

Instruction of VAT rates changes via JPOS library

In order to change VAT rates in UPOS fiscal printers via JPOS, the following functions must be done:

1. Connect with a fiscal printer

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

Parameters:

printerName – device you will connect with (e.g. UPOS FP-T88FVA, FP-T260FVA)
timeout – Maximum time of printer timeout

2. Check if fiscal day is closed

```
bool getDayOpened();
```

3. Close fiscal day if it is open

```
void printZReport();
```

4. Set table of VAT rates

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

Command should be called 7 times (for every VAT rate separately).

Parameters:

VatID – numbers from 0 to 6 adequate for rates from A to G

VatValue –VAT rates values in centesimal percentage parts or with values -1 (inactive rate), -2 (exempted rate)

5. Send table of VAT rates to a printer

```
void setVatTable();
```

Example of a program setting VAT rates

```

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");
    }
}
```