

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using CRA.clima.rain;
using CRA.clima.rain.interfaces;
using ChartDirector;

namespace CRA.clima.webapplications.rain
{
    /// <summary>
    /// Descrizione di riepilogo per WebForm1.
    /// </summary>
    public class CRA_clima_rain : System.Web.UI.Page
    {
        .. .. ..
        .. .. ..

        private void Button1_Click(object sender, System.EventArgs e)
        {
            if (RadioButton1.Checked == true) {DoHourly();}
            if (RadioButton2.Checked == true) {DoHalfHourly();}
        }

        private void DoHourly()
        {
            WebChartViewer1.Visible = false;

            //instance of the Rain interface class
            IRain r = new Rain();
            //instance of the RainData class
            RainData rd = new RainData();
            //set parameters
            CRA.clima.rain.HRMeteotest.meteotest_a = 1.5;
            CRA.clima.rain.HRMeteotest.meteotest_b = 2.096;
            CRA.clima.rain.HRMeteotest.meteotest_c = 0.15;
            CRA.clima.rain.HRMeteotest.meteotest_d = -0.09;
            CRA.clima.rain.HRMeteotest.meteotest_e = 0.15;
        }
    }
}

```

```

CRA.clima.rain.HRMeteotest.meteotest_f = 0.7;
CRA.clima.rain.HRMeteotest.meteotest_g = 0.2;
//instance of the HRMeteotest class
IHourlyDailyRain hd = new HRMeteotest();
//set inputs
rd.ModelHR = hd;
rd.rain = double.Parse(TextBox1.Text);
rd.rainMonthlyAverage = double.Parse(TextBox3.Text);
rd.instrumentSensibility = double.Parse(TextBox2.Text);

//call the method
r.RainHourly(ref rd);
//write results on labels
lblPeakAmount.Text = "==";
lblRainSegments.Text = rd.rainHourlyDuration.ToString();

//the data for the line chart
double[] dataY = new double[24];
double[] dataX = new double[24];

int i = 0;
for (i=0; i<24; i++)
{
    dataY[i] = rd.rainHourly[i];
    dataX[i] = i;
}

//Create a XYChart object of size 250 x 250 pixels
XYChart c = new XYChart(500, 340);

//Set the plotarea and add grids
c.setPlotArea(50, 25, 400, 250, -1, -1, 0xc0c0c0, 0xc0c0c0, -1);
//Add titles to the top and bottom of the chart using 7.5pt Arial font.
//The text is white 0xffffffff on a deep blue 0x31319C background.
c.addTitle2(Chart.Top, "hourly values of rain", "arial.ttf", 7.5,
    0xffffffff, 0x31319c);
//Add text in the plot area (top left corner of plotarea)
c.addText(50, 26, "hourly", "arial.ttf", 10, 0x31319C).setAlignment(Chart.TopLeft);
//Use 10 pts Arial as the y axis label font
c.yAxis().setLabelStyle("", 7.5);
c.yAxis().setTitle("rain (mm h-1)", "arial.ttf", 10);
//Use 7.5 pts Arial as the x axis label font
c.xAxis().setLabelStyle("", 7.5);
c.xAxis().setTitle("hour", "arial.ttf", 10);

```

```

        //set labels
        c.xAxis().setLinearScale(0,23,6);
        //Add a red diamonds layer to the chart
        c.addScatterLayer(dataX, dataY,"hourlyRain",
            Chart.DiamondSymbol, 8, 0xc00000);
        //output the chart
        WebChartViewer1.Image = c.makeWebImage(1);
        WebChartViewer1.Visible = true;
    }

private void DoHalfHourly()
{
    WebChartViewer1.Visible = false;

    //instance of the Rain interface class
    IRain r = new Rain();
    //instance of the RainData class
    RainData rd = new RainData();
    //set parameters of the HHRArnoldWilliams class
    CRA.clima.rain.HHRArnoldWilliams.arnoldWilliams_minDailyRain = double.Parse(TextBox2.Text);
    CRA.clima.rain.HHRArnoldWilliams.arnoldWilliams_durationMin = 30;
    CRA.clima.rain.HHRArnoldWilliams.arnoldWilliams_alfa05Mean = 0.3;
    CRA.clima.rain.HHRArnoldWilliams.arnoldWilliams_alfa05Min = 0.03;
    CRA.clima.rain.HHRArnoldWilliams.arnoldWilliams_erlangAlfa = 0.17;
    CRA.clima.rain.HHRArnoldWilliams.arnoldWilliams_erlangBeta = 5;
    //instance of the strategy class
    IHalfHourlyDailyRain h = new HHRArnoldWilliams();
    //set inputs
    rd.rain = double.Parse(TextBox1.Text);
    rd.rainMonthlyAverage = double.Parse(TextBox3.Text);
    rd.ModelHHR = h;
    //call the method
    r.RainHalfHourly(ref rd);
    //write results on labels
    lblPeakAmount.Text = Math.Round(rd.rainHalfHourlyPeakAmount, 2).ToString();
    lblRainSegments.Text = rd.rainHalfHourlyDuration.ToString();

    //the data for the line chart
    double[] dataY = new double[48];
    double[] dataX = new double[48];

    int i = 0;
    for (i=0; i<48; i++)

```

```

    {
        dataY[i] = rd.rainHalfHourly[i];
        dataX[i] = i;
    }

    //Create a XYChart object of size 250 x 250 pixels
    XYChart c = new XYChart(500, 340);
    //Set the plotarea and add grids
    c.setPlotArea(50, 25, 400, 250, -1, -1, 0xc0c0c0, 0xc0c0c0, -1);
    //Add titles to the top and bottom of the chart using 7.5pt Arial font.
    //The text is white 0xffffffff on a deep blue 0x31319C background.
    c.addTitle2(Chart.Top, "half hourly values of rain", "arial.ttf", 7.5,
        0xffffffff, 0x31319c);
    //Add text in the plot area (top left corner of plotarea)
    c.addText(50, 26, "half hourly", "arial.ttf", 10, 0x31319C).setAlignment(Chart.TopLeft);
    //Use 10 pts Arial as the y axis label font
    c.yAxis().setLabelStyle("", 7.5);
    c.yAxis().setTitle("rain (mm h/2-1)", "arial.ttf", 10);
    //Use 7.5 pts Arial as the x axis label font
    c.xAxis().setLabelStyle("", 7.5);
    c.xAxis().setTitle("half hour", "arial.ttf", 10);
    //set labels
    c.xAxis().setLinearScale(0,47,6);
    //Add a red diamonds layer to the chart
    c.addScatterLayer(dataX, dataY,"hourlyRain",
        Chart.DiamondSymbol, 8, 0xc00000);
    //output the chart
    WebChartViewer1.Image = c.makeWebImage(1);
    WebChartViewer1.Visible = true;
}

private void RadioButton1_CheckedChanged(object sender, System.EventArgs e)
{
    RadioButton2.Checked = false;
}

private void RadioButton2_CheckedChanged(object sender, System.EventArgs e)
{
    RadioButton1.Checked = false;
}
}
}

```