

úsilia musíme ešte aplikovať. Ostatné súbory s názvom arial môžeme pokojne vymazať, pretože ich nebudeme viac potrebovať. Font použijeme takto:

```
define('FPDF_FONTPATH','font/');
require('fpdf.php');
$pdf=new FPDF("P","pt","A4");
$pdf->Open();
$pdf->AddFont('arial','',arial.php);
$pdf->AddPage();
$pdf->SetFont('arial','',18);
$pdf->Cell(100,20,"test text",0,1,"C");
$pdf->Output();
```

Hneď v prvom riadku si nastavíme cestu k definícii fonu. V adresári font budú umiestnené súbory arial.php a arial.z. Pomocou metódy addFont zaregistrujeme nový font.

```
$pdf->AddFont('arial','',arial.php);
```

Všeobecné upozornenie vyplývajúce zo skúseností znie, že pre každý štýl písma musí byť vytvorený vlastný definíčný súbor a samostatný archív s dátami. Pozor na veľkosť výsledného súboru PDF, ktorá bude rásť s rastúcim počtom písom. Obsirnejšie riešenie problematiky hlavičky a päty dokumentu. Tieto prvky možno vkladať do dokumentu ručne, ale oveľa elegantnejší spôsob je použiť funkcie Header a Footer. Štandardne tieto funkcie neobsahujú nijaký kód, a preto ich môžeme spokojne predefinovať. Ďalej uvedený kód hneď v prvých riadkoch vytvára novú triedu FPDF. Spomínané funkcie sa zmenia na nami definované.

```
define('FPDF_FONTPATH','font/');
require('fpdf.php');

class PDF extends FPDF
{
function Header()
{
    $this->SetY(5);
    $this->SetFont('times','',14);
    $this->SetTextColor(255,0,0);
    $this->Cell(0,10,"Hlava pdf dokumentu",0,1,"C");
}
}
```

```
function Footer()
{
    $this->SetY(-15);
    $this->SetFont('times','',8);
    $this->SetTextColor(128);
    $this->Cell(0,10,"Päta pdf dokumentu ".$this->PageNo(),0,0,"C");
}
}
$pdf=new PDF();
$pdf->Open();
$pdf->AddPage();
$pdf->AddFont('arialce','',arial.php);
$pdf->SetFont('arialce','',10);
$pdf->SetTextColor(0,0,0);
$pdf->MultiCell(0,5,$text);
$pdf->Output();
```

Premenná \$text reprezentuje telo dokumentu. Na nastavenie autora dokumentu použijete metódu SetAuthor.

```
$pdf->SetAuthor("meno autora");
```

Nastavenie tvorca dokumentu:

```
$pdf->SetCreator("tvorca dokumentu");
```

Nastavenie titulku dokumentu:

```
$pdf->SetTitle("Titulok");
```

Záver

Platným účtovným dokladom je aj faktúra v PDF, obsahujúca všetky náležitosti spolu s pečiatkou a podpisom. Systém elektronických faktúr využíva viacero firmi a je overený právnikom, ako aj daňovým úradom, takže nie je problém vytvoriť systém, ktorý po objednaní zašle automaticky vygenerovanú predfaktúru a po zaplatení faktúry. Takže sme spoznali ďalšiu neznámu z programovania v PHP. Na to, aby ste mohli programovať inteligentné a veľmi efektívne skripty, potrebujete zdolať ešte zopár úskalí. Nabudúce budeme programovať WAP. Takže aj preto sa teším na ďalšie stretnutie s vami.

Jozef KOZÁK ml.

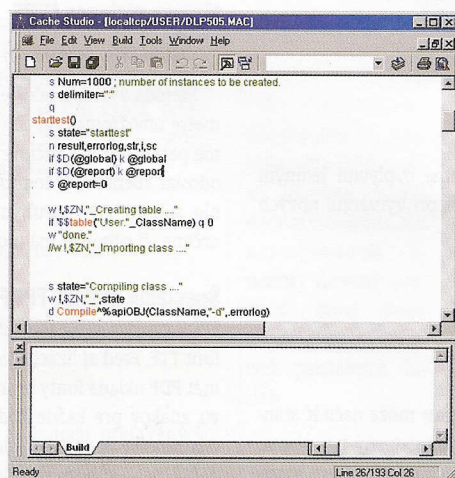
Caché Studio

Spoločnosť InterSystems je známa predovšetkým ako dodávateľ databázovej platformy Caché. Drvivá väčšina možností, ktoré Caché ponúka, je prístupná aj zo základného prístupového a vývojového nástroja – Caché Studia.

Databázová platforma Caché predstavuje veľmi vyspelé prostredie na tvorbu komplexných aplikácií založených na práci s perzistentnými (trvalo dostupnými) dátami. Na rozdiel od klasických relačných či čisto objektových platforiem je Caché vďaka svojej architektúre uloženia dát schopná pracovať s dátami z objektového aj relačného pohľadu. Vývojárom tento duálny prístup nijako nebráni pri využívaní ich doterajších znalostí, navyše môžu postupne prechádzať od pôvodných spôsobov vývoja aplikácií k novším.

Pre Caché je veľmi príznačné, že na rozdiel od objektovo-relačných platforiem nedochádza k strate výkonu pri prístupe k dátam pomocou objektivej nastavby, pretože Caché nič také neobsahuje. Ponúka iba dve rovnako výkonné projekcie dát, teda dve projekcie vo svojom základe, relačnú a objektovú. Za rovnakých podmienok môžu existovať ďalšie projekcie, napríklad so technológiou XML.

Základom modelovania komponentov aplikácií sú objekty. Objekty sú organizované do tried, v



Caché Studio

ktorých sú opísané vlastnosti (dáta) a metódy (správanie) objektov. Definície tried sú uložené v jednotnom úložisku, nazývanom Caché Class Dictionary, čo môžeme voľne preložiť ako slovník

tried. Tento slovník je sám osebe databáza a ostatné objekty k nemu môžu pristupovať objektovo. Fyzické úložisko dát sa nazýva globál a ide o viacrozmerné riedke pole.

V momente prekladu definície triedy dochádza k vytvoreniu dvoch rôznych, navzájom synchronizovaných zostáv kódu, ktoré zaisťujú optimálny prístup k inštanciam objektov triedy buď pomocou objektového, alebo pomocou relačného prístupu.

Objektový model Caché okrem iného zahŕňa:

- **Triedy** – základné kamene objektového programovania, Caché umožňuje definovať triedy uchovávajúce aplikačné dáta a vykonávajúce aplikačnú logiku

- **Vlastnosti** – opisujú dáta ukladané a spracúvané triedami

- **Metódy** – opisujú správanie objektov a vzájomnú interakciu

- **Relácie** – opisujú vzťahy medzi jednotlivými triedami v aplikácii

- **Dedičnosť** – triedy definované v Caché môžu byť odvodené od iných tried a preberať ich vlastnosti a metódy

- **Polymorfizmus** – triedy môžu modifikovať zdedené vlastnosti a metódy

Nabudúce sa pozrieme napríklad na to, ako definovať triedu pomocou jazyka *Class Definition Language*.

Marek Kocan