

Konrad Kieda 250997
Dmytro Matviichuk 240545

Termin: pt TN 17:15-20:15

Urządzenia peryferyjne

Temat: Drukarki atramentowe

1. Cel ćwiczenia

Celem ćwiczenia było zapoznanie się z obsługą drukarki atramentowej.

Zadania do wykonania:

1. Dokonać formatowania strony, czcionki i tekstu w oparciu o sekwencje języka PCL, w zakresie zmiany kroju czcionki oraz organizacji strony (plik tekstowy i wysłać do drukarki).
2. Zdefiniować różne kolory z zakresu modeli podstawowych (RGB, CMYK) oraz palet niestandardowych, dla trybu tekstowego.
3. Zdefiniować wybrane tryby graficzne z uwzględnieniem grafiki kolorowej.
4. Napisać i uruchomić program, wykonujący powyższe zadania z poziomu aplikacji graficznej (sterowanie przy pomocy przycisków, list wyboru).
5. Napisać program drukujący, wybraną w porozumieniu z prowadzącym, płaszczyznę (prostą) jednego ze wskazanych modeli składania barw.
6. Napisać program drukujący, kolorową bitmapę w jednym z modeli składania barw.

2. Wstęp

Drukarka atramentowa – drukarka wykorzystująca do druku tusz (barwnikowy lub pigmentowy), poprzez wystrzeliwanie malutkich kropeł środka barwiącego przez mikroskopijne dysze wykonane w głowicy drukarki przechodzącej nad elementem zadrukowywanym. Drukarka atramentowa wykorzystuje do druku tusz, który dzielimy na dwie grupy: rozpuszczalnikowy i pigmentowy. Każdy z podanych tuszy ma dobre i złe strony. Najczęściej w drukarkach atramentowych stosuje się naraz oba typy tuszu. Druk atramentowy dzielony jest na dwie grupy: metoda piezoelektryczna oraz metoda termiczna.

3. Informacja o programie oraz fragmenty kodu

Program został napisany w języku C#, jako drukarkę użyliśmy "Microsoft to PDF".

3.1 Biblioteki które zostały użyte w naszej aplikacji:

```
using System;  
using System.Drawing;  
using System.IO;  
using System.Windows.Forms;  
using System.Drawing.Printing;
```

3.2 Wydruk tekstu w aplikacji

Do wydruku tekstu użyliśmy wbudowanych klas C# - FileStream, StreamWriter oraz File

```
public bool Print()  
{  
    Font printFont;  
    StreamReader streamToPrint;  
    int fontSize = 11;           //standardowy rozmiar czcionki  
    string font = "TimesNewRoman"; //standardowa czcionka  
    string fileName = "File";  
    string path = @"C:\Users\KONJI\Desktop\";  
    streamToPrint = new StreamReader("C:\\Users\\KONJI\\Desktop\\file.txt");  
    ...  
    streamToPrint.Close();  
}
```

3.3 Zmiana stylu drukowanego tekstu

W celu zmiany stylów wydruku należy wybrać odpowiednie opcje przez wybór z listy i zaznaczenie CheckBoxów, comboBoxów.

Czcionka:

```
if(comboBoxFont.SelectedItem != null || comboBoxFont.SelectedItem == "")
{
    if(comboBoxFont.SelectedItem == "Arial")
    {
        font = "Arial";
    }
    else if (comboBoxFont.SelectedItem == "TimesNewRoman")
    {
        font = "TimesNewRoman";
    }
    else if (comboBoxFont.SelectedItem == "Comic Sans MS")
    {
        font = "Comic Sans MS";
    }
}
```

Rozmiar tekstu:

```
if (comboBoxSize.SelectedItem != null || comboBoxSize.SelectedItem == "")
{
    if (comboBoxSize.SelectedItem == "Duzy")
    {
        fontSize = 30;
    }
    else if(comboBoxSize.SelectedItem == "Sredni")
    {
        fontSize = 11;
    }
    else if(comboBoxSize.SelectedItem == "Maly")
    {
        fontSize = 6;
    }
}
```

Pogrubienie, podkreślenie oraz kursywa:

```
if (checkBoxBold.Checked)
{
    printFont = new Font(printFont, printFont.Style ^ FontStyle.Bold);
}
if (checkBoxUnderline.Checked)
{

```

```

        printFont = new Font(printFont, printFont.Style ^ FontStyle.Underline);
    }
    if (checkBoxItalic.Checked)
    {
        printFont = new Font(printFont, printFont.Style ^ FontStyle.Italic);
    }

```

3.4 Wydruk obrazka w aplikacji

Trzecie zadanie miało na celu wydrukowanie grafiki z poziomu aplikacji. Za wybór tej funkcjonalności odpowiada checkboxPrint "Drukuj obrazek". Do wydruku użyliśmy wbudowanego komponentu WindowsForms System.Drawing.Image:

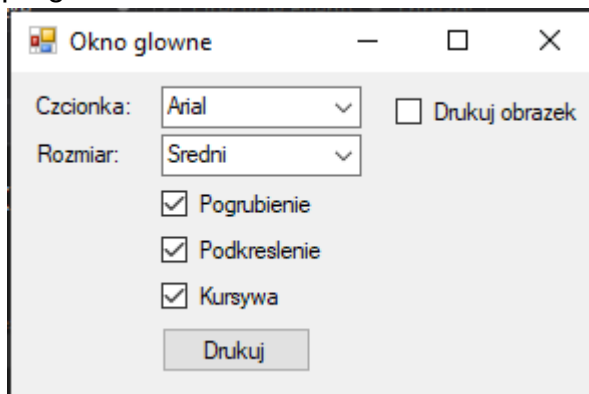
```

private void PrintPage(object sender, PrintPageEventArgs e)
{
    string pathPicture = @"C:\Users\KONJI\Desktop\obrazek.png";
    System.Drawing.Image img = System.Drawing.Image.FromFile(pathPicture);
    //biblioteka do drukowania obrazka
    Point location = new Point(100, 100);
    e.Graphics.DrawImage(img, location);
}

```

4. Działanie aplikacji

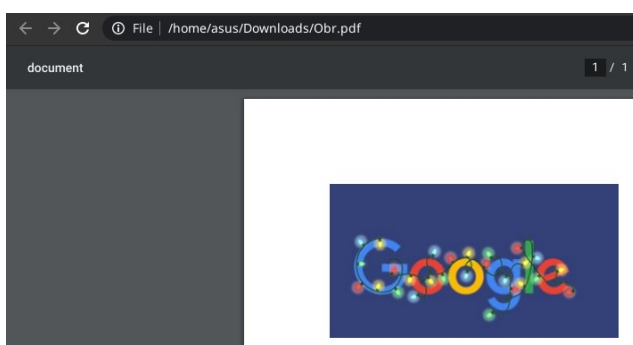
Został stworzony prosty graficzny interfejs użytkownika z dostępem do wszystkich funkcji programu:



4.1 Przykład działania programu:

TEXTtest1234!@#\$%^

4.2 Drukowanie obrazku:



5. Wnioski

Podczas laboratoriów nauczyliśmy się podstawowej obsługi drukarki atramentowej. Aplikacja działa poprawnie, zgodnie ze wszystkimi założeniami.