

## Лабораторная работа № 13

### Тема «Формирование графических изображений в документе PDF»

**Цель работы:** изучение возможностей языка PHP и библиотеки **fpdf.php** для формирования графических изображений в документе формата **PDF**.

**Задание.** Взять за основу нижеприведенную программу. Внести в неё изменения с целью вывода в документ PDF линейной диаграммы, отражающей суммы выплаченных сотрудникам учреждения заработных плат по отделам за 2018 год. Исходные данные находятся в базе данных **SotrZarplForeign**.

Фрагмент полученного документа PDF показан на рис. 1

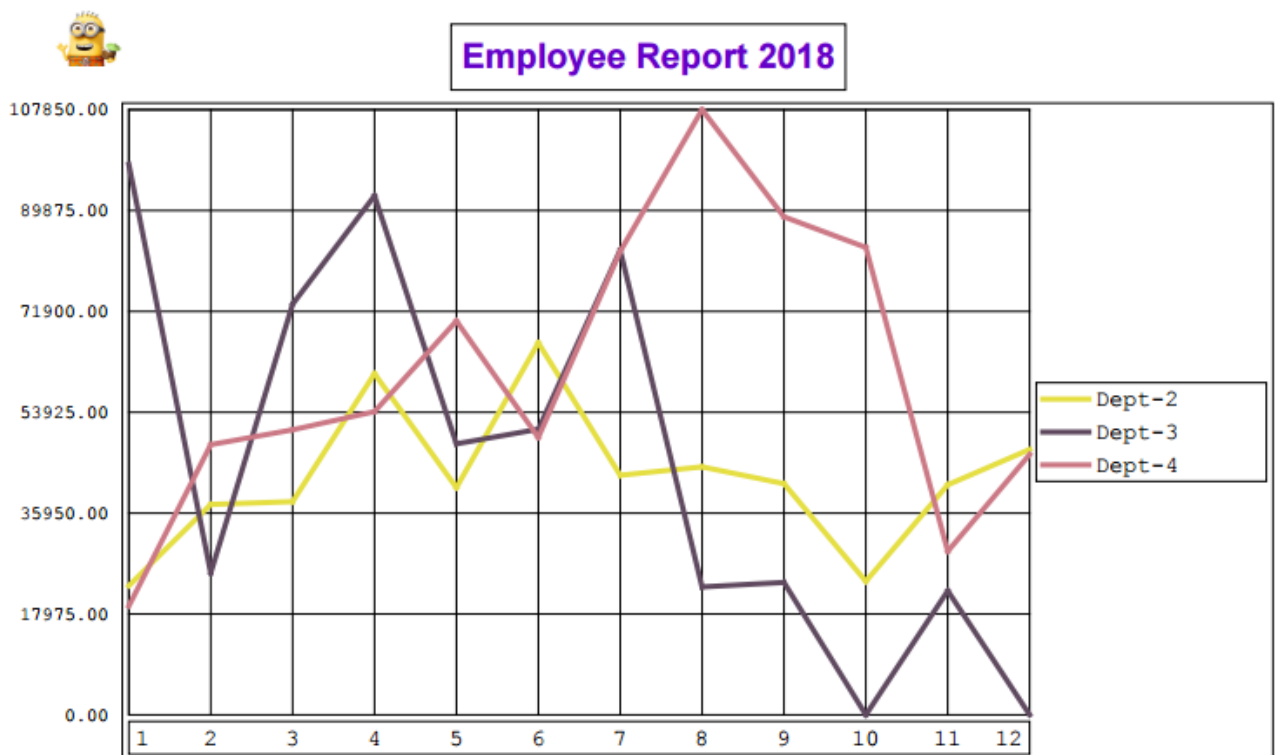


Рис. 1

## Текст исходной программы:

```
<?php

require('fpdf.php');

class PDF_LineGraph extends FPDF {
    function LineGraph($w, $h, $data, $colors=null,
        $maxVal=0, $nbDiv=4){
        /*
        variables:
        $w = the width of the diagram
        $h = the height of the diagram
        $data = the data for the diagram in the
                form of a multidimensional array
        $colors = A multidimensional array containing
                RGB values
        $maxVal = The Maximum Value for the graph vertically
        $nbDiv = The number of vertical Divisions
        */
        $this->SetDrawColor(0,0,0);
        $this->SetLineWidth(0.2);
        $keys = array_keys($data);
        $ordinateWidth = 10;
        $w -= $ordinateWidth+5;
        $valX = $this->getX()+$ordinateWidth;
        $valY = $this->getY();
        $margin = 1;
        $titleH = 8;
        $titleW = $w;
        $lineh = 5;
        $keyH = count($data)*$lineh;
        $keyW = $w/5;
        $graphValH = 5;
        $graphValW = $w-$keyW-3*$margin;
```

```

$graphH = $h-(3*$margin)-$graphValH;
$graphW = $w-(2*$margin)-($keyW+$margin);
$graphX = $valX+$margin;
$graphY = $valY+$margin;
$graphValX = $valX+$margin;
$graphValY = $valY+2*$margin+$graphH;
$keyX = $valX+(2*$margin)+$graphW;
$keyY = $valY+$margin+.5*($h-(2*$margin))-.5*($keyH);
//draw graph frame border
{
    $this->Rect($valX,$valY,$w,$h);
}
//draw graph diagram border
{
    $this->
    Rect($valX+$margin,$valY+$margin,$graphW,$graphH);
}
//draw key legend border
{
    $this->Rect($keyX,$keyY,$keyW,$keyH);
}
//draw graph value box
{
    $this->
    Rect($graphValX,$graphValY,$graphValW,$graphValH);
}
//form an array with all data values from the
//multi-demensional $data array
$ValArray = array();
foreach($data as $key => $value){
    foreach($data[$key] as $val){
        $ValArray[]=$val;
    }
}

```

```

}
//define max value
if($maxVal<ceil(max($ValArray))){
    $maxVal = ceil(max($ValArray));
}
//draw horizontal lines
$vertDivH = $graphH/$nbDiv;
{
    for($i=0;$i<=$nbDiv-1;$i++){
        if($i<$nbDiv){
            $this->
                Line($graphX,$graphY+
                    $i*$vertDivH,$graphX+
                    $graphW,$graphY+$i*$vertDivH);
        } else{
            //$this->
                Line($graphX,$graphY+
                    $graphH,$graphX+$graphW,
                    $graphY+$graphH);
        }
    }
}

//draw vertical lines
$horiDivW = ($graphW/(count($data[$keys[0]])-1));
{
    for($i=0; $i <= (count($data[$keys[0]])-2);$i++){
        if($i<=(count($data[$keys[0]])-2)){
            $this->Line($graphX+$i*$horiDivW,
                $graphY,$graphX+$i*$horiDivW,
                $graphY+$graphH);
        } else {
            //$this->

```

```

        Line($graphX+$graphW,
            $graphY,$graphX+$graphW,
            $graphY+$graphH);
    }
}
}
//draw graph lines
foreach($data as $key => $value){
    $this->
        setDrawColor($colors[$key][0],
            $colors[$key][1],$colors[$key][2]);
    $this->SetLineWidth(0.8);
    $valueKeys = array_keys($value);
    for($i=0;$i<count($value);$i++){
        if($i==count($value)-2){
            $this->Line(
                $graphX+($i*$horiDivW),
                $graphY+$graphH-
                ($value[$valueKeys[$i]]/
                $maxVal*$graphH),
                $graphX+$graphW,
                $graphY+$graphH-
                ($value[$valueKeys[$i+1]]/
                $maxVal*$graphH)
            );
        } else if($i<(count($value)-1)) {
            $this->Line(
                $graphX+($i*$horiDivW),
                $graphY+$graphH-
                ($value[$valueKeys[$i]]/
                $maxVal*$graphH),
                $graphX+($i+1)*$horiDivW,
                $graphY+$graphH-

```

```

        ($value[$valueKeys[$i+1]]/
        $maxVal*$graphH)
    );
}
}

//Set the Key (legend)
$this->SetFont('Courier','',10);
if(!isset($n))$n=0;
$this->Line($keyX+1,$keyY+$lineh/2+$n*$lineh,
        $keyX+8,$keyY+$lineh/2+$n*$lineh);
$this->SetXY($keyX+8,$keyY+$n*$lineh);
$this->Cell($keyW,$lineh,$key,0,1,'L');
$n++;
}

//print the abscissa values
foreach($valueKeys as $key => $value){
    if($key==0)
    {
        $this->SetXY($graphValX,$graphValY);
        $this->Cell(30,$lineh,$value+1,0,0,'L');
    }
    else if($key==count($valueKeys)-1){
        $this->SetXY($graphValX+$graphValW-
                30,$graphValY);
        $this->Cell(30,$lineh,$value+1,0,0,'R');
    }
    else {
        $this->
            SetXY($graphValX+$key*$horiDivW-15,
                    $graphValY);
        $this->Cell(30,$lineh,$value+1,0,0,'C');
    }
}

```

```

    }
    //print the ordinate values
    $this->SetFont('Courier','',8);
    for($i=0;$i<=$nbDiv;$i++){
        $this->
            SetXY($graphValX-10,
                $graphY+($nbDiv-$i)*$vertDivH-3);
        $this->Cell(8,6,sprintf('%.2f',
            $maxVal/$nbDiv*$i),0,0,'R');
    }
    $this->SetDrawColor(0,0,0);
    $this->SetLineWidth(0.2);
}

// Page header
function Header()
{
    // Text color
    $this->SetTextColor(100,0,200);
    // Logo
    $this->Image('2.jpg',10,8,10);
    // Arial bold 15
    $this->SetFont('Arial','B',15);
    // Move to the right
    $this->Cell(60);
    // Title
    $this->Cell(60,10,'Employee Report 2018',1,0,'C');
    // Line break
    $this->Ln(12);
}

// Page footer
function Footer()

```

```

{
    // Text color
    $this->SetTextColor(100,0,200);
    // Position at 1.5 cm from bottom
    $this->SetY(-15);
    // Arial italic 8
    $this->SetFont('Arial','I',8);
    // Page number
    $this->Cell(0,10,'Page '.$this->PageNo(),0,0,'C');
}

}

//=====
//Можно написать функция LoadData(), которая заполнит
// данными массив $maindata:
//=====
// Функция Load data
//=====
function LoadData()
{
    // . . .
    return $maindata;
}

$pdf = new PDF_LineGraph();
$pdf->SetFont('Arial','',10);

//Пример структуры данных в массиве $data:
$data = array(
    'Group 1' => array(
        '08-02' => 2.7,
        '08-23' => 3.0,

```



```

        '09-13' => 3.3928571,
        '10-04' => 3.2903226,
        '10-25' => 3.1
    ),
    'Group 2' => array(
        '08-02' => 2.5,
        '08-23' => 2.0,
        '09-13' => 3.1785714,
        '10-04' => 2.9677419,
        '10-25' => 3.33333
    )
);

//Здесь сформировать массив цветов $colors
//для рисования линий случайным образом

//$colors = array();
// . . .
$colors = array(
    'Group 1' => array(114,171,237),
    'Group 2' => array(163,36,153)
);

$pdf->AddPage();
$pdf->LineGraph(190,100,$data,$colors,6,6);
$pdf->Output();

?>

```