetricUnit, MetricUnit toMetric)

{

this.fromMetric = fromMetricUnit;

this.toMetric = toMetric;

this.rate = fromMetricUnit.getMValue() / toMetric.getMValue();

}

private MetricUnit fromMetric;

private MetricUnit toMetric;

private double rate;

public MetricUnit getFromMetric()

{

return fromMetric;

}

public MetricUnit getToMetric()

{

return toMetric;

}

public double getRate()

{

return rate;

}

}

public class MetricObject

{

private MetricPair metricPair;

private double value;

public MetricObject(MetricPair metricPair, double value)

{

this.value = value;

this.metricPair = metricPair;

}

public MetricPair getMetricPair()

{

return metricPair;

}

public double getValue()

{

return value;

}

}

public static class Transformer

{

public static double Transform(MetricObject metricObject)

{

return metricObject.getValue() \* metricObject.getMetricPair().getRate();

}

}

public partial class Form1 : Form

{

private MetricUnit[] UNITS =

{

new MetricUnit("квадратні метри", 1),

new MetricUnit("гектар", 10000),

new MetricUnit("акр", 4046.85644643)

};

public Form1()

{

InitializeComponent();

InitializeData();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

if (!double.TryParse(textBoxNumber.Text.Trim().ToString(), out double number))

{

MessageBox.Show("Not double");

return;

}

MetricUnit metricFrom = Array.Find(UNITS, el => el.getName() == comboBoxFrom.SelectedItem.ToString());

MetricUnit metricTo = Array.Find(UNITS, el => el.getName() == comboBoxTo.SelectedItem.ToString());

MetricPair metricPair = new MetricPair(metricFrom, metricTo);

MetricObject metricObject = new MetricObject(metricPair, number);

double result = Transformer.Transform(metricObject);

labelResult.Text = $"Result: {Math.Round(result, 5)}";

}

private void InitializeData()

{

foreach (MetricUnit currentUnit in UNITS)

{

comboBoxFrom.Items.Add(currentUnit.getName());

comboBoxTo.Items.Add(currentUnit.getName());

}

}

}

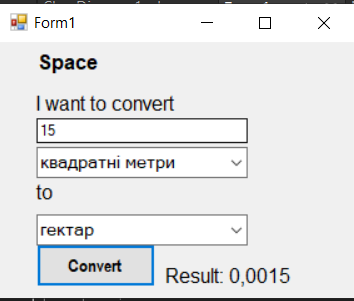


Рисунок 5. Додаток

1. Для готового програмного продукту реалізувати зворотню кодогенерацію

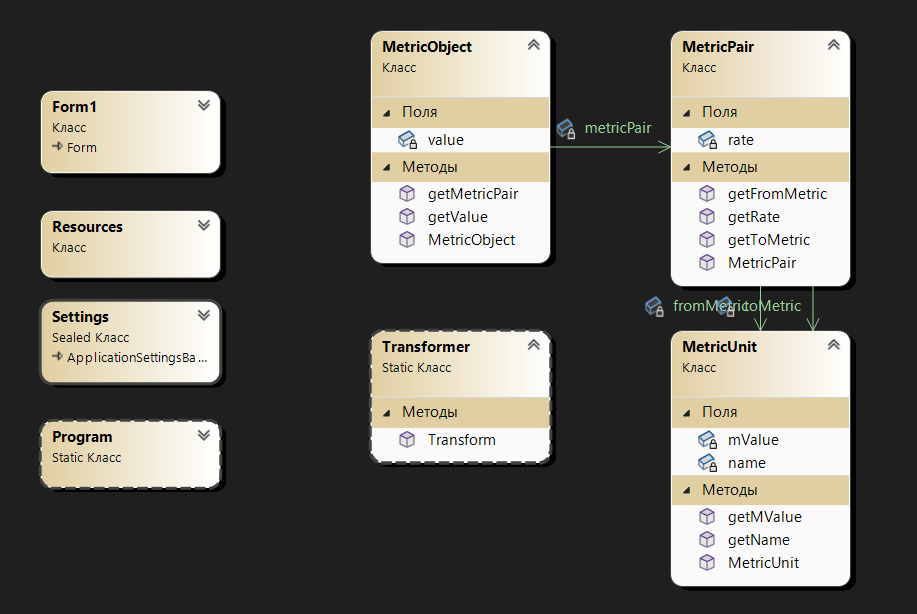


Рисунок 6. Зворотня кодогенерація