

# Urządzenia peryferyjne

## Sprawozdanie z laboratorium

Data	Tytuł zajęć	Uczestnicy
15.11.2017 13:00	Drukarki atramentowe	Szymon Abramski (226144) Iwo Bujkiewicz (226203)

### Cel ćwiczenia

Zajęcia miały na celu zapoznanie grupy z niskopoziomowym sposobem sterowania drukarką atramentową HP DeskJet 895Cxi z użyciem kodów PCL.

### Kolejne etapy realizacji

#### Teoria

Do realizacji zadania użyto komputera z systemem operacyjnym Linux Mint 18.2. Linux, oprócz zaawansowanych sterowników i serwerów druku, udostępnia komunikację z drukarkami w taki sam sposób, jak odczyt i zapis do plików na urządzeniu blokowym (np. dysku twardym). Dzięki temu najprostszym sposobem niskopoziomowego sterowania drukarką było wpisywanie kodów PCL do pliku znakowego drukarki.

#### Implementacja

Doskonałym językiem programowania do realizacji tego zadania byłby C, jednak ze względu na konieczność stworzenia graficznego interfejsu użytkownika zdecydowano się skorzystać z prostszej pod tym względem kombinacji Java + JavaFX.

Fundamentem działania aplikacji jest klasa `PrintController`, zawierająca definicje kodów PCL i metody odpowiadające za otwarcie pliku drukarki, wydrukowanie ciągu znaków oraz zresetowanie stanu drukarki.

```

public class PrintController {

    private static final char ESCAPE = 27;

    private static final String RESET = "E";
    private static final String EJECT_PAPER = "&l0H";

    private static final String ITALIC = "(s1S";
    private static final String BOLD = "(s3B";
    private static final String UNDERLINE = "&d0D";

    private static final String WORD_WRAP = "&s0C";
    private static final String ORIENTATION_VERTICAL = "&l00";
    private static final String QUALITY_NORMAL = "*o0M";
    private static final String PAPER_A4 = "&l26A";
    private static final String MARGIN_5 = "&a5C";
    private static final String CARRIAGE_POSITION = "&a3R";

    private final BufferedWriter printWriter;

    public PrintController(String printer) throws IOException {
        printWriter = new BufferedWriter(new FileWriter(printer));
    }

    public void print(String text, boolean bold, boolean italic, boolean underline)
    throws IOException {
        printWriter.write(ESCAPE);
        printWriter.write(PAPER_A4);
        printWriter.write(ESCAPE);
        printWriter.write(ORIENTATION_VERTICAL);
        printWriter.write(ESCAPE);
        printWriter.write(QUALITY_NORMAL);
        printWriter.write(ESCAPE);
        printWriter.write(MARGIN_5);
        printWriter.write(ESCAPE);
        printWriter.write(CARRIAGE_POSITION);
        printWriter.write(ESCAPE);
        printWriter.write(WORD_WRAP);

        if (bold) {
            printWriter.write(ESCAPE);
            printWriter.write(BOLD);
        }
        if (italic) {
            printWriter.write(ESCAPE);
            printWriter.write(ITALIC);
        }
        if (underline) {
            printWriter.write(ESCAPE);
            printWriter.write(UNDERLINE);
        }

        printWriter.write(text);

        //printWriter.write('\f');
        printWriter.write(ESCAPE);
        printWriter.write(EJECT_PAPER);

        printWriter.flush();
    }

    public void reset() throws IOException {
        printWriter.write(ESCAPE);
        printWriter.write(RESET);
        printWriter.flush();
    }

}

```

Punktem startowym aplikacji jest klasa `Main`, której zadaniem jest stworzenie głównego okna aplikacji na podstawie kodu FXML. Klasa `Main` rozszerza klasę `javafx.application.Application`, co umożliwia skorzystanie z interfejsu graficznego JavaFX.

```
public class Main extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource("main_window.fxml"));
        primaryStage.setTitle("HP DeskJet 895Cxi inkjet printer");
        primaryStage.setScene(new Scene(root, 640, 360));
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }

}
```

Kod FXML definiuje rozmieszczenie i wygląd elementów GUI aplikacji, a także klasę instancjowaną jako kontroler widoku.

```
<GridPane fx:controller="outfrost.peripheraldev.Controller"
    xmlns:fx="http://javafx.com/fxml" alignment="center" hgap="10" vgap="10">
    <columnConstraints>
        <ColumnConstraints prefWidth="120" />
        <ColumnConstraints prefWidth="360" />
    </columnConstraints>

    <Label text="Text"
        GridPane.columnIndex="0"
        GridPane.rowIndex="0" />
    <TextField fx:id="textField"
        GridPane.columnIndex="1"
        GridPane.rowIndex="0" />

    <HBox GridPane.columnIndex="1"
        GridPane.rowIndex="1">
        <ToggleButton fx:id="boldToggle"
            text="B"
            style="-fx-font-weight: bold;" />
        <ToggleButton fx:id="italicToggle"
            text="I"
            style="-fx-font-style: oblique;" />
        <ToggleButton fx:id="underlineToggle"
            text="U"
            underline="true" />
    </HBox>

    <Button text="Print"
        onAction="#printButtonAction"
        alignment="BASELINE_RIGHT"
        GridPane.columnIndex="2"
        GridPane.rowIndex="2" />

</GridPane>
```

Klasa `Controller` obsługuje akcje wykonywane przez użytkownika w GUI i używa `PrintController` a w celu wydrukowania wprowadzonego przez użytkownika ciągu znaków, z uwzględnieniem podstawowego formatowania (pogrubienie, pochylenie, podkreślenie).

```
public class Controller {

    @FXML private TextField textField;
    @FXML public ToggleButton boldToggle;
    @FXML public ToggleButton italicToggle;
    @FXML public ToggleButton underlineToggle;

    public void printButtonAction(ActionEvent event) {
        try {
            PrintController printController = new PrintController("/dev/usb/lp0");
            printController.reset();
            printController.print(textField.getText(), boldToggle.isSelected(),
italicToggle.isSelected(), underlineToggle.isSelected());
            printController.reset();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Przy użyciu tak przygotowanej aplikacji z powodzeniem wydrukowano kilka testowych ciągów znaków czcionką normalną, jak również pogrubioną, pochyłą i podkreśloną oraz ich kombinacją. Nie została zaimplementowana zmiana koloru drukowanego tekstu.