

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKEJ A POTRAVINÁRSKEJ TECHNOLOGIE
Ústav informatizácie, automatizácie a matematiky



Dynamické generovanie obrázkov vo formáte SVG

Bakalárska práca

Vypracoval: Pavel Bodnár

Školiteľ: Ing. Ľuboš Čírka, PhD.

Bratislava 2009



ZADANIE BAKALÁRSKEJ PRÁCE

Študent: **Pavel Bodnár**
ID študenta: **25522**
Študijný program: **automatizácia, informatizácia a manažment v chémii a potravinárstve**
Kombinácia študijných odborov: **5.2.14 automatizácia, 5.2.52 priemyselné inžinierstvo**
Vedúci práce: **Ing. Ľuboš Čirka, PhD.**

Názov práce: **Dynamické generovanie obrázkov vo formáte SVG**

Špecifikácia zadania:

Cieľom práce je vytvoriť modul s názvom "pečiatka". Jedná sa o webovú aplikáciu založenú na jazykoch HTML(SVG), PHP a SQL (MySQL).

Opis modulu:

- Zobrazenie základných typov pečiatok uložených v databáze
- Pre zvolený typ pečiatky vytvorenie objednávky
- Objednávka umožňuje vyber jednotlivých častí pečiatky
- Užívateľ si môže cez webové rozhranie navrhnuť otlačok pečiatky vo vektorovom tvare

Rozsah práce: **30**

Riešenie zadania práce od: **18. 02. 2009**

Dátum odovzdania práce: **22. 05. 2009**

Pavel Bodnár
študent

prof. Dr. Ing. Miroslav Fikar
vedúci pracoviska



prof. Dr. Ing. Miroslav Fikar
garant študijného programu

Touto cestou by som sa chcel poďakovať vedúcemu bakalárskej práce, Ing. Ľubošovi Čirkovi, PhD za odborné vedenie, cenné rady i pripomienky, ktoré mi poskytol pri spracovaní bakalárskej práce.

Abstrakt

Cieľom bakalárskeho projektu je oboznámenie sa s jazykom SVG určeným na opísanie dvojrozmernej vektorovej grafiky a jeho praktickou aplikáciou do webovej stránky.

Teoretická časť sa zaoberá štruktúrou jazyka SVG a jeho základnými funkciami. Obsah praktickej časti spočíva vo vytvorení základných vektorových tvarov, ktoré sa použijú na zobrazenie pečiatok. Tieto vektorové tvary sa vo forme SVG kódu prepoja s PHP formulárom.

Po vytvorení sa modul zakomponuje do webovej stránky firmy zaoberajúcej sa výrobou pečiatok.

Abstract

The aim of bachelor project is an acquaintance with SVG language which is assigned to describe two-dimensional vector graphics and its practical application to a web site.

The theoretical part deals with a structure of SVG language and its basic functions. The contents of practical part consist in the creation of basic vector shapes that will be used to display rubber stamps. These vector shapes in the form of SVG code will be connected with PHP form.

After creation this modul will be inserted in the website of a company which deals with rubber stamps production.

Obsah

Úvod	9
1 Teoretická časť	10
1.1 SVG.....	10
1.2 Štruktúra SVG.....	11
1.2.1 Základný element – svg.....	11
1.2.2 Element – g.....	12
1.2.3 Element – defs	12
1.2.4 Element – symbol, use	13
1.2.5 Element – image.....	14
1.3 Základné vektorové tvary	15
1.3.1 Obdĺžnik (Rectangle)	15
1.3.2 Kruh (Circle)	15
1.3.3 Elipsa (Ellipse)	15
1.3.4 Čiara (Line)	16
1.3.5 Zalomená čiara (Polyline)	16
1.3.6 Mnohouholník (Polygon)	16
1.3.7 Cesta (Path)	17
1.3.8 Bézierova krivka (Bézier curve)	17
1.4 Text	18
1.4.1 Element – text.....	18
1.4.2 Element – tspan	19
1.4.3 Element – textPath	20
2 Praktická časť	21
2.1 Vytvorenie formulárov.....	21
2.1.1 Formulár pre obdĺžnikovú pečiatku	21
2.1.2 Formulár pre kruhovú pečiatku	22
2.2 Vytvorenie obdĺžnikovej pečiatky	23
2.2.1 Podklad pečiatky	23
2.2.2 Cesty.....	23
2.2.3 Text.....	25
2.2.4 Výsledná ukážka	26

2.3 Vytvorenie kruhovej pečiatky	28
2.3.1 Podklad pečiatky	28
2.3.2 Cesty	28
2.3.3 Text	29
2.3.4 Vloženie obrázka	29
2.3.5 Výsledná ukážka	31
3 Záver	33
Zoznam použitej literatúry	34
Prílohy	35

Zoznam obrázkov

- Obr. 1 Tri kruhy, ktoré zdedili vlastnosti (farbu a hrúbku okraju) elementu `<g>`
- Obr. 2 Kruh s lineárnym farebným prechodom, ktorý bol zadefinovaný vnútri elementu `<defs>` a následne sa tento prechod zavolať v elemente `<circle>`.
- Obr. 3 Dva oranžové kruhy sú zadefinované pomocou elementu `<symbol>` a pomocou elementu `<use>` sú umiestnené medzi červený a žltý kruh
- Obr. 4 Dva oranžové kruhy sú zadefinované pomocou elementu `<symbol>` a pomocou elementu `<use>` sú umiestnené nad červeným a žltým kruhom
- Obr. 5 Ukážka práce s atribútmi `dx`, `dy` a `rotate`
- Obr. 6 Umiestnenie textu na krivke
- Obr. 7 Ukážka formulára v prehliadači Mozilla Firefox
- Obr. 8 Ukážka vygenerovanej obdĺžnikovej pečiatky
- Obr. 9 Ukážka formulára v prehliadači Mozilla Firefox
- Obr. 10 Ukážka vygenerovanej kruhovej pečiatky s obrázkom

Úvod

Vo svojich počiatkoch slúžil Internet len na šírenie jednoduchých textových dokumentov. S príchodom nových technológií sa ale využitie tejto celosvetovej siete mnohonásobne zväčšilo. Jednou z dôležitých úloh internetu sa postupom času stalo distribuovanie obrazových dát. Koncom 20. storočia sa ako webové štandardy pre vektorovú grafiku používali a dodnes sa ešte aj v určitej miere používajú rôzne značkovacie jazyky ako VML, či PGML. Ich nevýhodou však je, že sú často prepojené len s jedným konkrétnym programom ako napr. v prípade jazyka VML, kde nie je možné zobraziť grafiku na inom internetovom prehliadači ako je Internet Explorer. Vznikla tak potreba existencie jazyka, ktorý by bol podporovaný čo najväčším počtom počítačových firiem. Tým by sa zjednodušila a zefektívnila práca s webovou, vektorovou grafikou.

V roku 1998 sa začal vývoj nového značkovacieho jazyka pod názvom SVG, ktorý prebieha pod patronátom konzorcia W3C. Prvá verzia SVG 1.0 bola odporučená konzorciom W3C v roku 2001. O dva roky neskôr prišla verzia 1.1, ktorá priniesla implementáciu SVG na mobilné zariadenia. Táto verzia obsahuje dva moduly. Modul SVG Basic je určený pre vreckové počítače (PDA) a SVG Tiny pre mobilné telefóny. V súčasnosti sa pripravuje nová verzia SVG 1.2.

Formát SVG okrem toho, že môže zlepšiť vzhľad webových stránok je napríklad výborným prostriedkom na vizualizáciu geodát. Dnes je už totiž často používaným formátom pri tvorbe internetových máp. Výhody SVG oproti iným formátom poskytujú veľmi široký priestor na jeho využitie v rôznych odvetviach hospodárstva ako i v oblasti automatizácie.

Cieľom tejto práce je opísať štruktúru SVG formátu a jeho použitie pri tvorbe dynamických webových stránok.

1. Teoretická časť [1]

1.1 SVG

SVG je konzorciom W3C definované ako jazyk na popísanie dvojdimenzionálnej grafiky a grafických aplikácií v XML. Je to skratka anglického výrazu Scalable Vector Graphics:

- **Scalable** – môžeme preložiť ako stupňovateľný. Znamená to, že objekt sa dá zväčšovať, alebo zmenšovať bez straty kvality. Objekt sa dá upraviť tak, aby sa prispôbil rozlíšeniu obrazovky počítača, či displeju mobilného zariadenia.
- **Vector Graphics** – vektorová grafika, ktorá definuje geometrické tvary pomocou bodov a liniek.

SVG definuje tri základné typy grafických objektov:

- vektorové tvary
- rastrové obrazy
- textové objekty

Všetky tieto objekty sa dajú ľubovoľne zoskupovať, či formátovať napr. pomocou CSS. S vektorovými objektmi sa dá pracovať podobne ako u iných vektorových formátov. To znamená používanie nástrojov na tvorbu čiar, kriviek, či napríklad výplní. Do všetkých objektov sa dajú vkladať filtre, ktorých výhodou je, že sa na daný objekt aplikujú až po ich zobrazení v prehliadači a nezmenia tak pôvodný objekt. Veľkým pozitívom je práca s textom. Textová zložka SVG obrázku je totiž plne editovateľná a má možnosť fulltextového vyhľadávania. Ďalším plusom je, že si uloží tvar daného písma a dokáže ho zobraziť aj na počítači, na ktorom sa táto konkrétna písмовá sada nenachádza.

SVG taktiež podporuje animácie a obrázok v tomto formáte dokáže byť plne interaktívny. Takýto obrázok vie meniť svoju podobu na základe interakcie s užívateľom.

1.2 Štruktúra SVG

SVG má podobne ako iné jazyky svoje syntaktické pravidlá, ktoré treba dodržiavať kvôli bezchybnému zobrazovaniu a funkčnosti. Kód musí obsahovať základné informácie, ktoré zaručia jeho správne zobrazenie:

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
    "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg xmlns="http://www.w3.org/2000/svg"
    xmlns:xlink="http://www.w3.org/1999/xlink">
Vlastný kód
</svg>
```

1.2.1 Základný element – svg

Fragment SVG dokumentu je normou definovaný ako ľubovoľný počet grafických objektov obsiahnutých v elemente svg. Tieto elementy sa môžu nachádzať aj vnútri iných SVG fragmentov, čo znamená, že sa môžu do seba vnárať.

Fragment SVG dokumentu môže existovať ako samostatný súbor, alebo ho môžeme vložiť do iného XML dokumentu, napr. do HTML použitím troch rôznych tagov:

embed

```
<embed src="Ukazka.svg" type="image/svg+xml"/>
```

object

```
<object data="Ukazka.svg" type="image/svg+xml"></object>
```

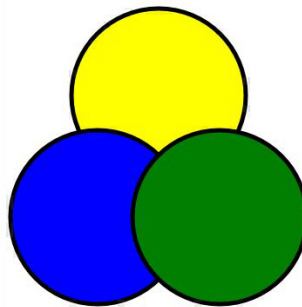
iframe

```
<iframe src="Ukazka.svg"></iframe>
```

1.2.2 Element – g

Element <g> označovaný v angličtine ako container element združuje elementy, ktoré sú v ňom obsiahnuté do skupiny a tie potom dedia jeho vlastnosti. To znamená, že atribúty definované pre <g> budú platiť pre všetky elementy, ktoré sú v ňom obsiahnuté (obr. 1).

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
    "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg xmlns="http://www.w3.org/2000/svg"
    xmlns:xlink="http://www.w3.org/1999/xlink">
  <g style="stroke:black; stroke-width:5px">
    <circle cx="190px" cy="120px" r="100px" style="fill:yellow" />
    <circle cx="120px" cy="260px" r="100px" style="fill:blue" />
    <circle cx="260px" cy="260px" r="100px" style="fill:green" />
  </g>
</svg>
```



Obr. 1 – Tri kruhy, ktoré zdedili vlastnosti (farbu a hrúbku okraju) elementu <g>.

1.2.3 Element – defs

Element <defs> sa používa pre elementy, na ktoré sa chceme neskôr odkázať. Má podobné vlastnosti ako <g>. Zaradenie elementov do <defs> zabezpečí ich rýchlejšie načítanie a zobrazenie v prehliadači (obr. 2).

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
    "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg xmlns="http://www.w3.org/2000/svg"
    xmlns:xlink="http://www.w3.org/1999/xlink">
  <defs>
    <linearGradient id="Prechod">
      <stop offset="20%" stop-color="#39F" />
      <stop offset="90%" stop-color="#F3F" />
    </linearGradient>
  </defs>
  <circle cx="190px" cy="120px" r="100px" style="fill:url(#Prechod)" />
</svg>
```




Obr. 2 – Kruh s lineárnym farebným prechodom, ktorý bol zadenovaný vnútri elementu `<defs>` a následne sa tento prechod zavola v elemente `<circle>`.

1.2.4 Element – `symbol`, `use`

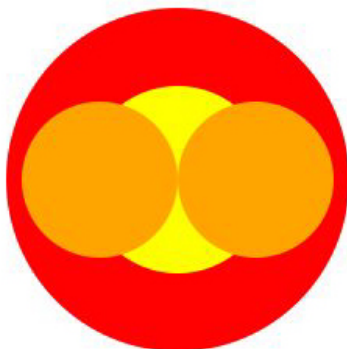
Element `<symbol>` definuje elementy pre viacnásobné použitie, ktoré sú definované atribútom - `id="názov_symbolu"`. Element `<use>` odkazuje na `<symbol>`, resp. na všetky elementy obsiahnuté v `<defs>`.

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
    "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg xmlns="http://www.w3.org/2000/svg"
    xmlns:xlink="http://www.w3.org/1999/xlink">
  <defs>
    <symbol id="Kruhy">
      <circle cx="140px" cy="120px" r="50px" style="fill:orange" />
      <circle cx="240px" cy="120px" r="50px" style="fill:orange" />
    </symbol>
  </defs>
  <circle cx="190px" cy="120px" r="110px" style="fill:red" />
  <use xlink:href="#Kruhy"/>
  <circle cx="190px" cy="120px" r="60px" style="fill:yellow" />
</svg>
```

Z kódového zápisu je vidieť, že element `<use>` je umiestnený medzi dvoma elementmi `<circle>`. Z toho dôvodu sa oranžové kruhy zobrazujú medzi červeným a žltým kruhom (obr. 3). Znamená to, že grafické prvky sa zobrazujú v poradí v akom sú jednotlivé elementy zapísané. Ak umiestnime `<use>` za oba elementy `<circle>`, tak oranžové kruhy budú zobrazené v popredí (obr. 4).



Obr. 3 – Dva oranžové kruhy sú zadefinované pomocou elementu `<symbol>` a pomocou elementu `<use>` sú umiestnené medzi červeným a žltým kruhom.



Obr. 4 – Dva oranžové kruhy sú zadefinované pomocou elementu `<symbol>` a pomocou elementu `<use>` sú umiestnené nad červeným a žltým kruhom.

1.2.5 Element – image

Element `<image>` slúži na vkladanie obrázkov. Okrem obrázkov vo formáte PNG, či JPEG sa môže vložiť aj ďalší SVG súbor [2].

1.3 Základné vektorové tvary

1.3.1 Obdĺžnik (Rectangle)

Element <rect> sa definuje pomocou niekoľkých atribútov. Na určenie súradnicovej polohy sa používajú “x”, “y”. Šírka a výška sa zadáva pomocou “width” a “height”. Pomocou “rx” a “ry” sa dajú zaobľovať rohy.

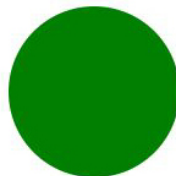
```
<rect x="55" y="55" rx="5" ry="5" width="80" height="50" fill="green"/>
```



1.3.2 Kruh (Circle)

Element <circle> sa definuje pomocou súradníc “cx” a “cy”, ktoré určujú polohu stredu kruhu a polomer sa určí atribútom “r”.

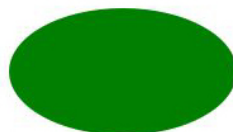
```
<circle cx="100" cy="100" r="70" fill="green"/>
```



1.3.3 Elipsa (Ellipse)

Element <ellipse> sa definuje rovnako ako kruh s tým rozdielom, že má dva polomery a tým pádom aj dva atribúty “rx” a “ry”.

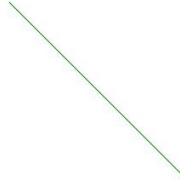
```
<ellipse cx="150" cy="120" rx="90" ry="50" fill="green"/>
```



1.3.4 Čiara (Line)

Element <line> sa definuje pomocou štyroch súradníc – dvoma začiatočnými “x1”, “y1” a dvoma koncovými “x2”, “y2”.

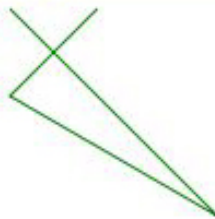
```
<line x1="50" y1="50" x2="255" y2="255" stroke="green"/>
```



1.3.5 Zalomená čiara (Polyline)

Zalomená čiara <polyline> je súbor spojených jednoduchých liniek, ktorých súradnice sa zapisujú do atribútu “points”. Hodnoty môžu byť oddelené medzerami alebo čiarkami, ale pre lepšiu prehľadnosť kódu je vhodné ich oddelovať čiarkami.

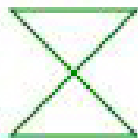
```
<polyline points="5,5 100,100 5,45 45,5" stroke="green" fill="none" />
```



1.3.6 Mnohouholník (Polygon)

Elementom <polygon> sa zapisuje uzavretý vektorový tvar, ktorý sa definuje tak ako element <polyline> pomocou atribútu “points”.

```
<polygon points="5,5 45,45 5,45 45,5" stroke="green" fill="none"/>
```



1.3.7 Cesta (Path)

Cesty sa dajú najlepšie opísať, keď si predstavíme virtuálne pero, ktoré sa posúva po obrazovke počítača. Týmto spôsobom vytvára čiary. Pero sa môže samozrejme zdvihnúť a presunúť na inú pozíciu a odtiaľ opäť pokračovať v kreslení.

Pre posúvanie virtuálneho pera sa používajú nasledujúce príkazy:

M – moveto (nastaví na dané súradnice bez kreslenia čiary)

L – lineto (kreslí čiaru)

H – horizontal link (vykreslí horizontálnu čiaru od počiatočných súradníc)

C – curveto (kreslí kubickú Bézierovu krivku)

A – arc (kreslí výseč v tvare elipsy)

Z – (uzavrie aktuálnu vektorovú plochu)

Všetky príkazy sa zapisujú do atribútu “d” príslušným písmenom, za ktorým nasledujú súradnice.

1.3.8 Bézierova krivka (Bézier curve)

Pri Bézierovej krivke existujú dva príkazy: “C” a “S”.

Cx₁,y₁ x₂,y₂ x₃,y₃

x₁ y₁ – súradnice prvého riadiaceho bodu na začiatku krivky

x₂ y₂ – súradnice druhého riadiaceho bodu na konci krivky

x₃ y₃ – súradnice koncového bodu

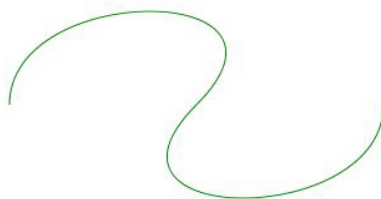
Sx2,y2 x3,y3

x2 y2 – súradnice druhého riadiaceho bodu na konci krivky

x3 y3 – súradnice koncového bodu

Tento príkaz je podobný príkazu “C“. Predpokladá sa však, že prvý riadiaci bod príkazu “S“ je zrkadlovým obrazom druhého riadiaceho bodu z predchádzajúceho príkazu “C“.

```
<path d="M100,200 C100,100 350,100 250,200 S400,300 400,200" stroke="green" fill="none"/>
```



1.4 Text

Možnosti formátovania textu v SVG sú pomerne široké. Či už sa jedná o umiestnenie textu na krivku, alebo zápis japonských znakov.

1.4.1 Element – text

Je to základný element pre vkladanie textu. Pri formátovaní sa používajú atribúty z kaskádových štýlov (CSS).

Niektoré vybrané atribúty:

x, y – súradnice polohy účaria prvého znaku textového reťazca

font-family – druh písma (font písma)

font-style – štýl písma (normal, kurzíva...)

font-variant – normálne, kapitálky

font-size – veľkosť písma (malá, stredná, veľká, percentá...)

font-weight – hrúbka písma

kerninf – veľkosť medzery medzi písmenami

word-spacing – veľkosť medzery medzi slovami

text-decoration – rôzne dekorácie textu

1.4.2 Element – tspan

Element <tspan> umožní priradiť špecifické typografické vlastnosti každému jednotlivému znaku, ktorý sa v tomto elemente nachádza (obr. 5). Využíva k tomu tieto atribúty:

dx

Stanovuje dodatočný posun znakov v rámci aktuálneho elementu <tspan>

dy

Stanovuje dodatočný posun smerom hore/dole v rámci aktuálneho elementu <tspan>

rotate

Stanovuje dodatočné pootočenie znakov

```
<text y="30" style="font-family:Arial; font-weight:bold">
  <tspan style="fill:red" x="5" dx="0 5 10 15 0 0 0 0 -2">
    Ukážka zmeny vzdialenosti medzi jednotlivými znakmi pomocou atribútu dx
  </tspan>

  <tspan style="fill:purple" x="5" dy="30 0 0 0 15 20" rotate="0 0 0 0 10 -25 -60">
    Ukážka rotácie znakov a posunu pomocou atribútu dy
  </tspan>
</text>
```

U k á žka zmeny vzdialenosti medzi jednotlivými znakmi pomocou atribútu dx

Ukáž
k

• rotácie znakov a posunu pomocou atribútu dy

Obr. 5 – Ukážka práce s atribútmi dx, dy a rotate.

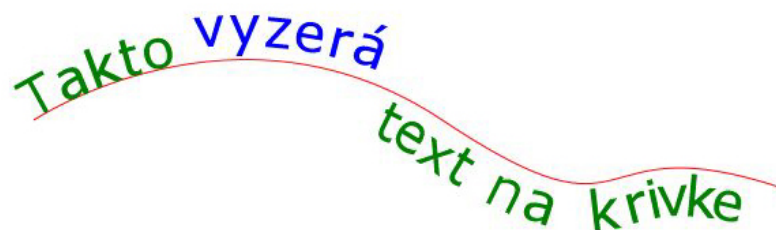
1.4.3 Element – textPath

Umiestnenie textu na krivke patrí medzi pomerne atraktívne efekty práce s textom (obr. 6). Pri jeho tvorbe je dôležité skombinovať znalosti ohľadom práce s vektorovými tvarmi ako i samotného textu. Najskôr si treba zadať cestu (krivku), na ktorú sa text umiestni. Text sa umiestni do elementu <textPath> v ktorom sa pomocou atribútu “xlink:href” odkážeme na predtým vytvorenú krivku.

```
<defs>
<path id="Krivka"
  d="M50,100 C50,100 200,0 350,100 S450,100 600,150"
  style="stroke:red; fill:none"/>
</defs>

<use xlink:href="#Krivka"/>

<text style="fill:green; font-family:Verdana; font-size:250%">
  <textPath xlink:href="#Krivka">
    Takto
    <tspan dy="-10" style="fill:blue">
      vyzerá
    </tspan>
    <tspan dy="45" dx="0 0 0 0 0 0 0 0 0 0 2 4 6 8">
      text na krivke
    </tspan>
  </textPath>
</text>
```



Obr. 6 – Umiestnenie textu na krivke.

2. Praktická časť

2.1 Vytvorenie formulárov

Prvou úlohou je určiť všetky atribúty pečiatky, ktoré si bude môcť zákazník na webovej stránke nastaviť. Po nastavení príslušných parametrov sa vygeneruje obrázok pečiatky vo formáte SVG. Nastavenie atribútov pečiatky sa zabezpečí pomocou PHP formulára.

2.1.1 Formulár pre obdĺžnikovú pečiatku

Pri obdĺžnikovej pečiatke bolo potrebné zvoliť nasledujúce parametre:

- Farbu pozadia/podušky
- Typ písma
- Veľkosť písma
- Štýl písma
- Text, resp. texty pečiatky

Všetky atribúty sa zapíšu do formulára, ktorý sa potom spracuje pomocou jazyka PHP.

```
$text .= "<form action='generovanie_peciatky.php'
        name='form' method='get'>\n";
$text .= "<div class='number'>Farba</div>\n";
$text .= "<div class='select'>
        <select name='Color'>
        <option value='red'>Červená
        <option value='blue'>Modrá
        </select></div>";
$text .= "<div class='number'>Typ písma</div>\n";
$text .= "<div class='select'>
        <select name='Fonttype'>
        <option value='Times'>Times New Roman
        <option value='Courier'>Courier New
        <option value='Century Gothic'>Centhury Gothic
        <option value='Arial'>Arial
        </select></div>";
$text .= "<div class='number'>Veľkosť písma v bodoch</div>\n";
$text .= "<div class='select'>
        <select name='Size'>
        <option value='8'>8
        <option value='9'>9
        <option value='10'>10
        <option value='11'>11
        <option value='12'>12
        </select></div>";
```

```

$text .= "<div class='number'>Štýl písma</div>\n";
$text .= "<div class='input'>
        <input type='checkbox' name='style1' value='bold'>Tučné</div>";
$text .= "<div class='input'>
        <input type='checkbox' name='style2' value='italic'>Kurzíva</div>";
$text .= "<div class='number'>Text č.1</div>\n";
$text .= "<div class='input'>
        <input type='text' name='t1' size='40'></div>";
$text .= "<div class='number'>Text č.2</div>\n";
$text .= "<div class='input'>
        <input type='text' name='t2' size='40'></div>";
$text .= "<div class='number'>Text č.3</div>\n";
$text .= "<div class='input'>
        <input type='text' name='t3' size='40'></div>";
$text .= "<div class='number'>Text č.4</div>\n";
$text .= "<div class='input'>
        <input type='text' name='t4' size='40'></div>";
$text .= "<div class='number'>Text č.5</div>\n";
$text .= "<div class='input'>
        <input type='text' name='t5' size='40'></div>";
$text .= "<p><input type='submit' value='Odoslať'></p>";
$text .= "</form>";

```

2.1.2 Formulár pre kruhovú pečiatku

Pri kruhovej pečiatke bolo potrebné zvoliť nasledujúce parametre:

- Farbu pozadia/podušky
- Typ písma
- Veľkosť písma
- Štýl písma
- Text, resp. texty pečiatky

Všetky atribúty pečiatky sa podobne ako u obdĺžnikovej pečiatky zapíšu do formulára, ktorý sa potom spracuje pomocou jazyka PHP. Malá zmena sa týka formulárových prvkov pre vkladanie textu, či textov.

```

$text .= "<div class='number'>Horný text</div>\n";
$text .= "<div class='input'><input type='text' name='t' size='40'></div>";
$text .= "<div class='number'>Dolný text</div>\n";
$text .= "<div class='input'><input type='text' name='t1' size='40'></div>";
$text .= "<div class='number'>Stredný text</div>\n";
$text .= "<div class='input'><input type='text' name='t2' size='40'></div><br>";
$text .= "<div class='input'><input type='file' name='file' id='file'></div>";
$text .= "<p><input type='submit' name='submit' value='Odoslať'></p>";

```

2.2 Vytvorenie obdĺžnikovej pečiatky

Ak je formulár s príslušnými parametrami pečiatky hotový, je potrebné vytvoriť SVG kód. Ten na základe vložených údajov od užívateľa zhotoví obraz pečiatky, ktorý sa následne vykreslí v internetovom prehliadači. Samotná pečiatka je síce tvorená v SVG kóde, ale aby mohla byť vytvorená na základe vložených údajov z formulára, je potrebné tieto údaje zakomponovať vo forme PHP kódu. Zabezpečí sa to tak, že celý súbor v ktorom sa vygeneruje pečiatka bude mať príponu .php. Aby dokázal daný PHP kód pracovať s SVG jazykom je treba na začiatok celého súboru vložiť tento zápis:

```
<?php
header("Content-type: image/svg+xml");
echo '<?xml version="1.0" encoding="UTF-8" standalone="no"?>';
?>
```

2.2.1 Podklad pečiatky

Prvým krokom je vytvorenie obdĺžnika, ktorý posluží ako obraz pečiatky:

```
<rect x="0" y="0" width="84.98" height="56.65"
fill="<?php echo $_GET['Color'];?>" />
```

Ako vidno z ukážky kódového zápisu, obdĺžnik bude mať pevne stanovenú šírku a výšku v pixeloch. Oba rozmery sa nastaví podľa formátu pečiatky, ktorá sa má zobraziť. V tomto prípade sa zobrazí pečiatka o rozmeroch 30x20mm. Vyplnená bude farbou, ktorú si zákazník sám zvolí pri vyplňovaní formulára. Nastavenie farby pečiatky sa zabezpečí tak, že miesto pevnej hodnoty atribútu *fill* sa vloží PHP kód.

2.2.2 Cesty

Ako ďalší krok musíme zdefinovať cesty (paths), ktoré poslužia ako riadky pre texty na pečiatke. Počet ciest sa zvolí podľa rozmeru pečiatky a podľa veľkosti písma. V našom prípade má pečiatka rozmery 30x20mm (84,98x56,65px) a tak je optimálne zvoliť maximálne päť ciest. To znamená, že na pečiatke sa bude dať vyobraziť až päť riadkov textu. Koľko riadkov textu sa však bude môcť reálne na pečiatke zobraziť bude záležať od nastavenej veľkosti písma. Ak si zákazník zvolí veľkosť v rozmedzí 8 až 10 bodov bude možné zobraziť päť riadkov textu. V prípade, že si zvolí veľkosť 11 alebo 12 bodov, počet riadkov sa zníži na číslo štyri.

```

<defs>
<path id="Ciara" d="M-2,<?php switch($_GET['Size'])
{
    case '8' :
        echo "10";
        break;
    case '9' :
        echo "11";
        break;
    case '10' :
        echo "12";
        break;
    case '11' :
        echo "13";
        break;
    case '12' :
        echo "16";
        break;
    default:
        echo "10";
}
?> H81" style="stroke:none; fill:none" />
</defs>

```

Ako vidno z kódového zápisu, zadefinovanie cesty v podstate spočíva len v zadaní začiatkových a konečných súradníc. Namiesto hodnoty začiatkovej y-súradnice sa však použije podmienka *switch*, ktorá zvolí súradnicu podľa toho, akú veľkosť písma si nastavil zákazník. Identickým spôsobom sa zadefinujú cesty aj pre druhý, tretí a štvrtý riadok textu. Pre piaty riadok však treba nastaviť, aby sa v prípade veľkosti fonu 11 až 12 bodov daná cesta nezobrazovala. Inak by sa totiž text zobrazoval mimo formátu pečiatky. V príkaze H, ktorý vytvorí horizontálnu čiaru po zadaní x-súradnicu miesto pevnej hodnoty súradnice vložíme opäť podmienku *switch*. Pomocou neho sa v prípade fonu o veľkosti 11-12 bodov nastaví nulová hodnota súradnice. Vďaka tomu sa cesta nevytvorí a na pečiatke sa nebude nachádzať piaty riadok.

```

H<?php switch($_GET['Size'])
{
    case '8' :
        echo "81";
        break;
    case '9' :
        echo "81";
        break;
    case '10' :
        echo "81";
        break;
    case '11' :
        echo "0";
        break;
    case '12' :
        echo "0";
        break;
    default:
        echo "81";
}

```

2.2.3 Text

Poslednou časťou tvorby pečiatky je zadenovanie všetkých vlastností textu. Ešte predtým je ale potreba vložiť zadenovanú cestu, ktorá je zatiaľ iba definovaná v tagu <defs>. Na to použijeme nasledovný zápis:

```
<use xlink:href="#Ciara" />
```

Zákazník má možnosť si na webovej stránke nastaviť vlastnosti textu, ktoré sú spomenuté v časti 2.1.1. Nastavenie týchto parametrov sa opäť vykoná tak, že namiesto pevnej hodnoty príslušného atribútu sa vloží PHP kód.

Nastavenie typu písma

```
font-family="<?php echo $_GET['Fonttype'];?>"
```

Nastavenie veľkosti písma

```
font-size="<?php echo $_GET['Size'];?>pt"
```

Nastavenie štýlov písma

```
font-weight="<?php echo $_GET['style1'];?>"
```

```
font-style="<?php echo $_GET['style2'];?>"
```

Jednotlivé vlastnosti textu sú nastavené a tak už iba zostáva umiestniť samotný text na neviditeľnú čiaru, ktorá je vytvorená pomocou cesty. Docielime to vložením tohto kódu:

```
<textPath xlink:href="#Ciara">  
<?php echo $_GET['t1']; ?>  
</textPath>
```

Postup opísaný v tejto časti sa vzťahuje len na nastavenie a vloženie jedného riadku textu na pečiatke. Preto je treba ho zopakovať pre každý jeden riadok textu.

2.2.4 Výsledná ukážka

Ak máme vytvorený formulár, ktorého kódový zápis je uvedený v časti 2.1.1 môžeme ho zobraziť v internetovom prehliadači (obr. 7). Z obrázka vidíme, že pečiarka bude mať červený podklad, typ písma bude Centhury Gothic a text bude mať veľkosť 9 bodov. Jednotlivé texty pečiatky sú zobrazené v kolónkach Text č. 1 až Text č. 5. Ak takto nastavený formulár odošleme, zobrazí sa nám požadovaná pečiarka (obr. 8).



Mozilla Firefox

Súbor Upraviť Zobraziť História Záložky Nást

http://local

Najobľúbenejšie Jak začít Přehled zpráv

Farba
Červená

Typ písma
Centhury Gothic

Veľkosť písma v bodoch
9

Štýl písma
☐ Tučné
☐ Kurzíva

Text č. 1
Ukážka jedna

Text č. 2
Ukážka dva

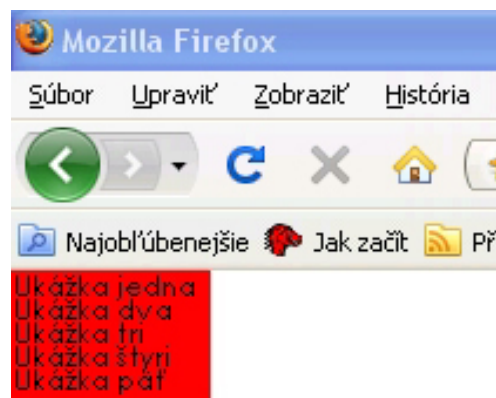
Text č. 3
Ukážka tri

Text č. 4
Ukážka štyri

Text č. 5
Ukážka päť

Odoslať

Obr. 7 – Ukážka formulára v prehliadači Mozilla Firefox.



Obr. 8 – Ukážka vygenerovanej obdĺžnikovej pečiatky.

2.3 Vytvorenie kruhovej pečiatky

Začiatok postupu tvorby kruhovej pečiatky je totožný s tým, ktorý bol použitý aj u obdĺžnikovej pečiatky v kapitole 2.1.

2.3.1 Podklad pečiatky

Na začiatok je treba vytvoriť kruh, ktorý posлúži ako obraz pečiatky:

```
<circle cx="176" cy="200" r="80" fill="<?php echo $_POST['Color'];?>" />
```

Kruhový podklad bude mať presne stanovený rozmer v závislosti od požadovanej pečiatky a jeho farba sa nastaví podľa hodnoty zadanej zákazníkom.

2.3.2 Cesty

Pri kruhovej pečiatke je možné zobraziť tri samostatné texty. Jeden text sa umiestni oblúkovito v hornej polovici, druhý sa zobrazí taktiež oblúkovito v dolnej polovici a tretí text sa bude nachádzať v strede pečiatky. Taktiež bude možnosť vložiť do stredu miesto textu obrázok. Aby sme zadane texty mohli umiestniť potrebujeme si vytvoriť tri cesty.

Prvá cesta bude slúžiť ako účarie pre text, ktorý sa bude nachádzať v hornej časti pečiatky a druhá cesta posлúži pre text v spodnej časti. Obe cesty budú mať tvar pol kružnice a preto je ich definícia komplikovanejšia ako u tretej cesty, ktorá je len obyčajnou úsečkou. Zadefinovanie a vloženie ciest je znázornené v nasledujúcom kódovom zápise:

```
<defs>
<path id="Krivka1" d="M120,205 C120,125 230,125 230,200" style="stroke: none; fill: none" />
</defs>

<defs>
<path id="Krivka2" d="M111,205 C120,280 230,280 237,210" style="stroke: none; fill:none" />
</defs>

<defs>
<path id="Krivka3" d="M126,203 L215,203" style="stroke: none; fill:none" />
</defs>

<use xlink:href="#Krivka1" />
<use xlink:href="#Krivka2" />
<use xlink:href="#Krivka3" />
```


2.3.3 Text

Ak máme vytvorené cesty môžeme prejsť na nastavenie a vloženie textu. Použijeme ten istý postup ako pri obdĺžnikovej pečiatke, spomenutý v podkapitole 2.2.3.

Nastavenie typu písma

```
font-family="<?php echo $_POST['Fonttype'];?>"
```

Nastavenie veľkosti písma

```
font-size="<?php echo $_POST['Size'];?>pt"
```

Nastavenie štýlov písma

```
font-weight="<?php echo $_POST['style1'];?>"
```

```
font-style="<?php echo $_POST['style2'];?>"
```

Tieto nastavenia sa vzťahujú len na jeden zadaný text a preto sa tento postup musí zopakovať aj pre zvyšné texty.

2.3.4 Vloženie obrázka

Pečiatka môže okrem textov obsahovať aj obrázok. V našom prípade bude obrázok umiestnený v strede pečiatky. Skôr ako sa bude dať obrázok vložiť do samotnej pečiatky, musí sa preniesť z užívateľovho počítača na server, na ktorom je umiestnená web stránka. Najlepšie je vytvoriť samostatný adresár, do ktorého sa budú obrázky ukladať. Načítanie obrázka sa zabezpečí pomocou formulárového prvku:

```
$text .= "<div class='input'><input type='file' name='file' id='file'></div>";
```

Ďalšou úlohou je načítaný obrázok uložiť. Na to využijeme znovu PHP kód, v ktorom si na začiatku nastavíme podmienky, ktoré obrázok musí spĺňať, aby mohol byť ďalej spracovaný. Obrázok teda musí byť vo formáte jpeg, alebo gif a jeho veľkosť musí byť nižšia ako 20kb [3]. Až keď sa splnia tieto podmienky, môže byť obrázok uložený.

```
<?php
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/pjpeg")))
&& ($_FILES["file"]["size"] < 20000))
{
    if ($_FILES["file"]["error"] > 0)
    {
        echo "Return Code: " . $_FILES["file"]["error"] . "<br />";
    }
    else
    {
        if (file_exists("Upload Images/" . $_FILES["file"]["name"]))
        {
            echo $_FILES["file"]["name"] . " subor existuje. ";
        }
        else
        {
            move_uploaded_file($_FILES["file"]["tmp_name"],
            "Upload Images/" . $_FILES["file"]["name"]);
            echo "Subor je ulozeny v: " . "Upload Images/" . $_FILES["file"]["name"];
        }
    }
}
else
{
    echo "Chybny subor";
}
??>
```

Posledným krokom je vloženie obrázka do pečiatky. Na tento účel sa používa tag <image> [2]. Aby bol obrázok umiestnený v strede musíme zadať jeho presné súradnice a takisto aj jeho šírku a výšku. Nakoniec zadáme adresár a pomocou PHP kódu aj jeho názov.

```
<image x="130" y="175" width="90px" height="50px"
xlink:href="Upload Images/<?php echo $_FILES['file']['name']; ?>" />
```

2.3.5 Výsledná ukážka

Keď máme vytvorený formulár, ktorého kódový zápis je uvedený v časti 2.1.1 a 2.1.2, môžeme ho zobrazíť v internetovom prehliadači (obr. 9). Z obrázka vidíme, že pečiatka bude mať modrý podklad, typ písma bude Arial, text bude zobrazený tučne o veľkosti 9 bodov. Na pečiatke bude horný a dolný text a v strede bude umiestnený obrázok. Ak takto nastavený formulár odošleme, zobrazí sa nám požadovaná pečiatka (obr. 10).

Mozilla Firefox

Súbor Upraviť Zobrazit' História Záložky Nástroje Pomocník

http://localhost/bc formulare/Okruh

Najobľúbenejšie Jak začít Přehled zpráv

http://localhos...load%20img.php

Farba
Modrá

Typ písma
Arial

Veľkosť písma v bodoch
12

Štýl písma
☒ Tučné
☐ Kurzíva

Horný text
Ukážka horného textu

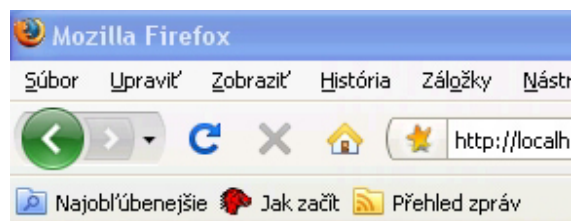
Dolný text
Ukážka dolného textu

Stredný text

C:\Bakalarka\ychpt_logc Prehľadávať...

Odoslať

Obr. 9 – Ukážka formulára v prehliadači Mozilla Firefox.



Obr. 10 – Ukážka vygenerovanej kruhovej pečiatky s obrázkom.

3. Záver

Úlohou práce bolo vytvoriť SVG modul, pomocou ktorého by sa v internetovom prehliadači generovali obrázky pečiatok na základe nastavenia ich parametrov.

Pred začatím samotnej práce bolo treba určiť aké parametre pečiatok sa budú dať nastaviť. Až na základe týchto informácií sa potom vytvorili formuláre. Vytvorenie daných formulárov spočívalo v skombinovaní XHTML a PHP jazyka, ktoré boli súčasťou výučby v letnom semestri. Z tohto dôvodu ich tvorba nebola natoľko náročná. Komplikovanejším problémom bolo ale zabezpečiť prepojenie SVG a PHP kódu. Po preštudovaní rôznych materiálov sa ale podarilo tento problém vyriešiť a celý projekt sa tak mohol dokončiť.

SVG grafika sa v určitých oblastiach internetu už udomácnila, avšak jej možnosti poskytujú oveľa väčší priestor na uplatnenie. Aby sa tak udialo, je ale potrebná lepšia podpora vývojárskych firiem. V najväčšej miere sa to týka hlavne internetových prehliadačov.

Použitá literatura

- [1] <http://interval.cz/serialy/kurz-svg-tvorba-vektorove-grafiky-v-xml>, Kurs SVG – Tvorba vektorové grafiky v XML
- [2] J. David Eisenberg, SVG Essentials, O'Reilly Media 2002
- [3] <http://www.w3schools.com/php/default.asp>, PHP Tutorial

Prílohy

Zdrojové kódy aplikácie, ako aj elektronická verzia bakalárskej práce sa nachádzajú na priloženom DVD.