

APLIKACJE INTERNETOWE

PHP



TEMAT 5-07: Podstawy generowania grafiki.

Autor dokumentu: Wojciech Galiński

środa, 7 września 2016 r.

351203 Technik informatyk

ŹRÓDŁA WIEDZY:

<http://www.php.net/manual/pl>, <http://webmaster.helion.pl/index.php/kurs-php>, <http://pl.wikipedia.org/>,
<http://pl.wikibooks.org/wiki/PHP>, <http://phpkurs.pl/>, <http://kursphp.com/>.

Zagadnienia obowiązkowe

- Generowanie obrazków w PHP** – możliwe jest dzięki bibliotece „GD”. Biblioteka ta umożliwia wczytywanie i generowanie obrazów oraz manipulowanie obrazami w czasie rzeczywistym.
 - Formaty plików graficznych zalecane do generowania grafiki w PHP** – wyróżniamy:
 - ➔ **JPEG** – stratny format kompresji obrazów przeznaczony przede wszystkim dla obrazów o bardzo zróżnicowanej kolorystyce (np. zdjęcia);
 - ➔ **GIF** – bezstratny format kompresji obrazów przeznaczony dla obrazów zawierających duże jednobarwne obszary (np. flagi, przyciski) oraz 2 stopniową przeźroczystość. Format ten obsługuje 256 kolorów dla każdego bloku obrazu, więc dzieląc obraz na wiele bloków, można ominąć to ograniczenie.
 - ➔ **PNG** – bezstratny format kompresji obrazów przeznaczony dla obrazów zawierających duże jednobarwne obszary (np. flagi, przyciski) oraz 256-stopniową przeźroczystość. W pełni obsługuje 24-bitową paletę kolorów.Format GIF nie jest domyślnie obsługiwany.
 - Formaty obrazów dostępne za pomocą biblioteki „GD”** – w rozwiązaniu tego zadania pomagają poniższy skrypt:

```
print_r(gd_info()); echo '<br />';
echo 'Obsługa JPEG? <b>'.(imagetypes() & IMG_JPEG? 'TAK': 'NIE').'</b>.';
echo 'Obsługa GIF? <b>'.(imagetypes() & IMG_GIF? 'TAK': 'NIE').'</b>.';
echo 'Obsługa PNG? <b>'.(imagetypes() & IMG_PNG? 'TAK': 'NIE').'</b>.';
```
 - Generowanie prostego obrazu** – oto przykład tekstu będącego obrazem, np.

```
$obraz = imagecreatetruecolor(500, 300);
imagefill($obraz, 0, 0, 0xFF0000); // wypełnienie obrazu czerwonym tłem
imagefilledrectangle($obraz, 0, 0, 499, 149, 0xFFFFFFFF); // biały prostokąt
header('Content-type: image/png'); imagepng($obraz); // utworzenie obrazu
imagedestroy($obraz); // zwolnienie pamięci
```
 - Wstawianie wygenerowanego obrazka w dokumencie HTML** – w tym celu wystarczy wywołać skrypt tam, gdzie podaje się nazwę pliku graficznego wewnątrz znacznika „img”, np.

 - Zapisanie wygenerowanej grafiki do pliku** – stosujemy w tym celu drugi parametr, np.

```
$obraz = imagecreatetruecolor(500, 300);
imagefill($obraz, 0, 0, 0xFF0000); // wypełnienie obrazu czerwonym tłem
imagefilledrectangle($obraz, 0, 0, 499, 149, 0xFFFFFFFF); // biały prostokąt
$nazwa_pliku = 'plik.png';
imagepng($obraz, $nazwa_pliku); // utworzenie pliku obrazu
list($szerokosc, $wysokosc) = getimagesize($nazwa_pliku);
echo '<p>Rozmiar obrazu: '. imagesx($obraz). 'x'. imagesy($obraz). ' ('.
    $szerokosc. 'x'. $wysokosc. ')</p>';
imagedestroy($obraz); // zwolnienie pamięci
```
 - Odczytywanie koloru wybranego piksela z obrazu** – oto przykład:

```
$obraz = imagecreate(10, 10);
imagefilledrectangle($obraz, 0, 0, 5, 5, 0xFFFFFFFF); // biały prostokąt
echo 'Kolor piksela w lewym górnym rogu: '. imagecolorat($obraz, 0, 0);
```
 - Zmiana grubości linii** – stosujemy do tego funkcję „imagesetthickness”.
- PRZYKŁAD:**
- ```
$obraz = imagecreatetruecolor(200, 100);
$biały = imagecolorallocate($obraz, 0xFF, 0xFF, 0xFF);
$czarny = imagecolorallocate($obraz, 0x00, 0x00, 0x00);
```

```

imagefilledrectangle($obraz, 0, 0, 299, 99, $bialy);
imagesetthickness($obraz, 5); // ustawienie grubości
imagerectangle($obraz, 10, 10, 190, 90, $czarny);
imageline($obraz, 50, 20, 150, 30, $czarny);
imagedashedline($obraz, 50, 80, 150, 80, $czarny);
for ($x=50; $x<=150; $x+=2)
 for ($y=40; $y<=60; $y+=2) imagesetpixel($obraz, $x, $y, $czarny);
imageinterlace($obraz, true); // włączenie przeplotu
header('Content-Type: image/png'); imagepng($obraz); imagedestroy($obraz);

```

W powyższym skrypcie wygenerowano obrazek z włączonym przeplotem (więcej o przeplocie znajduje się m. in. pod adresem: [http://en.wikipedia.org/wiki/Interlacing\\_%28bitmaps%29](http://en.wikipedia.org/wiki/Interlacing_%28bitmaps%29)).  
Inne funkcje modyfikujące sposób rysowania to m. in.: [imagesetbrush](#), [imagesetstyle](#).

9. **Dynamiczne rysowanie figur geometrycznych i opisów tekstowych** – za pomocą biblioteki „GD” można rysować m. in. punkty, odcinki, okręgi, koła i elipsy; prostokąty i kwadraty; wielokąty.

PRZYKŁAD 1:

```

$obraz = imagecreatetruecolor(100, 100);
imagesetthickness($obraz, 3);
imagepolygon($obraz, array(50, 30, 30, 70, 70, 70), 3, 0x00FF00);
imagefilltoborder($obraz, 50, 50, 0x00FF00, 0x0000FF);
header('Content-type: image/png'); imagepng($obraz); imagedestroy($obraz);

```

PRZYKŁAD 2:

```

// Utworzenie obszaru obrazka
$szeroosc = 600; $wysokosc = 300;
$obraz = imagecreate($szeroosc, $wysokosc);
// Rejestracja kolorów używanych do obrazka
$bialy = imagecolorallocate($obraz, 255, 255, 255);
$niebieski = imagecolorallocate($obraz, 0, 0, 255);
imagefill($obraz, 0, 0, $niebieski); // wypełnienie tła obrazka
// Rysowanie okręgu i jego opisu
imagearc($obraz, $szeroosc/2, $wysokosc/2, 200, 200, 0, 360, $bialy);
imagestringup($obraz, 5, 180, 250, 'y^2 = (x-a)^2 + (y-b)^2', $bialy);
// Rysowanie prostej i jej opisu
imageline($obraz, 0, $wysokosc/2, $szeroosc, $wysokosc/2, $bialy);
imageline($obraz, 0, $wysokosc/2, $szeroosc, $wysokosc/2, $bialy);
imageline($obraz, $szeroosc/2, 0, $szeroosc/2, $wysokosc, $bialy);
imageline($obraz, 0, $wysokosc, $szeroosc, 0, $bialy);
imagestring($obraz, 4, 500, 60, 'y = 1/2x', $bialy);
// Wyświetlenie obrazka – pokazano 3 podstawowe formaty obsługiwane w "GD"
header('Content-type: image/png'); imagepng($obraz);
// header('Content-type: image/gif'); imagegif($obraz);
// header('Content-type: image/jpeg'); imagejpeg($obraz);
imagedestroy($obraz); // zwolnienie pamięci

```

Inne funkcje rysujące figury geometryczne to: [imagearc](#), [imagefilledarc](#) – rysuje (wypełnione) okręgi lub elipsy; [imagepolygon](#), [imagefilledpolygon](#) – rysuje (wypełnione) wielokąty.

10. **Opis wszystkich funkcji biblioteki „GD” w Internecie** – można go znaleźć pod adresem: <http://php.net/manual/pl/ref.image.php>.

## Zadania

1. Zapoznaj się z przedstawionymi przykładami. Wykonaj je i przeanalizuj ich działanie.
2. Wykonaj wykres funkcji liniowej, kwadratowej oraz wykresy funkcji trygonometrycznych.
3. Przedstaw graficznie (na dwóch kołach symbolizujących zbiory) następujące operatory logiczne: NOT A, A AND B, A XOR B, A OR B, A / B (A – B).
4. Korzystając z poznanych funkcji, narysuj wymyślony przez siebie obrazek.