

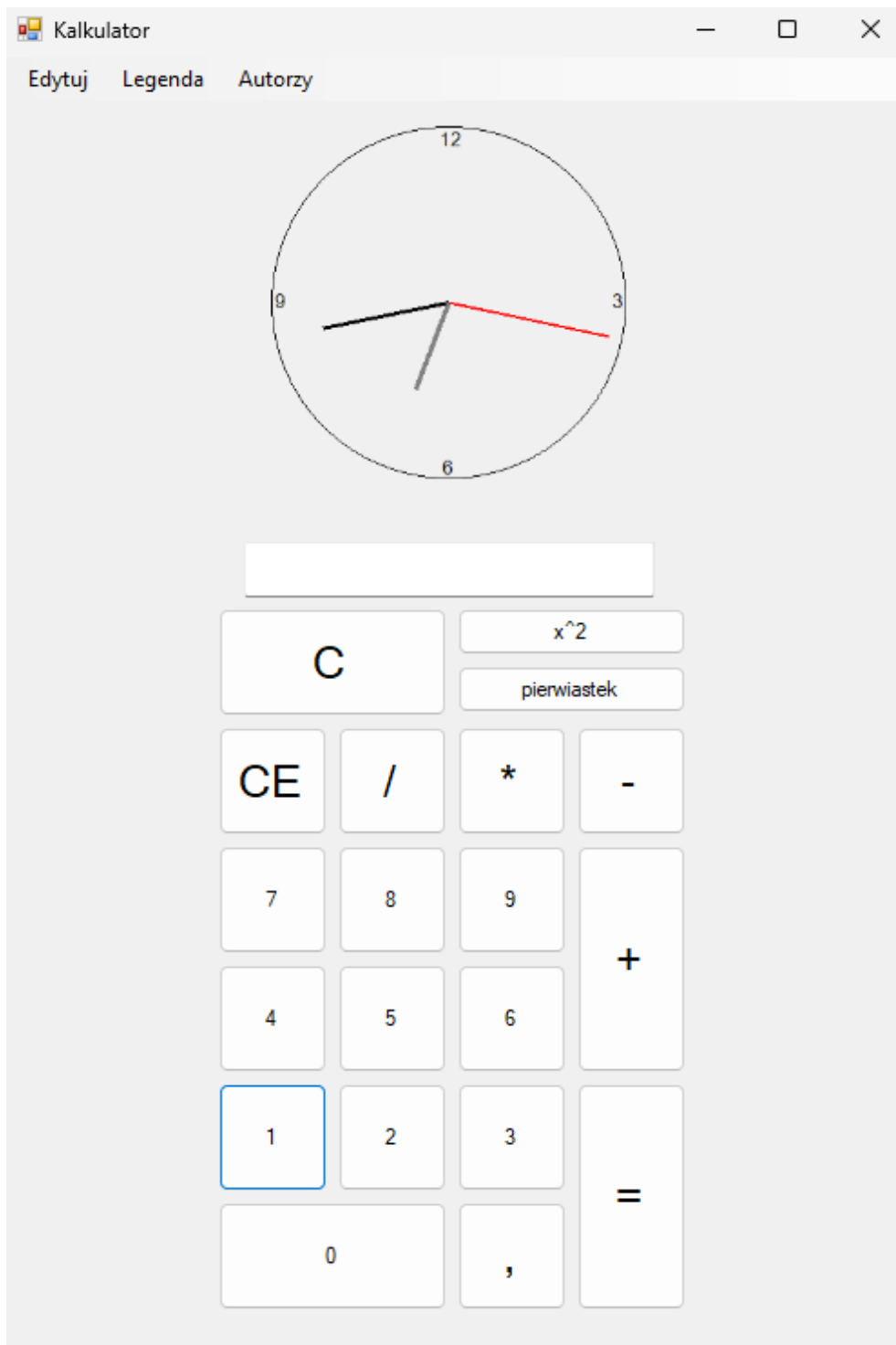
Sprawozdanie zadanie 1

“Kalkulator z zegarem”

Organizacja systemów komputerowych

Adrian Nowogrodzki 184332

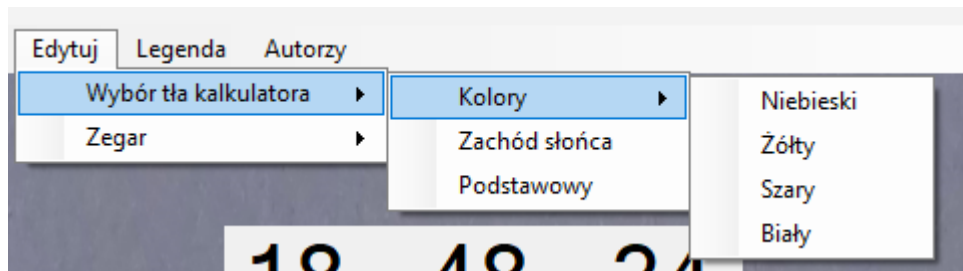
Oskar Nowak 184289



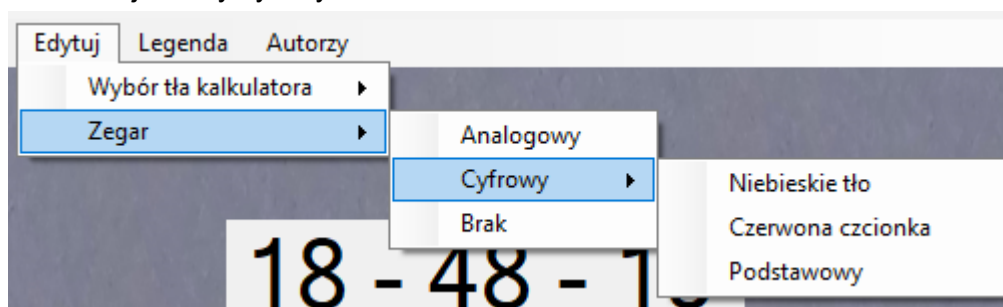
Program umożliwia podstawowe operacje tj. dodawanie, odejmowanie, mnożenie, dzielenie, pierwiastkowanie i podnoszenie do kwadratu. Zasada działania kalkulatora jest identyczna jak w najtańszych kalkulatorach fizycznych.



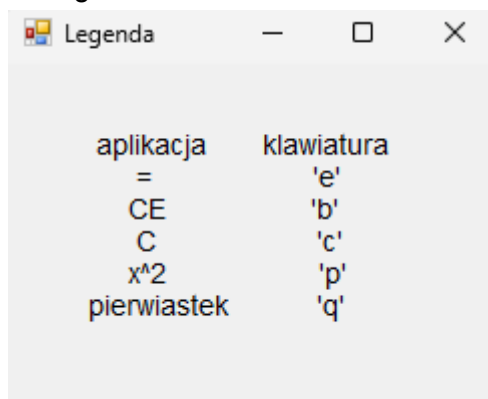
Zegar posiada kilka skórek, jedną z grafiką zachodu słońca oraz kilka motywów kolorystycznych.



Możliwy jest także wybór typu zegara, analogowego albo cyfrowego. Zegar cyfrowy posiada kilka wersji kolorystycznych.



Po wciśnięciu przycisku "Legenda" na górnym pasku otwiera się nowe okno z instrukcją obsługi kalkulatora.



Poniżej przedstawiono fragmentu kodu odpowiadające za wprowadzanie liczb, wprowadzanie operacji oraz operacji wykonywanych po wciśnięciu przycisku "=". Sposób wprowadzania liczb.

```
// akcje po kliknięciu w przyciski
Odwolania: 20
private void button_Click(object sender, EventArgs e)
{
    Button przycisk = (Button)sender;

    //wprowadzanie liczb
    if (przycisk.Text == "0")
    {
        this.liczba_wprowadzana_s = this.liczba_wprowadzana_s + "0";
        this.textBox1.Text = this.liczba_wprowadzana_s;
        this.liczba_wprowadzana_f = (float)Convert.ToDouble(this.liczba_wprowadzana_s);
    }
    if (przycisk.Text == "1")
    {
        this.liczba_wprowadzana_s = this.liczba_wprowadzana_s + "1";
        this.textBox1.Text = this.liczba_wprowadzana_s;
        this.liczba_wprowadzana_f = (float)Convert.ToDouble(this.liczba_wprowadzana_s);
    }
}
```

Wprowadzanie liczby po przecinku.

```
//wprowadzanie liczby po przecinku
if (przycisk.Text == ",")
{
    if (!liczba_wprowadzana_s.Contains(",")) //zabezpieczenie przed wieloma ","
    {
        this.liczba_wprowadzana_s = this.liczba_wprowadzana_s + ",";
        this.textBox1.Text = this.liczba_wprowadzana_s;
    }
}
```

Realizacja czyszczenia danej.

```
//czyszczenie
if (przycisk.Text == "CE")
{
    if (this.liczba_wprowadzana_s.Length > 0)
    {
        this.liczba_wprowadzana_s = this.liczba_wprowadzana_s.Remove(this.liczba_wprowadzana_s.Length - 1, 1);
        if (this.liczba_wprowadzana_s.Length == 0)
        {
            this.liczba_wprowadzana_s = "0";
        }
        else if (this.liczba_wprowadzana_s == "-")
        {
            this.liczba_wprowadzana_s = "0";
        }
        this.liczba_wprowadzana_f = (float)Convert.ToDouble(this.liczba_wprowadzana_s);
    }
    this.textBox1.Text = this.liczba_wprowadzana_s;
}
```

Zapisanie typu operacji oraz warunkowe przeniesienie liczby do akumulatora.

```
//wybór operacji, wynik po wciśnięciu "="  
if (przycisk.Text == "+")  
{  
    if (liczba_akumulowana_f == 0)  
    {  
        this.liczba_akumulowana_f = this.liczba_wprowadzana_f;  
    }  
    this.liczba_wprowadzana_f = 0;  
    this.liczba_wprowadzana_s = "";  
    this.operacja = "+";  
}  
if (przycisk.Text == "-")
```

Wykonanie operacji dodawania po wciśnięciu przycisku "="

```
if (przycisk.Text == "=")  
{  
    if (operacja == "+")  
    {  
        this.liczba_akumulowana_f = this.liczba_akumulowana_f + this.liczba_wprowadzana_f;  
        this.textBox1.Text = this.liczba_akumulowana_f.ToString();  
    }  
    if (operacja == "-")
```

Realizację obsługi klawiatury wykonano za pomocą metody **button.PerformClick()**.

```
//obsługa klawiatury  
Odwołania: 21  
private void button_KeyPress(object sender, KeyPressEventArgs e)  
{  
    if (e.KeyChar == '0')  
    {  
        this.but_nr0.PerformClick();  
    }  
    if (e.KeyChar == '1')  
    {  
        this.but_nr1.PerformClick();  
    }  
    if (e.KeyChar == '2')  
    {  
        this.but_nr2.PerformClick();  
    }  
    if (e.KeyChar == '3')  
    {  
        this.but_nr3.PerformClick();  
    }  
    if (e.KeyChar == '4')  
    {  
        this.but_nr4.PerformClick();  
    }  
}
```

Skórki dla klawiatury zrealizowane są jako zmiana koloru tła, dodanie jako tło obrazka. Zmiana skórki na niebieską.

```
1 odwołanie
private void niebieskiToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.BackgroundImage = null;
    this.BackColor = System.Drawing.Color.Blue;
}
```

Ustawienie motywu zachodu słońca (zdjęcie zachodu jako tło)

```
private void zachódSłońcaToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.BackgroundImage = global::zadanie_1.Properties.Resources._20220812_054610;
    this.BackColor = System.Drawing.SystemColors.Control;
}
```

Zmiana motywu zegara cyfrowego.

```
private void niebieskieTłoToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.zegar_cyfrowy.BackColor = System.Drawing.Color.Blue;
}

1 odwołanie
private void żółtaCzcionkaToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.zegar_cyfrowy.ForeColor = Color.Red;
}
```

Tworzenie i wyświetlanie zegara cyfrowego w takt timera.

```
else if (anal_cyfr == "cyfr")
{
    //pobieranie czasu
    int ss = DateTime.Now.Second;
    int mm = DateTime.Now.Minute;
    int hh = DateTime.Now.Hour;

    //sprawdzanie czy wstawić zero, żeby był format [xx-xx-xx]
    if (ss >= 0 & ss <= 9)
    {
        zero_num_s = "0";
    }
    if (mm >= 0 & mm <= 9)
    {
        zero_num_m = "0";
    }
    if (hh >= 0 & hh <= 9)
    {
        zero_num_h = "0";
    }

    //wyświetlanie zegara
    this.zegar_cyfrowy.Text = (zero_num_h + hh.ToString() + " - " + zero_num_m + mm.ToString() + " - " + zero_num_s + ss.ToString());

    zero_num_s = ""; zero_num_m = ""; zero_num_h = "";
}
```

Tworzenie i wyświetlanie zegara analogowego w takt timera. Na wcześniej utworzonym BMP będzie rysowany zegar. Pobierane są dane czasu z systemu, Dane przetwarzane są na zależności kątowe po czym następuje czyszczenie obszaru, rysowanie tarczy, podpisanie jej oraz na podstawie danych z msCoord() rysowane są wskazówki.

```
private void timer1_Tick(object sender, EventArgs e)
{
    if (anal_cyfr == "anal")
    {
        //tworzenie grafiki
        g = Graphics.FromImage(bmp);

        //pobieranie czasu
        int ss = DateTime.Now.Second;
        int mm = DateTime.Now.Minute;
        int hh = DateTime.Now.Hour;

        //tablica na zmienne x,y położenia końcówki wskazówek
        int[] handCoord = new int[2];

        //odświeżanie
        g.Clear(this.BackColor);

        //rysowanie okręgu
        g.DrawEllipse(new Pen(Color.Black, 1f), 0, 0, WIDTH, HEIGHT);

        //podpisywanie godzin zegara
        g.DrawString("12", new Font("Arial", 12), Brushes.Black, new PointF(140, 2));
        g.DrawString("3", new Font("Arial", 12), Brushes.Black, new PointF(286, 140));
        g.DrawString("6", new Font("Arial", 12), Brushes.Black, new PointF(142, 282));
        g.DrawString("9", new Font("Arial", 12), Brushes.Black, new PointF(0, 140));

        //rysowanie wskaźnika sekund
        handCoord = msCoord(ss, secHAND);
        g.DrawLine(new Pen(Color.Red, 2f), new Point(cx, cy), new Point(handCoord[0], handCoord[1]));

        //rysowanie wskaźnika minut
        handCoord = msCoord(mm, minHAND);
        g.DrawLine(new Pen(Color.Black, 3f), new Point(cx, cy), new Point(handCoord[0], handCoord[1]));

        //rysowanie wskaźnika godzin
        handCoord = hrCoord(hh % 12, mm, hrHAND);
        g.DrawLine(new Pen(Color.Gray, 4f), new Point(cx, cy), new Point(handCoord[0], handCoord[1]));

        //ładowanie BMP do pictureBox'a
        pictureBox1.Image = bmp;
    }
}
```

MsCoord() dla wskazówki godzinowej, oblicza położenie końcówki wskazówki godzinowej na tarczy zegara.

```
private int[] msCoord(int val, int hlen)
{
    int[] coord = new int[2];
    val *= 6; //6 stopni obrót co sekundę/minutę

    if (val >= 0 && val <= 180)
    {
        coord[0] = cx + (int)(hlen * Math.Sin(Math.PI * val / 180));
        coord[1] = cy - (int)(hlen * Math.Cos(Math.PI * val / 180));
    }
    else
    {
        coord[0] = cx - (int)(hlen * Math.Sin(Math.PI * val / 180));
        coord[1] = cy - (int)(hlen * Math.Cos(Math.PI * val / 180));
    }
    return coord;
}
```

Legenda realizowana jest jako utworzenie nowego okienka, który ma w sobie label z instrukcją obsługi

```
private void legendaToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Legenda = new FormLegenda();
    this.Legenda.Show();
}
```

Label z drugiego okna.

```
private void FormLegenda_Load(object sender, EventArgs e)
{
    label1.Text = " aplikacja      klawiatura \n      =      " +
        "      'e' \n      CE      'b' \n      C      " +
        "      'c' \n      x^2      'p' \npierwiastek      'q'";
    label1.Font = new Font("Arial", 10);
}
```