

Instrukcja zmiany stawek VAT za pomocą bibliotek JPOS

W celu zmiany zmiany stawek VAT w drukarkach UPOS za pomocą biblioteki JPOS należy wykonać następujące funkcje:

1. Połączyć się z drukarką

```
void open(printerName);  
void claim(int timeout);  
void setDeviceEnabled(true);
```

Parametry:

printerName – urządzenie do którego będziemy się łączyć (np.UPOSFP260FVA
UPOSFP88FVA)

timeout – maksymalny czas zarezerwowania drukarki

2. Sprawdzić czy doba fiskalna jest zamknięta

```
bool getDayOpened();
```

3. Zamknąć dobę fiskalną jeśli jest otwarta

```
void printZReport();
```

4. Ustawić tablicę stawek VAT

```
void setVatValue(int VatID, VatValue);
```

Komenda powinna być wywołana 7 razy (dla każdej stawki VAT osobno).

Parametry:

VatID – liczby od 0 do 6 odpowiadające stawka od A do G

VatValue – wartość stawki VAT w setnych częściach procenta lub o wartościach -1
(stawka nieaktywna), -2 (stawka zwolniona)

5. Wysłać tablicę stawek VAT do drukarki

```
void setVatTable();
```

Przykładowy program ustawiający stawki VAT

```

public static void main(String[] args) throws Exception, IOException
    {
        System.out.println("START");

        Main m=new Main();

        String printerName = new String("UPOSP260FVA"); // UPOSP260FVA UPOSP260FVA
        String ejName = new String("UPOSEJ2"); // UPOSEJ UPOSEJ2
        m.Change_VAT (printerName, ejName);

        System.out.println("END");
    }

public void Change_VAT(String pName, String eName)
    {

        printerName = pName;
        ejName = eName;

        fiscalPrinterCO=new FiscalPrinter();
        ejCO=new ElectronicJournal();

        fiscalPrinterInit = false;
        electronicJournalInit = false;

        try
        {
            {
                printDayReport();
                getVATRatesPL();
                setVATRatesPL();

                uninitEJ();
                uninitPrinter();
                ignoreWarnings();
            }
        }
        catch (JposException e)
        {
            System.out.println("Error: "+e.getErrorCode()+" extended:
"+e.getErrorCodeExtended()+" msg: "+e.getMessage());
        }
        catch (Throwable e)
        {
            e.printStackTrace();
        }
    }

private void initPrinter() throws Throwable
    {
        try
        {
            if (fiscalPrinterInit == true)
            {
                return;
            }
        }
    }

```

```
        uninitEJ();

        fiscalPrinterCO.open(printerName);
        fiscalPrinterCO.claim(2000);

        fiscalPrinterCO.setDeviceEnabled(true);
        fiscalPrinterInit = true;
    }
    catch(JposException e)
    {
        catchException(e, "initPrinter");
    }
}

private void printDayReport() throws Throwable
{
    try
    {
        initPrinter();

        // Z Report
        //
        if(fiscalPrinterCO.getDayOpened() )
        {
            fiscalPrinterCO.printZReport();
        }
    }
    catch (JposException e)
    {
        catchException(e, "printDayReport");
    }
}

private void setVATRatesPL() throws Throwable
{
    try
    {
        initPrinter();

        fiscalPrinterCO.setVatValue(0, "2300"); // A 22.00
        fiscalPrinterCO.setVatValue(1, "700"); // B 7.00
        fiscalPrinterCO.setVatValue(2, "0"); // C 0.00
        fiscalPrinterCO.setVatValue(3, "-1"); // D niezdef
        fiscalPrinterCO.setVatValue(4, "-1"); // E niezdef
        fiscalPrinterCO.setVatValue(5, "-1"); // F niezdef
        fiscalPrinterCO.setVatValue(6, "-2"); // G zwol

        fiscalPrinterCO.setVatTable();
    }
    catch (JposException e)
    {
        catchException(e, "setVATRates");
    }
}
```