

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## NÁSTROJ PRO SLEDOVÁNÍ ZMĚN WEBU

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

HYNEK ŠVÁBEK

BRNO 2008



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## **NÁSTROJ PRO SLEDOVÁNÍ ZMĚN WEBU**

APPLICATION FOR MONITORING CHANGES OF THE WEB

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**HYNEK ŠVÁBEK**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. LUKÁŠ GRULICH**

BRNO 2008

## Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav inteligentních systémů

Akademický rok 2007/2008

### Zadání bakalářské práce

Řešitel: **Švábek Hynek**

Obor: Informační technologie

Téma: **Nástroj pro sledování změn webu**

Kategorie: Počítačové sítě

Pokyny:

1. Analyzujte požadavky kladené na požadovaný nástroj. Cílem je vytvořit aplikaci, která bude v daných intervalech načítat zadanou webovou stránku (stránky), vyhodnocovat a nějakým způsobem zpracovávat změny.
2. Navrhněte strukturu a uživatelské rozhraní takové aplikace a vhodné implementační prostředí.
3. Aplikaci implementujte, testujte, demonstруйте na vhodném příkladě.

Literatura:

- Dle pokynů vyučujícího.

Při obhajobě semestrální části projektu je požadováno:

- První dva body zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Grulich Lukáš, Ing.**, UITS FIT VUT

Datum zadání: 1. listopadu 2007

Datum odevzdání: 14. května 2008

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav informačních systémů  
612 66 Brno, Božetěchova 2

  
\_\_\_\_\_  
doc. Dr. Ing. Petr Hanáček  
vedoucí ústavu

# LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

## 1. Pan

Jméno a příjmení: **Hynek Švábek**  
Id studenta: 78714  
Bytem: Kostelní 803, 696 32 Ždánice  
Narozen: 29. 04. 1985, Kyjov  
(dále jen "autor")

a

## 2. Vysoké učení technické v Brně

Fakulta informačních technologií  
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305  
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....  
(dále jen "nabyvatel")

### Článek 1

#### Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):  
bakalářská práce

Název VŠKP: Nástroj pro sledování změn webu

Vedoucí/školitel VŠKP: Grulich Lukáš, Ing.

Ústav: Ústav inteligentních systémů

Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

tištěné formě                      počet exemplářů: 1

elektronické formě                počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

## Článek 2 Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
  - ihned po uzavření této smlouvy
  - 1 rok po uzavření této smlouvy
  - 3 roky po uzavření této smlouvy
  - 5 let po uzavření této smlouvy
  - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....

Nabyvatel

  
.....

Autor

## Abstrakt

Cílem této bakalářské práce je vytvoření aplikace Nástroj pro sledování změn webu, která v určitých časových intervalech kontroluje zadané internetové stránky a o změnách dává uživateli určitým způsobem vědět. V návrhu popisuji, co vše bylo nutno vyřešit před vlastní implementací, a dále také popisuji technologie použité k realizaci aplikace. V další části textu nazvané Implementace se zabývám popisem implementace jednotlivých částí aplikace. Implementace je provedena pomocí jazyka Java za využití knihoven dom4, systray4j. Aplikace dále využívá technologie XML a pro překlad a spuštění aplikace je použit program Apache Ant.

## Klíčová slova

web, stránky, Java, html, css, http, xml

## Abstract

The target of this bachelor thesis is creation of application Application for Monitoring Changes of the Web, which checks the changes of the web page in given time intervals and notifies the user about these changes. In a design section I describe, what had to be done before implementation and I also describe technologies used in application. The next part of text named Implementace considers implementation of application parts. Implementation has been done using Java language and libraries dom4, systray4j. Application also uses XML technology and Apache Ant application for compiling and running.

## Keywords

web, pages, Java, html, css, http, xml

## Citace

Hynek Švábek: Nástroj pro sledování změn webu, bakalářská práce, Brno, FIT VUT v Brně, 2008

# Nástroj pro sledování změn webu

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Lukáše Grulicha

.....  
Hynek Švábek  
11. května 2008

© Hynek Švábek, 2008.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Úvod do použitých technologií</b>	<b>4</b>
2.1	HTML 4.01	4
2.2	CSS	4
2.3	Protokol HTTP	5
2.4	Knihovna java-diff	6
2.5	XML	7
2.6	Apache Ant	8
2.7	Swing	9
2.8	SysTray for Java	9
<b>3</b>	<b>Analýza a návrh</b>	<b>10</b>
3.1	Požadavky	10
3.1.1	Funkčnost	10
3.1.2	Konfigurovatelnost	10
3.1.3	Indikace změn	10
3.1.4	Přenositelnost	11
3.1.5	Rozšiřitelnost	11
3.2	Volba implementačního prostředí	11
3.3	Stažení stránky	11
3.3.1	Volba http protokolu	11
3.3.2	Návrh adresářové struktury	12
3.4	Znovunačtení za časový interval	12
3.4.1	Aktivní čekací smyčka	12
3.4.2	Vlákna	13
3.4.3	Časovač a vlákna	13
3.5	Vyhledání změn	13
3.6	Možnosti zvýraznění	13
3.7	Ukládání konfigurace	13
3.7.1	Textový soubor	14
3.7.2	XML soubor	14
3.8	System Tray	14
3.8.1	Java 1.6	14
3.8.2	Knihovna jdic	14
3.8.3	Knihovna Systray for Java	14



<b>4</b>	<b>Implementace a testy</b>	<b>15</b>
4.1	Balík main	15
4.2	Balík zmeny	15
4.2.1	Třída CSS	16
4.2.2	Třída Databaze	16
4.2.3	Třída Download	17
4.2.4	Třída Img	17
4.2.5	Třída Konfigurace	17
4.2.6	Třída Porovnani	17
4.2.7	Třída Prohlizec	18
4.2.8	Třída Soubory	18
4.2.9	Třída VlaknoStahovani	18
4.3	Balík gui	19
4.3.1	Třída ChybaApp	19
4.3.2	Třída KonfiguraceApp	19
4.3.3	Třída MenuAboutBox	20
4.3.4	Třída MenuView	20
4.3.5	Třída PopupOkno	22
4.3.6	Třída PridejApp	23
4.3.7	Třída SysTray	24
4.4	Balík xml	25
4.4.1	Třída UlozXML	25
4.4.2	Třída NactiXML	26
4.5	Překlad a spuštění pomocí Apache Ant	26
4.6	Testy	26
4.6.1	Test č. 1	26
4.6.2	Test č. 2	27
4.6.3	Test č. 3	27
<b>5</b>	<b>Závěr</b>	<b>29</b>
<b>A</b>	<b>Přílohy</b>	<b>32</b>
A.1	Nápověda	32
A.2	Médium CD	36

# Kapitola 1

## Úvod

Tato práce popisuje návrh a tvorbu nástroje pro sledování změn webu. V kapitole (2) jsou popsány prostředky, které byly potřeba ke správné analýze, návrhu a implementaci požadovaného nástroje. Je zde popsán univerzální značkovací jazyk HTML 4.01 a s ním související kaskádové styly, dále protokol http, který slouží k přenosu internetových stránek, knihovna sloužící k porovnávání textu, obecný značkovací jazyk XML a je zde také popsáno, k čemu slouží aplikace Apache Ant. Další podkapitoly popisují grafické uživatelské rozhraní Swing v Javě a knihovnu pro umístění aplikace do system tray.

V následující části textu (3) je popsána analýza a návrh zadané aplikace, a to zhodnocení základních požadavků na Nástroj pro sledování změn webu a také volba vhodného implementačního prostředí. Dále se zde zaměřuji na možnosti stahování internetových stránek a jejich znovunačítání za časový interval. Je zde uvedeno, jakými způsoby lze vyhledávat změny na kontrolovaných internetových stránkách a jak tyto nalezené změny zvýraznit. Jedna z podkapitol se zabývá možnostmi ukládání nastavení aplikace do souboru.

V další kapitole (4) popisují realizaci aplikace, tedy implementaci jednotlivých tříd obsažených v balících. Je zde popsán například balík `main`, obsahující pouze jednu třídu `Main`, která slouží ke spuštění aplikace, balík `zmeny`, obsahující třídy, jež slouží ke stažení internetových stránek, k nalezení změn a k jejich zvýraznění, a třídy sloužící ke stažení obrázků a kaskádových stylů uvedených na kontrolované internetové stránce. Dále zde popisují balík `gui`, jenž implementuje grafické uživatelské rozhraní pomocí knihovny Swing. Tento balík obsahuje třídu `ChybaApp`, sloužící k zobrazení dialogového okna s popisem chyby, třídu `KonfiguraceApp`, realizující grafické uživatelské rozhraní pro třídu `Konfigurace`, třídu `MenuAboutBox`, implementující dialogové okno s informacemi o aplikaci, třídu `MenuView`, implementující hlavní okno aplikace, třídu `PopupOkno`, realizující vyskakovací popup okno, třídu `PridejApp`, která implementuje přidání nové kontrolované stránky do databáze seznamu stránek a editaci již existující kontrolované internetové stránky v databázi, a třídu `SysTray`, implementující rozhraní aplikace v oblasti system tray. Poslední balík `xml` obsahuje dvě třídy. První z nich `UlozXML` realizuje ukládání konfigurace a seznamu kontrolovaných stránek do xml souboru. Naproti tomu druhá třída `NactiXML` implementuje načítání konfigurace a seznam stránek ze souboru xml. Podkapitola (4.5) pojednává o tom, jak výslednou aplikaci přeložit a spustit pomocí programu Apache Ant.

Závěrečná kapitola (5) zhodnocuje přínosy aplikace a zabývá se jejím možným budoucím rozšířením.

## Kapitola 2

# Úvod do použitých technologií

V této kapitole jsou popsány a vysvětleny principy, na kterých je postaven Nástroj pro sledování změn webu. V jednotlivých podkapitolách textu se čtenář dozví, jak takový nástroj pracuje a jakým způsobem se konfigurace ukládá na pevný disk.

### 2.1 HTML 4.01

Jazyk HTML je HyperText Markup Language, neboli hypertextový jazyk značek (tagů), a je čistě textový a tudíž nezávislý na platformě. Slouží k vytváření internetových stránek. Umožňuje do nich vkládat odkazy na jiné stránky, formátovat text, vkládat obrázky, tabulky, formuláře, objekty, skripty atd. Vše se provádí pomocí značek (tagů), které jsou uzavřeny mezi znaky `<` a `>`. Většina tagů je párová, takže existuje úvodní i koncová značka např.: `<tag>text</tag>`.

Nejnovější verze HTML 4.01 [16] je zpětně kompatibilní s verzí HTML 3.2. Pro zobrazení html stránky je potřeba internetový prohlížeč (Internet explorer, Mozilla Firefox, Opera, Safari). Ověřit si, zda je stránka validní, je možné pomocí skriptu označovaného jako validátor, který zkontroluje zdrojový text podle přesně stanovených pravidel. Je ho možné najít třeba na adrese <http://validator.w3.org/>. Jednoduchá internetová stránka může vypadat například takto:

```
<html>
<head><title> Titulek stránky </title></head>
<body>
```

Text dokumentu

```
</body>
</html>
```

Více o html můžete najít na [19]

### 2.2 CSS

CSS [15] je zkratka z Cascading Style Sheets, česky kaskádové styly, které vznikly kvůli oddělení obsahu od vzhledu. Dříve byla definice vzhledu přímo v html souboru pomocí k tomu určených tagů. Například `<b>tučný</b>` pro tučný text, `<i>kurzíva</i>` pro kurzívu

atd. Tento způsob začal být nevyhovující, proto vzniklo CSS, které je možno zapisovat třemi způsoby:

- Přímo v textu zdrojového souboru u formátovaného elementu pomocí atributu `style=" "`

```
<h1 style="color: blue">Nadpis první úrovně bude modrý</h1>
```

- Pomocí stylopisu (stylesheet), jenž se zapisuje mezi tagy `<style></style>` uvedenými v hlavičce dokumentu. Mezi těmito tagy je uvedeno, jakým způsobem má být který element naformátován.

```
<style>
h1    {color: blue}
</style>
```

- Pomocí externího stylopisu uvedeného v souboru `*.css`, na který se stránka odkazuje tagem `<link>`. V souboru pojmenovaném například `style.css` je pak uvedeno:

```
h1    {color: red}
```

Tento soubor se připojí k dokumentu pomocí tagu `<link rel="stylesheet" href="style.css">` v hlavičce dokumentu.

Poslední uvedený způsob je nejlépe využitelný. Jeden css soubor je totiž možno použít pro více stránek, které poté mají podobný vzhled, a změnou jednoho css souboru můžeme dosáhnout změny celého webu, aniž bychom pracně měnili všechny zdrojové html soubory. Další informace o kaskádových stylech jsou k nalezení na [9].

## 2.3 Protokol HTTP

HTTP (Hypertext Transfer Protocol) je internetový protokol původně navržen a implementován pro výměnu hypertextových dokumentů ve formátu HTML. Standardně běží pod portem TCP číslo 80. V současné době je používána převážně verze HTTP 1.1, která je definována v RFC 2616 [8], ale také verze HTTP 1.0 [7]. Tento protokol je společně s elektronickou poštou nejvyužívanější službou internetu a stojí za jeho velkým rozšířením v posledních letech. V současné době je používán i pro přenos dalších informací. Pomocí rozšíření MIME umí přenášet jakýkoli soubor, používá se společně s formátem XML pro tzv. webové služby (spouštění vzdálených aplikací) a pomocí aplikačních bran zpřístupňuje i další protokoly, jako je např. FTP nebo SMTP. HTTP používá jako některé další aplikace tzv. jednotný lokátor prostředků (URL, Uniform Resource Locator), jenž specifikuje jednoznačné umístění nějakého zdroje v Internetu. K protokolu HTTP existuje také jeho bezpečnější verze HTTPS, která umožňuje přenášená data šifrovat a tím chránit před odposlechem či jiným narušením.

Protokol funguje způsobem dotaz-odpověď. Uživatel (pomocí programu, obvykle internetového prohlížeče) pošle serveru dotaz ve formě čistého textu, obsahujícího označení požadovaného dokumentu, informace o schopnostech prohlížeče apod. Server poté odpoví pomocí několika řádků textu popisujících výsledek dotazu, za kterými následují data samotného požadovaného dokumentu.

Pokud uživatel bude mít po chvíli další dotaz na stejný server, bude se jednat o další, nezávislý dotaz a odpověď. Z hlediska serveru nelze poznat, zda tento druhý dotaz jakkoli souvisí s předchozím. Kvůli této vlastnosti se protokolu HTTP říká bezstavový protokol, jelikož neumí uchovávat stav komunikace – dotazy spolu nemají souvislost. Tato vlastnost je nepříjemná pro implementaci složitějších procesů přes HTTP (např. internetový obchod potřebuje uchovávat informaci o identitě zákazníka, o obsahu jeho nákupního košíku apod.). K tomuto účelu byl protokol HTTP rozšířen o tzv. HTTP cookies, které umožňují serveru uchovávat si informace o stavu spojení na počítači uživatele.

Ukázka komunikace přes telnet

Uživatel zadá:

```
telnet www.seznam.cz 80
Trying 77.75.72.3...
Connected to www.seznam.cz.
Escape character is '^]'.
GET / HTTP/1.0
Host: www.seznam.cz
```

Server odpoví:

```
HTTP/1.0 200 OK
Connection: close
Date: Sat, 19 Apr 2008 11:38:31 GMT
Server: Apache/2.2.3 (Debian)
Cache-Control: max-age=0, no-cache, must-revalidate
Expires: Sat, 19 Apr 2008 11:38:31 GMT
Pragma: no-cache
Content-Type: text/html; charset=UTF-8
X-Pad: avoid browser bug
Content-Length: 41793
```

Za touto hlavičkou následuje html dokument oddělen znaky `\r\n\r\n`, tedy oddělen jedním prázdným řádkem. Tuto hlavičku internetové prohlížeče automaticky odstraňují. Pro její získání je nutno použít program telnet či online sniffer mapř. <http://web-sniffer.net/>.

Více informací je k nalezení na [20].

## 2.4 Knihovna java-diff

Knihovna java-diff [10] je implementována jako dvě třídy. Porovnává dvě kolekce. Hlavní metoda `diff()` z třídy `Diff` vrací seznam rozdílných objektů v datovém typu `List`. Každý objekt obsahuje popis přidání, smazání nebo změny mezi dvěma kolekcemi.

Příklad použití knihovny:

```
Object[] a = new Object[] {
    "a",
    "b",
    "c",
    "d",
```

```

        "e"
    };

    Object[] b = new Object[] {
        "a",
        "x",
        "y",
        "b",
        "c",
        "j",
        "e",
    };

    Difference[] expected = new Difference[] {
        new Difference(1, -1, 1, 2),
        new Difference(3, 3, 5, 5),
    };

    Diff diff = new Diff(a, b);
    List diffOut = diff.diff();

```

Knihovna Diff podporuje porovnávání objektů s imlementovaným rozhraním Comparable nebo těch objektů, jejichž komparátor byl předán do konstruktoru objektu Diff. Pokud objekt neimplementuje rozhraní Comparator, tak místo toho bude použito porovnání pomocí hash kódu.

## 2.5 XML

Zkratka XML znamená eXtensible Markup Language. Jazyk XML je obecný značkovací jazyk, jenž byl vyvinut konsorciem W3C a také jím byl standardizován. XML umožňuje snadné vytváření značkovacích jazyků pro nejrůznější účely a velkého množství různých datových typů. Je převážně určen pro publikování dokumentů, sdílení dat mezi aplikacemi či ukládání konfigurace programů. Jazyk popisuje strukturu dokumentu z hlediska věcného obsahu, vzhledem se nezabývá – ten je možno definovat v připojeném stylu (např. CSS). Dokument ve formátu XML je také možno transformovat do jiné struktury XML pomocí jazyků rodiny XSL (eXtensible Stylesheet Language).

XML je založen na jednoduchém textu a je zpracovatelný libovolným textovým editorem. Specifikace je zdarma přístupná a každý si může do svých aplikací bez problémů implementovat podporu XML narozdíl od jiných formátů, které jsou většinou binární a nemají k dispozici dokumentaci. Díky tomu lze s výhodou jazyk XML využít jako formát nezávislý na platformě a jednom jediném programu. Defaultně využívá kódování UTF, tudíž není problém i s jiným jazykem, než je angličtina.

Ukázka dokumentu v jazyku XML

```

<?xml version="1.0" encoding="UTF-8"?>
<nastaveni jméno="program" nazev="krusader" verze="1.9">
  <titulek>Konfigurace programu</titulek>
  <otevřenéPanely>Jedna</otevřenéPanely>

```

```

<otevřenéPanely>Dva</otevřenéPanely>
<otevřenéPanely>Tři</otevřenéPanely>
<umístění>
  <souradniceX>122</souradniceX>
  <souradniceY>355</souradniceY>
  <šířka>800</šířka>
  <výška>800</výška>
</umístění>
</nastaveni>

```

Pro práci s XML je v nástroji pro sledování změn webu použita knihovna dom4j. Tato knihovna umožňuje pracovat mimo XML také s XPath a XSLT na platformě Java, používající rozhraní Java Collection Framework s plnou podporou pro DOM, SAX a JAXP.

Další informace o XML jsou k nalezení na [18].

## 2.6 Apache Ant

Apache Ant [2] je nástroj, sloužící k sestavování Java aplikací principem podobným nástroji make, ale je celý napsán v Javě, tudíž je nezávislý na platformě. Každá operace je implementována za pomoci standardních knihoven jazyka Java, bez využití nativních programů hostitelského operačního systému. Sestavovací soubory jsou ve formátu XML. [1]

Sestavovací soubor obsahuje:

- projekt (project)
- cíle (target)
- úlohy (task)
- závislosti

Ukázka souboru build.xml

```

<project>
  <target name="clean">
    <delete dir="build"/>
    <delete dir="jar"/>
  </target>

  <target name="compile">
    <mkdir dir="build"/>
    <mkdir dir="build/org"/>
    <copy todir="build/org">
      <fileset dir="src/org"/>
    </copy>
    <javac srcdir="src" destdir="build"/>

    <mkdir dir="jar/img"/>

```

```

<copy todir="jar/img">
  <fileset dir="src/img"/>
</copy>

<mkdir dir="jar"/>

<jar destfile="jar/ija2007.jar">
  <fileset dir="build"/>
  <manifest>
    <attribute name="Main-Class" value="ija2007.ija2007"/>
  </manifest>
</jar>
</target>

<target name="run">
  <java jar="jar/ija2007.jar" fork="true"/>
</target>
</project>

```

Příkaz

- `ant compile` – přeloží zdrojové soubory a zároveň vytvoří jar soubor
- `ant clean` – smaže adresáře build a jar
- `ant run` – spustí jar soubor

## 2.7 Swing

Swing [17], [3] je grafickým uživatelským rozhraním Javy od verze 1.2, do té doby jím byl AWT. Obsahuje nepřeborné množství grafických komponent. Jsou to tlačítka, textová pole a podobné ovládací prvky. Všechny komponenty jsou umístěny v kontejnerech, kterými mohou být např. `JPanel`, `JFrame` či `JDialog`. Tyto kontejnery jsou pak zobrazeny v okně a na nich jsou námi umístěné komponenty. Swing využívá některé knihovny z `java.awt`.

## 2.8 SysTray for Java

Knihovna `SysTray for Java` [5] slouží k přístupu programu do system tray oblasti operačního systému. Ve verzi 2.4.1 jsou podporovány operační systémy Windows 2000, Windows XP a Linux s okenním manažerem KDE od verze 3.



# Kapitola 3

## Analýza a návrh

### 3.1 Požadavky

Požadavky

- Funkčnost
- Konfigurovatelnost
- Indikace změn
- Přenositelnost
- Rozšiřitelnost

#### 3.1.1 Funkčnost

Program musí v pravidelných intervalech kontrolovat internetové stránky a vyhodnocovat na nich změny, které zpracuje, a určitým způsobem o nich dá uživateli vědět.

#### 3.1.2 Konfigurovatelnost

Dalším požadavkem je konfigurovatelnost. Uživatel má možnost přizpůsobit si program pomocí velkého množství nastavení. Barva, velikost, styl a tloušťka rámečku, zvýraznění textu, odkazů a obrázků pro každou kontrolovanou stránku zvlášť. Dále existuje možnost konfigurace, zda se mají stahovat i obrázky a soubory css. Je také možno nastavit, od kolika procent změn na stránce se má stránka automaticky otevřít v prohlížeči, a zadání času znovunačtení je samozřejmostí. Některé položky pro všechny kontrolované stránky je možno změnit globálně z hlavní obrazovky, ať už jde o zakázání automatického otevírání změněné stránky v prohlížeči, vyskakování popup okna či zvukové indikace změn. V hlavním nastavení aplikace může uživatel zadat pracovní adresář a změnit další individuální nastavení, jako je barva popup okna apod.

#### 3.1.3 Indikace změn

Velký důraz je kladen na správnou a přehlednou indikaci změn. Program by měl být schopen správně vyhledat změny textu, odkazů a také obrázků. Odkaz bude považován za změněný pokud se změní cíl odkazu `href` nebo titulek `title`. U obrázku je to podobné s tím, že se navíc uvažuje změna rozměrů obrázku. Změna textu a odkazů je indikována zvýrazněním

předem definovanou barvou, velikostí písma a také tloušťkou a barvou rámečku, který samozřejmě nemusí být nastaven. Změna obrázků je indikována rámečkem předem definované barvy, tloušťky a typu kolem tohoto obrázku.

### 3.1.4 Přenositelnost

Program by měl být nezávislý na jedné jediné architektuře a operačním systému. Ne vždy se podaří naprogramovat a odladit aplikaci, která je stoprocentně přenositelná a funkční na různých operačních systémech, ale u Nástroje pro sledování změn webu bude stačit, pokud nebude ovlivněna základní funkčnost sledování a indikace změn. Například to, že aplikace automaticky nedokáže otevřít internetový prohlížeč se změněnou stránkou, neovlivní její správnou funkčnost.

### 3.1.5 Rozšiřitelnost

Program by měl být jednoduše rozšiřitelný, například o možnost kontrolovat internetové stránky pomocí HTTPS, což je komunikace pomocí protokolu HTTP, ale data jsou šifrována pomocí SSL (Secure Sockets Layer) nebo TLS (Transport Layer Security). HTTPS standardně komunikuje pod TCP portem číslo 443. Dále by mohla být přidána možnost kontrolovat stránky, kde je nutná autentizace pomocí uživatelského jména a hesla, přidání podpory protokolu FTP či šifrované verze SFTP.

## 3.2 Volba implementačního prostředí

Dalším důležitým bodem analýzy je volba vhodného implementačního prostředí. V úvahu připadaly dvě možnosti: C++ s knihovnou wxwidgets pro grafické uživatelské rozhraní nebo Java s knihovnou Swing, která je dostupná standardně od verze Javy 1.2.

Zvolil jsem druhou možnost, tedy provést implementaci aplikace pomocí jazyka Java s grafickou knihovnou Swing, protože s ní mám větší zkušenosti a jedním z požadavků, jenž jsem si vytýčil, je přenositelnost a nezávislost na platformě. Jako vývojové prostředí jsem zvolil Netbeans IDE 6.0 pro jeho pokročilé funkce při návrhu grafického uživatelského prostředí a také pro velmi dobrou správu projektů.

Program s GUI je řízen událostmi:

- obecnější pojem pro označení asynchronního řízení programu
- základní princip tvorby grafického uživatelského rozhraní
- tok programu je řízen událostmi
- událostní aplikace by měly být programovány jako vícevláknové

## 3.3 Stažení stránky

### 3.3.1 Volba http protokolu

Nejdříve bylo nutno zvolit verzi http protokolu, verze 1.1 přináší pro uživatele tyto významné funkce:

- přenos více dokumentů při jednom spojení
- přenos dynamicky vytvářených dokumentů v jednom spojení
- podpora virtuálních serverů na jedné síťové adrese
- automatický výběr typu dokumentu, jazyka a kódování
- přenos vybrané části dokumentů

Tato verze je rozšířením HTTP verze 1.0. Každý server a klient zvládající HTTP verze 1.1 je schopen komunikovat i s verzí HTTP 1.0, proto jsem se také rozhodl pro tuto starší verzi, jelikož bych žádnou z nových funkcí z verze 1.1 nevyužil a zbytečně bych se musel zaobírat implementací zpracování složitějších http hlaviček oproti verzi 1.0.

### 3.3.2 Návrh adresářové struktury

Adresářová struktura se odvíjí od umístění pracovního adresáře, který je po prvním spuštění aplikace v pracovním adresáři uživatele. Toto umístění může uživatel změnit v nastavení aplikace. Poté struktura adresářů vypadá takto `pracovni adresar/data/adresa stranky/podslozka`, kdy podsložkou může být adresář `akt`, který obsahuje aktuální staženou stránku, s níž se porovnávají znovunačtené stránky; dále jí může být adresář `temp`, do něž se ukládají stránky na porovnávání. Pokud dojde ke změně, bude obsah adresáře `temp` přesunut do `akt` a zároveň budou změny zvýrazněny v html souboru. Takto zvýrazněný html soubor bude uložen do podložky `zmeny`. Jestliže bude v nastavení stránky povoleno stahování css souborů a obrázků, budou tyto css soubory a obrázky staženy a uloženy také do podadresáře `zmeny`.

## 3.4 Znovunačtení za časový interval

Stránky je nutno znovunačítat po zadaném časovém intervalu.

Zde se nabízí několik možností řešení:

- pomocí aktivní čekací smyčky
- pomocí vláken
- pomocí timeru (časovače) a vláken

### 3.4.1 Aktivní čekací smyčka

Aktivní čekací smyčku by bylo možno implementovat jako cyklus např. `while`, `do-while`, `for`, kde by se kontroloval seznam stránek. Pokud by měla být některá stránka již obnovena, začala by se stahovat a poté by pokračovala kontrola dalších stránek. Toto řešení je snad to nejhorší, které by šlo použít. Dále by šla použít varianta, kde by se kontrolovaná stránka při obnově stahovala v samostatném vlákně – tedy více stránek by se kontrolovalo paralelně. Ale ani toto řešení není kvůli aktivnímu čekání příliš vhodné.

### 3.4.2 Vlákna

Řešení pomocí vláken se mi ze začátku jevilo jako nejlepší. Každá kontrolovaná stránka by běžela jako samostatné vlákno, které by se vždy uspalo na čas, za nějž se má stránka znovunačíst. Nebylo by skoro žádné zpoždění a stránky by se opravdu kontrolovaly za přesně zadaný čas. Ale problém vidím v tom, kdyby se kontrolovalo velké množství stránek – ty by pak zbytečně zabíraly místo v operační paměti.

### 3.4.3 Časovač a vlákna

Další možností je použití časovače spolu s vlákny. Kdy časovač jednou za předem definovaný časový interval zkontroluje seznam stránek, a ty, které mají být již znovunačteny, jsou staženy – každá v samostatném vlákne, které se po stažení stránky ukončí. Toto řešení může mít menší prodlevy ve stahování stránek než řešení pouze za pomoci vláken, ale je méně náročné na operační paměť, a při dobře zvolené periodě časovače se mi jeví jako nejlepší řešení, které jsem si proto také zvolil.

## 3.5 Vyhledání změn

Rozhodoval jsem se mezi dvěma možnostmi, a to buď si naprogramovat vlastní metodu, která porovnává dva soubory a nalezené změny vrací jako seznam změn v datové struktuře, jež je dobře zpracovatelná v další části programu; nebo použít knihovnu `java-diff`, porovnávající dva objekty či kolekce a vracející seznam objektů, které obsahují seznam změn. Zvolil jsem druhé řešení, jelikož datová struktura obsahující seznam změn je velmi dobře zpracovatelná a není potřeba programovat vlastní řešení, což by bylo časově náročně.

## 3.6 Možnosti zvýraznění

Možnosti zvýraznění jsou dvě. První možnost je uvádět styl do samostatného css souboru, ale zde je potřeba ještě v souboru html označit stylované elementy a přiřadit class id. Druhou možností je uvádět styl přímo do html souboru, kdy změněné atributy jsou mezi ostylovanými tagy např. `<span style="color: rgb( 255 , 0 , 255 )">` a `</span>`. První možnost je složitější a nevidím v ní ani žádnou výhodu, snad jen jednoduchou změnu vzhledu již zvýrazněného souboru, a proto jsem zvolil druhé řešení – stylování přímo do souboru html.

## 3.7 Ukládání konfigurace

U navrhované aplikace připadá v úvahu také ukládání uživatelských nastavení a seznamu stránek, aby uživatel nemusel znovu vše nastavovat po opětovném spuštění aplikace a také aby nemusel stále zadávat kontrolované stránky. Také by bylo vhodné oddělit ukládání konfigurace programu od seznamu stránek. Aby měl program vždy po spuštění možnost automaticky načíst konfiguraci a poslední použitý seznam stránek, tak se tato nastavení uloží do domovského adresáře uživatele, kde je program nalezne hned po spuštění. Pokud se tak nestane, budou použity defaultní hodnoty. Samozřejmě bude mít uživatel možnost za běhu programu ukládat a načítat různé seznamy stránek z libovolného umístění.

Pro ukládání souborů s konfigurací bylo opět několik možností:

- textový soubor
- XML soubor

### 3.7.1 Textový soubor

Tato varianta vyniká svou jednoduchostí, kdy jsou položky nastavení zapisovány do souboru postupně a ve stejném pořadí jsou také čteny při spuštění aplikace. Nevýhodu vidím v tom, že takový konfigurační soubor je nepřehledný.

### 3.7.2 XML soubor

Ukládat nastavení do souboru ve formátu v XML je přehlednější řešení než do textového souboru vlastního formátu. Pro práci s XML je pro jazyk Java k dispozici knihovna `dom4j`, kterou jsem se rozhodl použít pro její rozsáhlé možnosti.

## 3.8 System Tray

Dalším bodem návrhu byla možnost minimalizace aplikace do oblasti system tray, kde by bylo přístupné také ovládací menu. Opět bylo několik řešení:

- Java verze 1.6
- knihovna `jdjc`
- knihovna `Systray for Java`

### 3.8.1 Java 1.6

První varianta pomocí standardních balíčků `java.awt.SystemTray` a `java.awt.TrayIcon` [6] vypadá na první pohled nejlépe, ale jelikož by pro správnou činnost aplikace byla nutná pouze Java verze 1.6 tak sem od ní upustil, aby bylo možno nástroj pro sledování změn webu provozovat i pod Javou verze 1.5. Další nevýhodou této možnosti je, že pod linuxem vyžaduje některé další knihovny z Gnome pro správnou funkčnost v linuxu.

### 3.8.2 Knihovna `jdjc`

Druhou možností byla knihovna `jdjc` [4], která v sobě integrovala i další funkčnost, např. internetový prohlížeč, který by šel zajisté také využít v mnou navrhované aplikaci. Ale bohužel se naskytly nečekané komplikace s kompatibilitou na jiných operačních systémech tak jsem byl nucen přistoupit k třetí variantě.

### 3.8.3 Knihovna `Systray for Java`

Tato knihovna podporuje aplikaci přístup do oblasti system tray v operačních systémech Windows 2000, Windows XP a v linuxu s KDE verze 3. Funkčnost a kompatibilita jsou na velmi dobré úrovni, kdy je pouze potřeba v operačním systému uživatele do adresáře `vas_adresar_s_javou/lib/i386` nakopírovat knihovnu `systray4j.dll` pro operační systémy Windows a `libsystray4j.so` pro operační systémy linux. Nakonec jsem se rozhodl pro tuto variantu, protože nevyžaduje žádné speciální knihovny závislé na operačním systému a je tudíž velmi dobře toto řešení přenositelné.

# Kapitola 4

## Implementace a testy

V této kapitole bude popsána implementace jednotlivých částí Nástroje pro sledování změn webu. Začnu popisem jednotlivých balíků tříd a budu pokračovat popisem implementace pro každou třídu zvlášť.

### 4.1 Balík main

Balík obsahuje pouze jednu třídu, a to `Main`. Tato třída obsahuje jednu jedinou metodu `main()`, která slouží ke spuštění celé aplikace.

### 4.2 Balík zmeny

Tento balík obsahuje všechny důležité třídy nezbytné pro správnou činnost aplikace, od třídy sloužící ke stáhnutí stránky, přes třídu, která vyhodnocuje změny, až po třídu, která automaticky otevře prohlížeč se stránkou, kde jsou změny již zvýrazněny.

- CSS
- Databaze
- Download
- Img
- Konfigurace
- Porovnani
- Prohlizec
- Soubory
- VlasknoStahovani

### 4.2.1 Třída CSS

Třída `CSS` obsahuje metodu pro nalezení cesty k umístění souboru `css` v souboru `html`, metodu pro stažení a uložení tohoto souboru na pevný disk, metodu pro přepsání cesty v `html` souboru ke staženému `css` souboru na našem pevném disku. Hlavní metoda této třídy je metoda `StahniCSS()`, ve které jsou volány ostatní metody. Nejříve je nutno nalézt cestu k `css` souboru – k tomu slouží metoda `jeCSS()`, která vrací `true`, pokud kontrolovaný tag obsahuje cestu k požadovanému souboru. Poté je zavolána metoda `ziskejHref()`, které je jako formální parametr předán tag s cestou k souboru `css`. Tato metoda vrací absolutní cestu k požadovanému souboru, a pokud je v `html` souboru relativní cesta k `css` souboru, je tato cesta změněná na absolutní. Pomocí metody `souborCSS()`, kam jako formální parametr dosadíme zjištěnou absolutní cestu, získáme název souboru `nazev.css`. Poté metoda `ulozSoubor()`, stáhne a uloží požadovaný `css` soubor. Dále je zavolána metoda `nahradOdkaz()` sloužící k úpravě cesty k právě staženému a uloženému souboru s kaskádovými styly.

### 4.2.2 Třída Database

Třída `Database` obsahuje metody pro práci s databází stránek. Tato databáze je tvořena několika statickými proměnnými datového typu `ArrayList`. Metody v této třídě slouží k přidání kontrolované stránky do databáze `pridejDatabase()`, úpravě stávající stránky `upravPolozku()` a smazání stránky `smazatPolozku()`. Dále tu jsou metody pro získání těchto nastavení samostatně např. `vratAdresu()`, `stahnoutCSS()`, `stahnoutImg()`, `vratZvyrazText()`, `vratZvyrazHref()`, `vratRefresh()` atd. Pro každou stránku samostatně existuje možnost různých nastavení uložených v databázi.

Příklad definice jednotlivých položek databáze, do které se pak přidávají jednotlivé kontrolované stránky včetně nastavení.

```
private ArrayList<String> databaseStranek= new ArrayList<String>();
private ArrayList<Boolean> stahnoutCSS= new ArrayList<Boolean>();
private ArrayList<Boolean> stahnoutImg= new ArrayList<Boolean>();
private ArrayList<String> zvyrazneniText= new ArrayList<String>();
private ArrayList<String> zvyrazneniImg= new ArrayList<String>();
private ArrayList<String> zvyrazneniHref= new ArrayList<String>();
private ArrayList<Integer> refreshTime= new ArrayList<Integer>();
private ArrayList<Long> nextRefresh= new ArrayList<Long>();
private ArrayList<Boolean> isRefresh= new ArrayList<Boolean>();
private ArrayList<Integer> procentoZmen = new ArrayList<Integer>();
private ArrayList<Boolean> zmenaDoTabulky = new ArrayList<Boolean>();
private ArrayList<Float> pocetZmen = new ArrayList<Float>();
private ArrayList<Boolean> oteviraniProhlizece = new ArrayList<Boolean>();
```

Pod stejným indexem v každé statické proměnné typu `ArrayList` jsou související nastavení a adresa stránky.

Dále třída obsahuje metodu na uložení databáze seznamu stránek do souboru ve formátu XML `ulozDoXML()` a také pro její načtení `nactiXML()`.

### 4.2.3 Třída Download

Třída `Download` tvoří základ balíku `zmeny`; její hlavní metoda `connection()` pošle http dotaz na vzdálený server, pomocná metoda `strankaNactena()` vrací `true`, pokud přijatá stránka byla přijata v pořádku, v opačném případě vrací `false`, což znamená, že stránka byla přesměrována. Poté dochází k analýze hlavičky http odpovědi, získání nové adresy a je stažena nová stránka. Po vrácení `true` je hlavička odstraněna a internetová stránka je pomocí metody `ulozStranku()` uložena do souboru.

### 4.2.4 Třída Img

Tato třída obsahuje metody pro nalezení cesty k umístění obrázku `jpg/png/gif` v html souboru, další metody slouží k jeho stažení a uložení na pevný disk. Je zde také obsažena metoda pro přepsání cesty ke staženému souboru na našem pevném disku. Hlavní metoda třídy je `stahniImg()`, ve které jsou volány ostatní metody. Nejdříve je nutno nalézt cestu k souboru `jpg/png/gif`; k tomu slouží metoda `jeImg()`, která vrací `true`, pokud kontrolovaný tag obsahuje cestu k obrázku. Dále je zavolána metoda `ziskejHref()`, které se jako formální parametr předá tag s cestou k souboru obrázku. Tato metoda vrací absolutní cestu k požadovanému souboru; pokud je v html souboru relativní cesta k obrázku, je tato cesta upravena na absolutní. Pomocí metody `souborImg()`, kam jako formální parametr dosadíme zjištěnou absolutní cestu, získáme název souboru `nazev.jpg/png/gif`. Poté si ověříme, zdali tento soubor není již stažen a uložen ve stejném umístění pod právě zjištěným názvem. Pokud není, je zavolána metoda `ulozSoubor()`, která stáhne a uloží obrázek ze získané adresy. Dále je ještě zavolána metoda `nahradOdkaz()`, jež upraví v html souboru cestu k právě staženému a uloženému obrázku.

### 4.2.5 Třída Konfigurace

Třída `Konfigurace` obsahuje statické proměnné s nastavením aplikace. Jsou v ní implementovány metody sloužící k získání i změně nastavení. Ve statických proměnných jsou uložena různá nastavení např. cesta k pracovnímu adresáři; perioda časovače pro kontrolu internetových stránek; zda se má automaticky otevírat prohlížeč; barva popup okna; jestli je zapnuto zvukové upozornění; kolikrát je třeba poklikt do tabulky, aby se otevřela změněná internetová stránka v prohlížeči; atd.

### 4.2.6 Třída Porovnani

Jednou z nejdůležitějších tříd balíku `zmeny` je třída `Porovnani`, obsahující metody pro nalezení změn a jejich zvýraznění. Její hlavní metoda `porovnej()` volá nejprve pomocnou metodu `otevri()`, která do proměnné typu `String` ze souboru načte internetovou stránku uloženou v podadresáři `akt`. Jestliže není s čím porovnávat (podadresář `temp` je prázdný), metoda `porovnej()` je ukončena. V opačném případě je zavolána metoda `otevri()` ještě jednou a i z tohoto umístění načte stránku do další proměnné typu `String`. Dále je zavolána metoda `tokenize()`, které je předána načtená internetová stránka. Tato metoda stránku rozdělí na jednotlivé tokeny, dle tohoto pravidla: tag obrázku `<img...>`, tag odkazu `<a href=...>`, ostatní tagy a dále jednotlivá slova textu oddělená bílými znaky nebo tagy; výsledek vrací v proměnné typu `ArrayList`. Stejným způsobem je zpracována i druhá načtená stránka v proměnné. Pokud již máme obě stránky rozděleny v proměnných typu `ArrayList`, je zavolána metoda `zpracujZmeny()`, která nalezne pomocí knihovny `java-diff`



změny. Pokud nejsou nalezeny žádné změny, je metoda `porovnej()` ukončena; v opačném případě jsou nalezené změny pomocí kaskádových stylů zvýrazněny. Takto upravená stránka je dále zpracována pomocí metod tříd `CSS` a `Img`, pokud je ovšem nastaveno, že se mají obrázky a kaskádové styly stahovat. Poté je stránka uložena do podadresáře `zmeny` a uživateli je předem nastaveným způsobem dáno na vědomí to, že došlo ke změně (otevření prohlížeče, popup okno, zvuková signalizace).

#### 4.2.7 Třída `Prohlizec`

Tato třída se stará o správné otevírání internetového prohlížeče v závislosti na operačním systému. Hlavní metodou třídy je metoda `otevriStranku()`, které je předán parametr obsahující adresu stránky, jež má být otevřena. V této metodě je volána pomocná metoda `jeWindows()`, která vrací `true`, pokud je aplikace spuštěna v operačním systému Windows, nebo `false`, jestliže se jedná o operační systém linux. Podle výsledku této metody je otevřen buď defaultní internetový prohlížeč operačního systému Windows, nebo Firefox v linuxu. Je zde přítomna i metoda pro zobrazení nápovědy aplikace. Princip jejího zobrazení je stejný jako výše popsany s tím rozdílem, že se metodě nepředává žádný parametr – umístění souboru s nápovědou je předem známé a neměnné.

#### 4.2.8 Třída `Soubory`

V této třídě jsou implementovány metody pro manipulaci se soubory. Pokud se stránka změnila, je přesunuta ze složky `temp` do složky `akt`. Dále je zde implementována metoda, která dostane cestu k adresáři jako parametr a tento adresář včetně všech podsložek a jejich obsahu smaže. Je zde přítomna také metoda, jež dostane cestu k souboru jako parametr a smaže pouze tento jeden soubor.

#### 4.2.9 Třída `VlaknoStahovani`

Třída `VlaknoStahovani` vznikla děděním od třídy `Thread`. Instance této třídy jsou vytvářeny ve třídě `MenuView`. Jsou vytvořeny pouze objekty od těch stránek, které mají být momentálně kontrolovány, a to je provedeno tak, že se v cyklu projde databáze kontrolovaných stránek, a pokud má být stránka znovunačtena, je vytvořen objekt této třídy. Jakmile je databáze zkontrolována celá, jsou všechna vlákna současně spuštěna a další činnost probíhá paralelně. Aby se předešlo problémům, kdy se stránka načítá neúměrně dlouho a měla by být již znovunačtena, je hned na začátku kontroly u kontrolované položky databáze nastaven atribut `isRefresh` na hodnotu `true`, který je po skončení kontroly opět nastaven na `false`.

K činnostem hlavní metody `run()` třídy `VlaknoStahovani` patří: stažení kontrolované stránky, vyhledání a zvýraznění změn a poté informování uživatele o tom, že stránka byla změněna. Všechny tyto činnosti, jak vyplývá z textu uvedeného výše, probíhají pro každou kontrolovanou stránku současně a nezávisle na sobě. Stránka je stažena pomocí metody `connection()` třídy `Download`. Pokud je s čím porovnávat, je zavolána metoda `porovnej()` třídy `Porovnani`, kde se také řeší zvýraznění změn a dání uživateli na vědomí změnu internetové stránky.

Tato třída dále obsahuje metodu `destroy()` sloužící k okamžitému ukončení kontroly stránky.

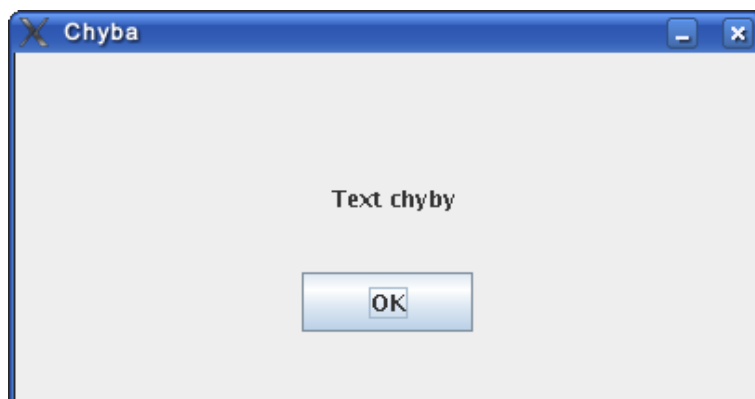
## 4.3 Balík gui

V balíku gui jsou obsaženy tyto třídy:

- ChybaApp
- KonfiguraceApp
- MenuAboutBox
- MenuView
- PopupOkno
- PridejApp
- SysTray

### 4.3.1 Třída ChybaApp

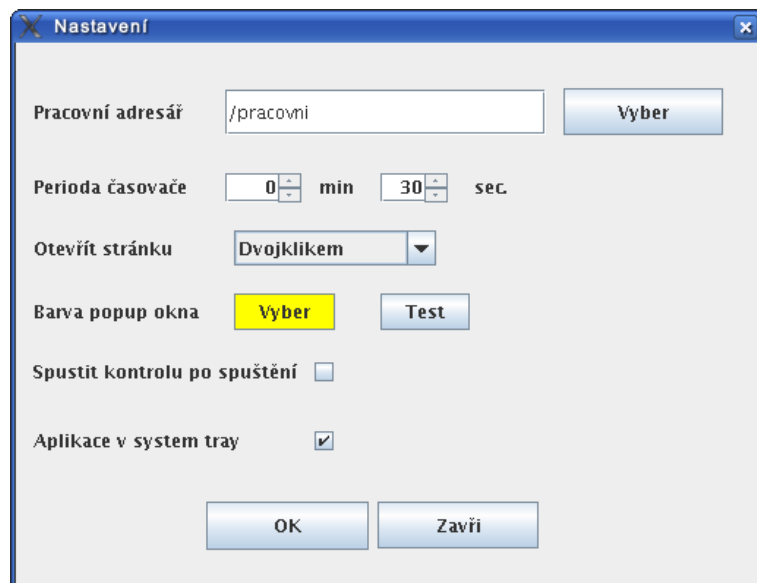
Třída `ChybaApp` je třída balíku `gui`, zajišující zobrazení dialogového okna s popisem nastalé chyby. Třída obsahuje pouze konstruktor `ChybaApp()`, kterému je předán jako parametr pouze text chyby.



Obrázek 4.1: dialogové okno chyba

### 4.3.2 Třída KonfiguraceApp

Tato třída je grafickým uživatelským rozhraním pro třídu `Konfigurace`. Konstruktor `KonfiguraceApp()` vytvoří okno s nabídkami nastavení, pomocí nichž lze měnit nastavení aplikace. Po stisku tlačítka `OK` jsou všechna nastavení pomocí metody `ulozKonf()` zapsána do statických proměnných třídy `Konfigurace`.



Obrázek 4.2: Okno s konfigurací aplikace

### 4.3.3 Třída MenuAboutBox

Tato třída zobrazuje dialogové okno s verzí programu. Obsahuje pouze konstruktor `MenuAboutBox()`, který vytvoří a inicializuje dialogové okno.



Obrázek 4.3: O aplikaci...

### 4.3.4 Třída MenuView

Třída `MenuView` je jednou z nejdůležitějších tříd grafického uživatelského rozhraní. Její konstruktor `MenuView()` vytvoří hlavní okno aplikace, které obsahuje hlavní menu, tabulku, v níž jsou zobrazovány kontrolované stránky včetně nejdůležitějších údajů, a také tlačítka pro přidání, editaci a smazání kontrolované stránky. Dále existuje možnost použít tlačítka pro globální nastavení zvukové výstrahy, vyskočení popup okna, automatické otevření internetového prohlížeče a okamžité pozastavení kontroly stránek.

Pokud bude uživatel chtít přidat či změnit novou stránku a stiskne tlačítko *Přidej* respektive *Edituj* bude vytvořena instance třídy `PridejApp`, kde uživatel provádí přidání resp. změnu kontrolované internetové stránky. Takto nově přidaná položka do databáze bude zobrazena v hlavním okně v tabulce.

V hlavním menu v položce Soubor má uživatel možnost ukončit program volbou Konec, které je přiřazena klávesová zkratka *CTRL-Q*. Další možností tohoto podmenu jsou položky Ulož (klávesová zkratka *CTRL-S*) respektive Otevři (klávesová zkratka *CTRL-O*), které slouží k uložení seznamu kontrolovaných stránek do souboru ve formátu XML respektive k jejímu znovunačtení z takto vytvořeného souboru.

Další položkou hlavního menu je položka Nastavení, která umožňuje konfiguraci programu. Po výběru této položky je vytvořeno nové okno třídy `KonfiguraceApp`, kde je možno provést různá nastavení aplikace.

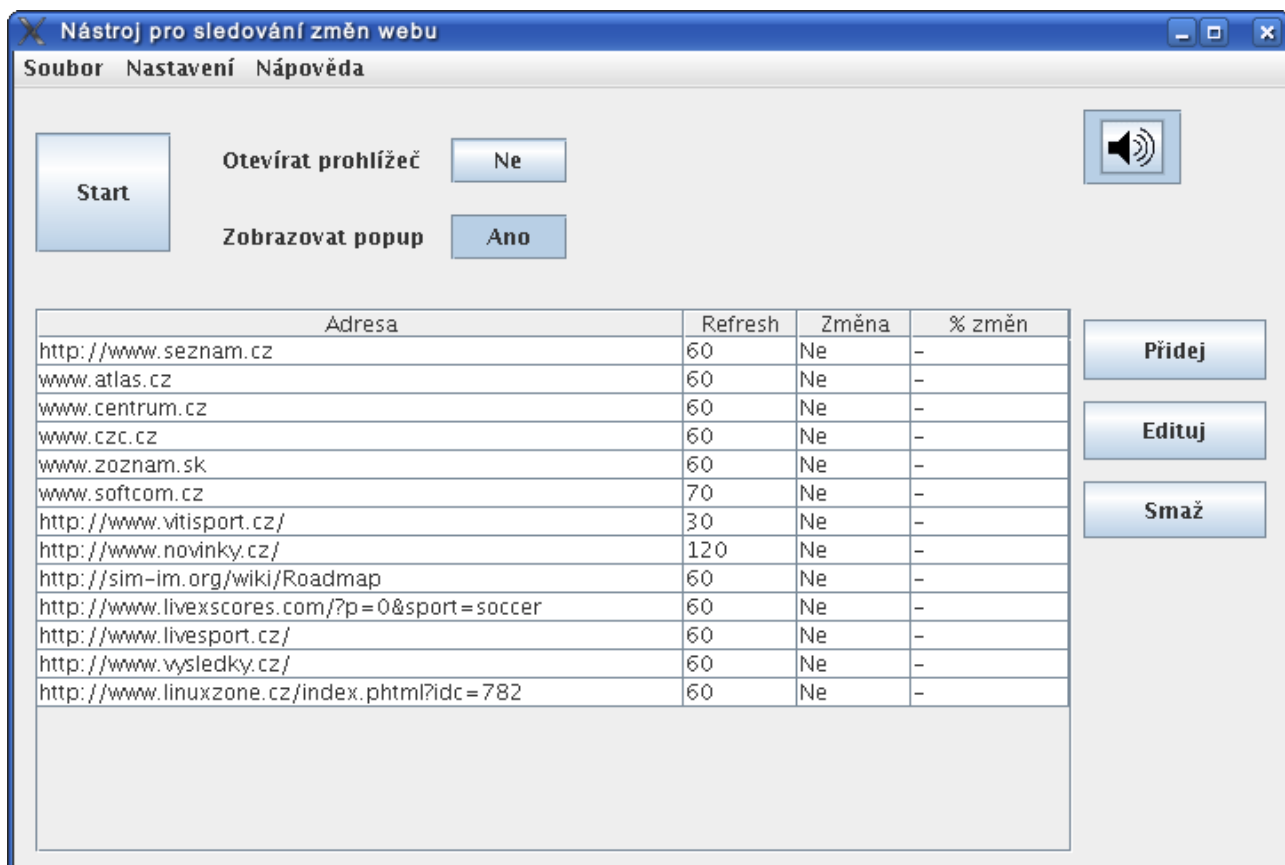
Poslední položkou menu je Nápověda (klávesová zkratka *F1*), která zahrnuje dvě možnosti, a to zobrazení nápovědy, pokud uživatel zvolí možnost Nápověda, a zobrazení informací o programu výběrem možnosti O aplikaci. Po volbě první možnosti je otevřena nápověda programu ve formátu html v internetovém prohlížeči. Pokud je zvolena druhá možnost, je vytvořeno dialogové okno třídy `MenuAboutBox` a jsou zobrazeny informace o aplikaci.

Tlačítko *reproduktoru* slouží k zapnutí/vypnutí zvukového upozornění pomocí pc speakeru na změnu kontrolované internetové stránky. Funkčnost takového upozornění je tedy podmíněna přítomností pc speakeru v počítači.

Použití tlačítka *možnosti zobrazovat popup okno* globálně povolí/zakáže vyskakování popup oken při změně některé z kontrolovaných stránek.

Volba otevírat prohlížeč ovlivňuje na globální úrovni zákaz automatického otevírání internetového prohlížeče. Pokud je tato volba aktivní, u změněných stránek se za žádných okolností internetový prohlížeč automaticky neotevře.

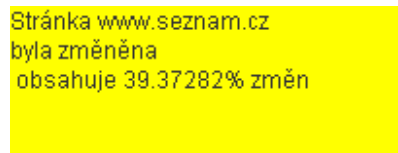
Tlačítko *Start* slouží k započetí kontroly internetových stránek. Pokud jsou již stránky kontrolovány, má toto tlačítko popisek *Zastavit*, pak tedy stisk tohoto tlačítka okamžitě pozastaví probíhající kontrolu stránek.



Obrázek 4.4: Hlavní okno aplikace

#### 4.3.5 Třída PopupOkno

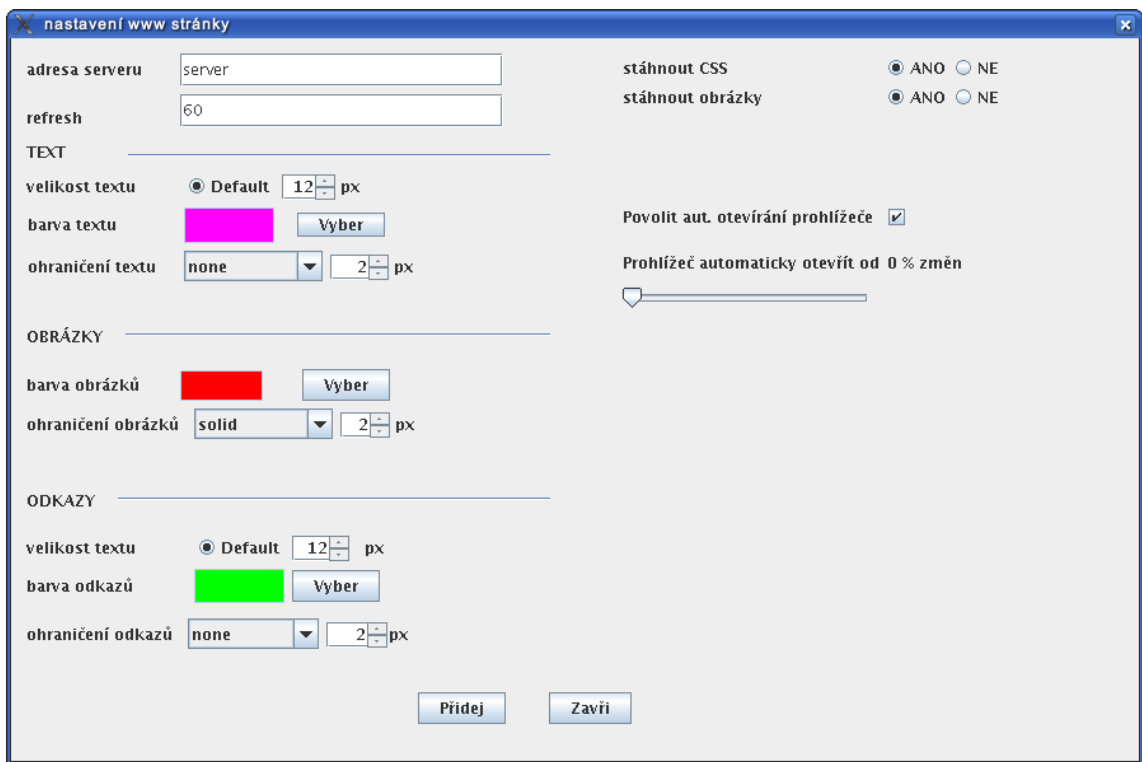
Třída `PopupOkno` slouží k zobrazení vyskakovacího popup okna v pravém dolním rohu. Toto okno obsahuje adresu internetové stránky, údaj, zdali byla tato stránka změněna, a procento všech změn provedených na stránce od poslední změny stránky. Počet zároveň zobrazených popup oken závisí na rozlišení obrazovky, protože vyskakovací popup okna se skládají nad sebe do max. výšky rozlišení. Pokud již nejde zobrazit popup okno nad jiné, začne se zobrazovat opět odzadla. Popup okno zmizí automaticky za 5 sekund od svého zobrazení díky vnořené třídě `autoZmiz` a její reakci na událost `actionPerformed()`, kterou spustí `Timer` z balíku `javax.swing.Timer` za 5 sekund od vytvoření popup okna, ale pokud uživatel klikne pravým tlačítkem na plochu popup okna, popup okno okamžitě zmizí. Další volbou je kliknutí levým tlačítkem na plochu popup okna. Pokud tak uživatel učiní, je změněná internetová stránka, zobrazená v popup okně, otevřena v internetovém prohlížeči se zvýrazněnými změnami.



Obrázek 4.5: Vyskakovací popup okno

### 4.3.6 Třída PridejApp

Třída `PridejApp` poskytuje grafické uživatelské rozhraní pro přidání nové stránky a jejího nastavení do databáze. Tato třída je použita i pro editaci stávající kontrolované internetové stránky, která v databázi již existuje. Je nutné její nastavení načíst a zobrazit v editačním okně. Pro tento účel jsou implementovány metody `editaceText()`, `editaceHref()` a `editaceImg()` pro načtení a zobrazení nastavení zvýraznění textu, odkazů a obrázků v editačním okně. Poté, co uživatel vyplní nebo změní všechny potřebné údaje a nastavení, a po stisku tlačítka *OK*, je buď stránka uložena do databáze u nově přidávané stránky, nebo je v databázi změněna, pokud se jedná o editaci. O vše se starají metody `zpracujTagText`, která zpracuje a uloží nastavení zvýraznění textu, metoda `zpracujTagHref()`, ukládající nastavení zvýraznění odkazů, a metoda `zpracujTagImg()`, jež má na starosti zpracování nastavení zvýraznění obrázků. Při stisku tlačítka *Zavři* se okno uzavře bez uložení stránky či její editace. Pokud uživatel nezadá kterýkoli z povinných údajů a bude chtít stránku uložit, bude na takto vzniklou skutečnost upozorněn chybovým hlášením pomocí třídy `ChybaApp`.

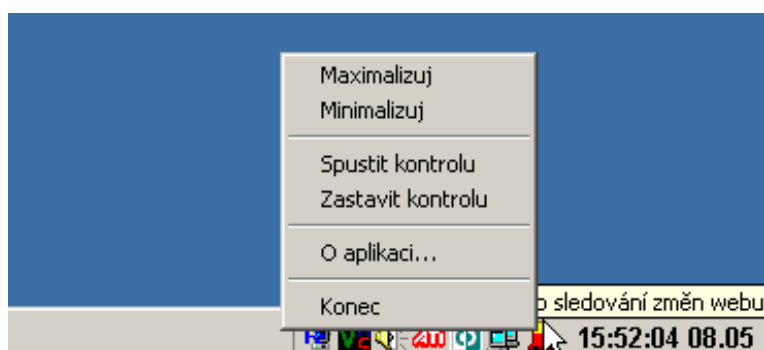


Obrázek 4.6: Přidání či editace položky

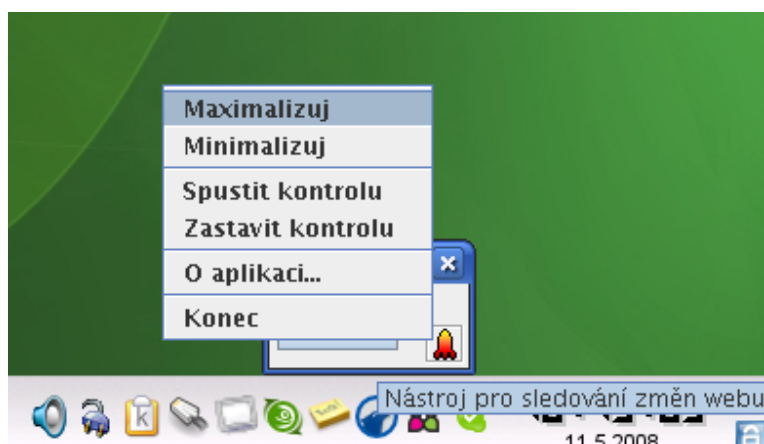
### 4.3.7 Třída SysTray

Tato třída slouží k umístění programu do oblasti system tray. Využívá knihovnu SysTray for Java. Pro její správnou činnost je nutno umístit do adresáře `adresarJava/lib/i386` soubor `libsystray4j.so` pro operační systém linux nebo soubor `systray4j.dll` pro Windows. Obě tyto knihovny jsou součástí archivu s aplikací. Dalším omezením je funkčnost pouze ve Windows 2000, Windows XP a v linuxu s KDE verze 3. Sami tvůrci v readme ke knihovně SysTray for Java uvádí, že na některých operačních systémech linux je aplikace využívající tuto knihovnu bohužel nestabilní. Pokud není požadovaná knihovna umístěna ve správném adresáři s Javou, je nad oblastí system tray zobrazeno okno, obsahující stejné menu, jaké by bylo v oblasti system tray.

Metoda `vytvorMenu()` vytvoří menu, které se zobrazí po kliknutí pravým tlačítkem myši na ikonu v oblasti system tray. Dále je zde metoda `menuItemSelected`, jež dle výběru z menu, které jsme vytvořili pomocí zmíněné metody `vytvorMenu()`, vyvolá odpovídající činnost. Dvojklik na ikonu v system tray způsobí minimalizaci či maximalizaci hlavního okna aplikace dle aktuálního stavu.



Obrázek 4.7: Menu aplikace v system tray



Obrázek 4.8: Menu aplikace pokud není knihovna pro system tray nalezena

## 4.4 Balík xml

V balíku xml jsou obsaženy tyto třídy:

- UlozXML
- NactiXML

### 4.4.1 Třída UlozXML

Třída UlozXML disponuje metodami sloužícími k uložení seznamu stránek či uložení konfigurace do souboru ve formátu XML s využitím knihovny dom4j. Metoda `ulozSeznamStranek()` má na starosti ukládání seznamu stránek, kdy tato metoda je volána v cyklu tolikrát, kolik položek má seznam stránek, a jako parametr je jí předána jedna položka tohoto seznamu, která se přidá do proměnné typu Document, jenž je touto metodou vrácen. Poté je obsah proměnné Document zapsán metodou `printDocument()` do souboru se seznamem stránek.

Ukázka XML souboru se dvěma uloženými stránkami:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <stranky>1</stranky>
  <wwwStranka0>
    <adresa>http://www.seznam.cz</adresa>
    <zvyrazneniTextu>&lt;span style="color: rgb( 255 , 0 , 255 )"&gt;</zvyrazneniTextu>
    <zvyrazneniImg>style="color: rgb( 255 , 0 , 0 );border: 2px solid"</zvyrazneniImg>
    <zvyrazneniHref>style="color: rgb( 0 , 255 , 0 )"</zvyrazneniHref>
    <refresh>60</refresh>
    <stahnoutCSS>true</stahnoutCSS>
    <stahnoutImg>true</stahnoutImg>
    <procentoZmen>34</procentoZmen>
  </wwwStranka0>
</root>
```

Dále se v této třídě nalézá metoda `ulozKonfigurace()` sloužící k uložení konfigurace, kdy je po zavolání této metody vrácena proměnná typu Document, která je již připravena pro zápis do souboru s konfigurací ve formátu XML pomocí `printDocument()`.

Ukázka XML souboru s uloženou konfigurací:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <pracovniAdresar>/home/tut/pracovni</pracovniAdresar>
  <periodaTimer>30000</periodaTimer>
  <pocetKliknuti>2</pocetKliknuti>
  <oteviraniProhlizece>true</oteviraniProhlizece>
  <barvaR>255</barvaR>
  <barvaG>255</barvaG>
  <barvaB>0</barvaB>
```



```
<autoStart>false</autoStart>
<systemTray>true</systemTray>
</root>
```

#### 4.4.2 Třída NactiXML

Třída `NactiXML` je třída sloužící k načtení seznamu stránek a načtení konfigurace aplikace ze souborů ve formátu XML. Opět je zde využito knihovny `dom4j`. Nejdříve je nutno načíst obsah XML souboru do proměnné typu `Document` – k tomu slouží metoda `parse`. Dále je při načítání seznamu stránek nutno zjistit počet stránek uložených v souboru – k tomu slouží metoda `pocetStranek()`, která tento počet vrátí. Poté je již volána metoda `nactiStranku()`, která vždy načte jednu položku databáze ze souboru, již přidá do databáze aplikace. Tato metoda je volána tolikrát, kolik uložených stránek soubor obsahuje.

### 4.5 Překlad a spuštění pomocí Apache Ant

Překlad probíhá za pomoci programu Apache Ant. Aplikace je přeložena příkazem `ant compile` do adresáře `build` a příkazem `ant run` je spuštěna. Je zde také možnost přeložené zdrojové soubory smazat pomocí příkazu `ant clean`. Všechny tyto příkazy je nutno zadávat z hlavního adresáře, ve kterém se soubor `build.xml` nachází.

### 4.6 Testy

Testy probíhaly na těchto operačních systémech a verzích Javy:

- Windows 2000, Java 1.6
- OpenSUSE 10.3, Java 1.5
- Windows XP, Java 1.6
- Windows 98

#### 4.6.1 Test č. 1

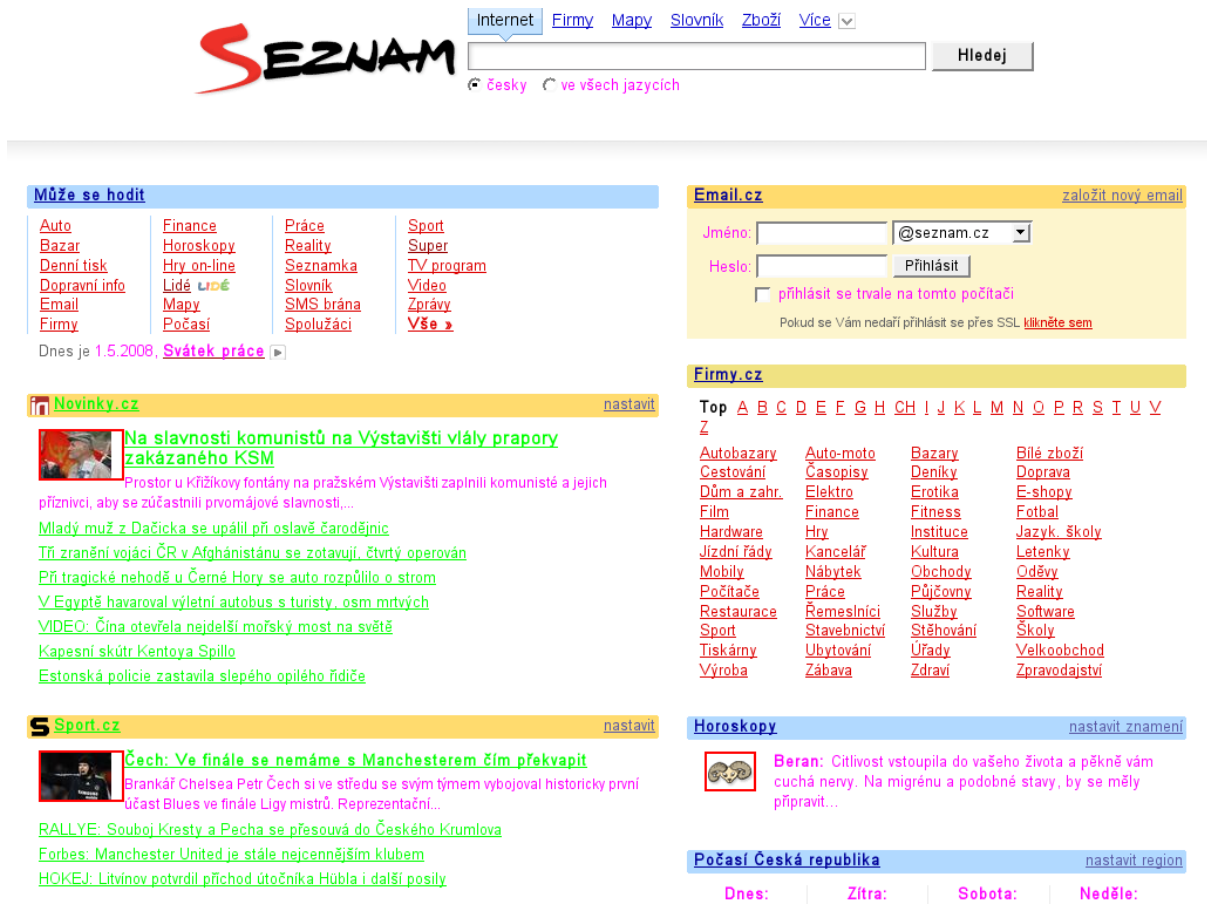
Kontrolovaná stránka: <http://www.seznam.cz>

Interval kontroly: 24 hodin (aby změny byly na první pohled patrné)

Výsledek:

Výsledek dopadl dle očekávání. Stránka se stráhla v pořádku a změny byly správně zvýrazněny – zelenou barvou odkazy, fialovou barvou text a červeným rámečkem obrázky.

Screenshot:



Obrázek 4.9: Zvýrazněné změny na stránce www.seznam.cz

#### 4.6.2 Test č. 2

Kontrolovaná stránka: <http://www.livescore.com>

Interval kontroly: libovolný

Výsledek:

Zadaná stránka je stažena, avšak obsahuje pouze odkaz na jinou stránku zadanou relativní adresou. Program bohužel nebyl navržen s ohledem na takové okolnosti a řeší pouze přesměrování v http hlavičce. Stránka proto nemůže být zkontrolována.

#### 4.6.3 Test č. 3

Kontrolovaná stránka: [www.vitisport.cz](http://www.vitisport.cz)

Interval kontroly: libovolný

Výsledek:

Stránka je v pořádku stažena, ale obsahuje prvky, které načítají obsah ze zadané URL adresy, až když se tento prvek má zobrazit. Proto se jeho obsah načítá až při zobrazení a

nemůže být porovnán na změny. Jsou zvýrazněny pouze prvky, které jsou obsaženy přímo na zadané www stránce.

Screenshot:

The screenshot displays the website www.vitissport.cz, which is a platform for sports betting and news. The page is divided into several main sections:

- Left Sidebar:** Contains navigation links for various sports and betting options, including 'Tenis Oficiální stránky', 'Roma', 'Berlin', 'Dresden', 'Ostrava', 'Rabat', 'Rijeka', 'LINKS', 'TenisInfo.cz', 'Sportka - výsledky losování', 'EXPEKT', 'SPORTINGBET', and 'UNIBET'. Each section includes a small image and a brief description of the service.
- Top Section:** Features a header for 'EUROFOTBAL.cz - FOTBALOVÉ ZPRAVODAJSTVÍ' and a 'Livescore' section with tabs for 'Fotbal', 'Tenis', 'Hokej', and 'Basketbal'. Below this is a navigation bar for 'All matches' with filters for 'Not started', 'In play', 'Finished', and 'Tennis | BWIN.com'.
- Main Content Area:**
  - News Articles:** Several articles are displayed, including 'Bristol šokoval Selhurst Park' (Bristol shocked Selhurst Park), 'celý článek' (full article), 'Skotská Premier League souhrn - 37. kolo' (Scottish Premier League summary - 37th round), 'Ramos připustil odchod Berbatova' (Ramos admitted Berbatov's departure), and 'celý článek' (full article).
  - Football Leagues:** A grid of football leagues is shown, including 'Champions League', 'Premier League', 'Championship', 'League One', 'League Two', 'Conference', 'Clausura', 'Jupiler League', 'Tweede Klasse', 'Premier League', 'Campeonato Serie B', 'PFG A', 'Serie A', 'Campeonato Serie B', 'Gambrinus liga', 'Druhá liga', 'Super League', 'SAS-Ligaen', '1st Division', 'Meistriliiga', 'Veikkausliiga', 'Ykkonen', 'Ligue 1', 'Ligue 2', 'National', 'Primera Division', '1. HNL', 'Eiroom Premier', 'Eiroom 1st Division', 'Landsbankadeild', 'Serie A', 'Serie B', 'Serie C1 - Girone A', 'Serie C1 - Girone B', 'Ligat ha'Al', 'J-League', 'A Lyga', 'Virsliga', 'Borsodi Liga', 'Primera Division', 'Bundesliga', '2. Bundesliga', 'Regionalliga Nord', 'Regionalliga Süd', 'Eredivisie', 'Gouden Gids', 'Tippeligaen', 'Adeccoligaen', 'Torneo Clausura', 'Orange Ekstraklasa', '2.liga', 'bwinLIGA', 'Liga Vitalis', 'T-Mobile Bundesliga', 'Red Zac Erste Liga', 'Liga 1', 'Premier league', '1. Division', 'Superliga', 'B Ethniki', 'S-League', 'Premier League', 'First Division', 'Second Division', 'Third Division', 'Corgoň liga', '1. Liga', 'Prva Liga', 'Primera Division', 'Segunda Division', 'Meridijan Superliga', 'Allsvenskan', 'Superttan', 'Superliga', 'Challenge League', 'Superliga', 'Lig A', 'Vyšja Liga', 'Primera Division', 'Premier League', 'MLS', 'Hokej', 'O2 extraliga', 'Oddset Ligaen', 'SM-liiga', 'Mestis', 'DEL Bundesliga', 'ASSTEL'.
  - Livescore Tables:** Multiple tables show live scores for various leagues:
    - Albania - Super League:** Beslidhja 1-0 SK Tirana, Dinamo Tirana 1-0 Besa, Kastrioti 0-0 Shkumbini, Partizani 1-0 Vllaznia, Skenderbeu 0-2 Flamurtari, Teuta 1-1 Elbasani.
    - Argentina - Apertura:** Banfield - Olimpo Bahia Blanca, San Martin - Lanus, Velez Sarsfield - Rosario Central, Racing Club - Argentinos Juniors.
    - Argentina - Nacional B:** Almirante Brown - Quilmes, Belgrano - Independiente Rivadavia, Club Ferro Carril Oeste - Tiro Federal, Chacarita Juniors - Ben Hur Rafaela, Defensa y Justicia - C. A. San Martin TUC.
    - Belarus - Vysshaya Liga:** Dnipro Dzerzhynsk 0-3 Dinamo Minsk, Granit Mikashevichi 1-0 Savit Mogilyov, Dnepr Mogilyov 1-1 MTZ-RIPO Minsk, FC Gomel 1-2 FC Vitebsk, FC Smorgon 0-0 BATE Borisov, Naftan Novopolotsk 1-1 Shakhtyor Soligorsk, Neman Grodno 0-1 Lokomotiv Minsk, Torpedo Zhodino 1-1 Dinamo Brest.
    - Belgium - Jupiler League:** Club Brugge - VC Westerlo, Excelsior Mouscron - Germinal Beerschot, FC V. Dender E. H. - AA Gent, KSC Lokeren OV - Roeselare, KV Mechelen - VV St. Truiden, Mons/Bergen - RSC Anderlecht, RKC Genk - Standard, SC Charleroi - Cercle Brugge KSV.
    - Bolivia - Torneo Apertura:** San Jose - Oriente Petrolero, Jorge Wilstermann - La Paz.
    - Bosnia and Herzegovina - Premier Liga:** Jedinstvo 0-0 HNK Orašje, Laktasi 2-0 NK Celik, Leotar Trebinje 1-1 Velez, Sloboda Tuzla 0-0 FK Sarajevo, Zeljeznica Srebrenica 0-0 Slavija Srebrenica.

Obrázek 4.10: www.vitissport.cz

# Kapitola 5

## Závěr

Výsledkem mé bakalářské práce je aplikace Nástroj pro sledování změn webu. Tento nástroj slouží ke kontrole zadaných internetových stránek a vyhodnocování jejich změn, které zvýrazní uživatelem definovanou barvou, velikostí písma popřípadě rámečkem. Je zde možnost nastavení zvýrazňování pro obyčejný text, odkazy a obrázky samostatně. O změně kontrolované stránky může být uživatel informován několika způsoby, například zvukovou signalizací pomocí pc speakru, vyskakovacím popup oknem či automatickým otevřením změněné stránky v internetovém prohlížeči. Je dokonce možno kontrolovat dynamické internetové stránky se zadanými parametry v adrese této stránky. Většina uživatelů preferuje ovládání počítače myší. S ohledem na tuto skutečnost je má aplikace ovládána převážně tímto způsobem, ale jsou zde k dispozici také klávesové zkratky do různých nabídek menu. V některých operačních systémech existuje také možnost aplikaci minimalizovat do oblasti system tray, aby uživateli příliš neprekážela. Seznam kontrolovaných internetových stránek je možno uložit na pevný disk ve formátu XML a poté jej také načíst. Aplikace při změně nastavení či změně seznamu stránek ukládá všechna nastavení aplikace a seznam kontrolovaných stránek do domovského adresáře uživatele, odkud si jej po spuštění aplikace automaticky načte. Uživatel si samozřejmě může seznam stránek kdykoli uložit do libovolného adresáře, odkud si jej může opět načíst.

Aplikace je navržena pro budoucí snadnou rozšiřitelnost, například o přidání podpory kontroly chráněných internetových stránek pomocí protokolu https či kontroly obsahu FTP serveru. Další možné rozšíření bych viděl v možnosti odesílat informaci o změně stránky či část změněného textu na email, kde si již uživatel může nastavit přeposlání na mobil pomocí SMS zpráv. Také by byla vhodná implementace system tray pomocí jiné knihovny či novější verze stávající knihovny, jelikož aplikace na některých verzích operačních systémů nefunguje v sytem tray vůbec či se chová nestabilně. V budoucnu by mohly existovat knihovny s větší podporovou operačních systémů a stabilitou.

Přínos projektu vidím v tom, že příliš mnoho podobných aplikací neexistuje, a pokud ano, nejsou multiplatformní a nenabízí tolik možností nastavení zvýrazňování změn, což může být pro některé uživatele problém. Aplikace je velmi dobře využitelná na internetových stránkách, které nedisponují technologií RSS či velmi často nemění svůj obsah. V druhém případě se nastaví interval kontroly například na 24 hodin, jelikož uživateli prodleva pár hodin na zjištění změny takového typu internetové stránky příliš vadit nebude.

Velmi velkým pomocníkem mi byly knihy od Pavla Herouta [12][13][11] a Pavola Mikla [14].

# Literatura

- [1] Ant [online]. rev. 6.2.2003. Dostupné na URL <http://www.root.cz/clanky/ant-nebojte-se-mravence/>.
- [2] Apache ant [online]. rev. 11.5.2008. Dostupné na URL <http://ant.apache.org/>.
- [3] Java a základy gui [online]. rev. 6.1.2008. Dostupné na URL <http://www.zaachi.com/cs/items/java-a-zaklady-gui.html>.
- [4] Jdic - jdesktop integration components [online]. rev. 10.11.2006. Dostupné na URL <https://jdic.dev.java.net/>.
- [5] Systray for java [online]. rev. 28.4.2004. Dostupné na URL <http://systray.sourceforge.net/>.
- [6] Robert Eckstein. System tray in java se 6 [online]. rev. listopad 2005. Dostupné na URL <http://java.sun.com/developer/technicalArticles/J2SE/Desktop/javase6/systemtray/>.
- [7] Network Working Group. Rfc 1945 [online]. rev. květen 1996. Dostupné na URL <http://www.ietf.org/rfc/rfc1945.txt>.
- [8] Network Working Group. Rfc 2616 [online]. rev. červen 1999. Dostupné na URL <http://www.ietf.org/rfc/rfc2616.txt>.
- [9] Dušan Janovský. Jak psát web [online]. rev. 26.4.2008. Dostupné na URL <http://www.jakpsatweb.cz/css/css-uvod.html>.
- [10] Jeff Pace. Java-diff [online]. rev. 17.8.2006. Dostupné na URL <http://www.incava.org/projects/java/java-diff/>.
- [11] Herout Pavel. *Java – bohatství knihoven*. Kopp, 2006. ISBN-80-7232-288-5.
- [12] Herout Pavel. *Učebnice jazyka java*. Kopp, 2006. ISBN-80-7232-115-3.
- [13] Herout Pavel. *JAVA – grafické uživatelské prostředí a čeština*. Kopp, 2007. ISBN-978-80-7323-328-9.
- [14] Mikle Pavol. *xCSS - referenční příručka*. Zoner Press, 2006. ISBN-80-86815-13-7.
- [15] W3C. Cascading style sheets, level 2 css2 specification [online]. rev. 12.5.1998. Dostupné na URL <http://www.w3.org/TR/REC-CSS2/>.
- [16] W3C. Html 4.01 specification [online]. rev. 24.12.1999. Dostupné na URL <http://www.w3.org/TR/REC-html40/>.

- [17] Wikipedia. Swing (java) [online]. rev. 19.4.2008. Dostupné na URL [http://en.wikipedia.org/wiki/Swing\\_\(Java\)](http://en.wikipedia.org/wiki/Swing_(Java)).
- [18] Wikipedie. Extensible markup language [online]. rev. 1.5.2008. Dostupné na URL <http://cs.wikipedia.org/wiki/XML>.
- [19] Wikipedie. Hypertext markup language [online]. rev. 3.5.2008. Dostupné na URL <http://cs.wikipedia.org/wiki/Html>.
- [20] Wikipedie. Hypertext transfer protocol [online]. rev. 11.5.2008. Dostupné na URL <http://cs.wikipedia.org/wiki/HTTP>.

# Dodatek A

## Přílohy

### A.1 Nápověda

Nápověda pro nástroj pro sledování změn webu

#### První spuštění

**Po prvním spuštění má program toto nastavení:**

Pracovní adresář je umístěn v uživatelské domovské adresáři.

Perioda časovače je nastavena na 30 sekund.

Počet kliknutí do buňky "změna" seznamu stránek, který otevře stránku v prohlížeči je nastaven na DVĚ

Automatické otevírání prohlížeče při změně stránky je POVOLENO

Barva popup okna je nastavena na žlutou

Zvuková signalizace změny stránky je POVOLENA

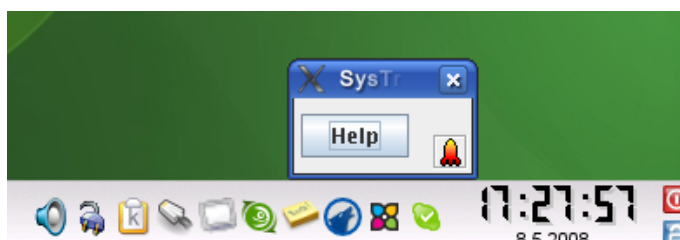
Automatické spuštění kontroly stránek je VYPNUTO

Zobrazení aplikace v system tray je VYPNUTO

#### Zobrazení aplikace v oblasti system tray

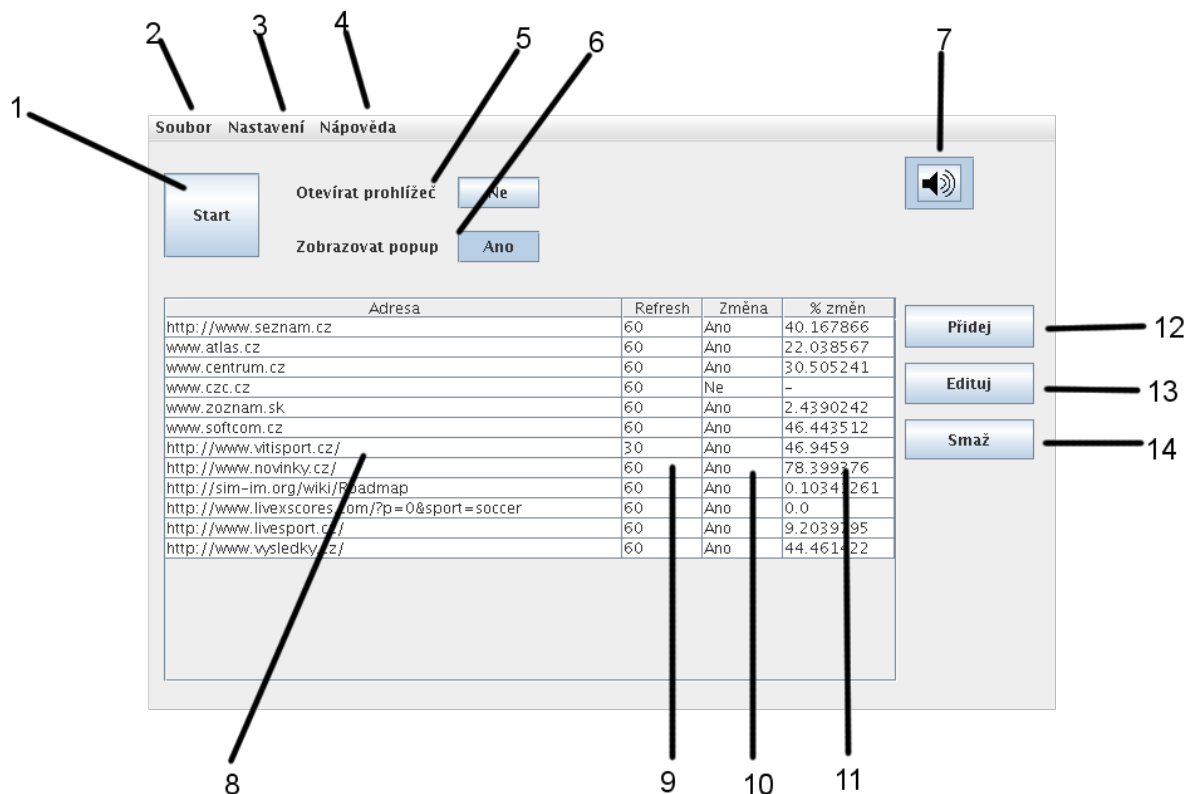
Aplikace má možnost být zobrazena také v oblasti system tray. Tato volba je defaultně vypnuta kvůli potřebě nahrát knihovnu *systray4j.dll* pro operační systém Windows nebo *lib-systray4j.so* pro linux do adresáře *váš\_adresář\_s\_javou/lib/i386*. Dalším omezením je funkčnost pouze v operačních systémech Windows 2000, Windows XP a linux s KDE verze 3. Bohužel ne vždy je tato knihovna v linuxu stabilní.

Pokud není knihovna umístěna na správném místě, je místo ikony v system tray zobrazeno okno, které je na obrázku níže. Po stisknutí tlačítka *Help* je uvedeno kam je nutno požadovanou knihovnu umístit.



## Hlavní okno - ovládání

Zde jsou umístěny základní ovládací prvky



- 1 - tlačítko sloužící ke startu/zastavení kontrolování stránek
- 2 - menu obsahující uložení/načtení seznamu stránek a položku na ukončení programu
- 3 - menu s konfigurací aplikace
- 4 - menu obsahující nápovědu a informace o aplikaci
- 5 - nastavení automatického otevírání internetového prohlížeče při změně stránky
- 6 - nastavení zdali mají vyskakovat popup okna při změně stránky
- 7 - nastavení zvukového upozornění při změně stránky
- 8 - sloupec tabulky obsahující adresy kontrolovaných stránek
- 9 - sloupec tabulky obsahující časy kontroly
- 10 - sloupec tabulky obsahující zdali od poslední kontroly došlo ke změně
- 11 - sloupec tabulky obsahující kolik procent změn obsahuje stránka od poslední kontroly
- 12 - tlačítko sloužící k přidání nové kontrolované stránky
- 13 - tlačítko sloužící k editaci již existující stránky v seznamu
- 14 - tlačítko sloužící ke smazání existující kontrolované stránky

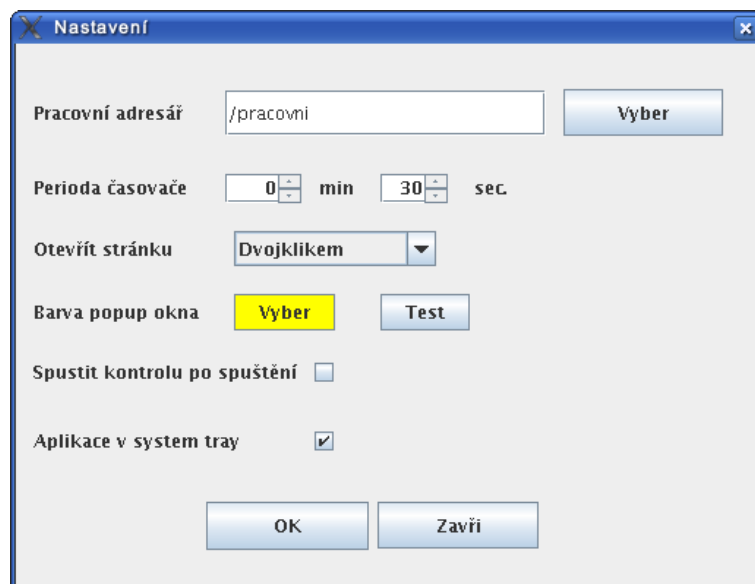
*Pokročilejší možnosti ovládání:* Dvojklik do buňky se zadanou adresou umožní její editaci Dvojklik do buňky se zadaným časem znovunačítání umožní jeho editaci Dvojklik či Jedno kliknutí – dle nastavení, zapříčiní v buňce změna otevření stránky v prohlížeči, pokud od



spuštění kontroly proběhla alespoň jedna změna

Pokud myš zůstane chvíli nad některým ovládacím prvkem zobrazí se nápověda co tento prvek dělá.

## Okno s konfigurací aplikace



Pracovní adresář - nastavení umístění pracovního adresáře

Perioda časovače - nastavení jak často se má kontrolovat seznam stránek, jestli se nemá zkontrolovat nějaká stránka. Hodnota by neměla být větší než nejnižší čas kontroly v seznamu stránek.

Otevřít stránku - nastavení zdali se má stránka otevřít dvojklikem do seznamu stránek do sloupce Změna

Barva popup okna - nastavení barvy popup okna s možností jeho otestování tlačítkem Test

Spustit kontrolu po spuštění - nastavení zdali se má po spuštění aplikace automaticky spustit kontrola stránek.

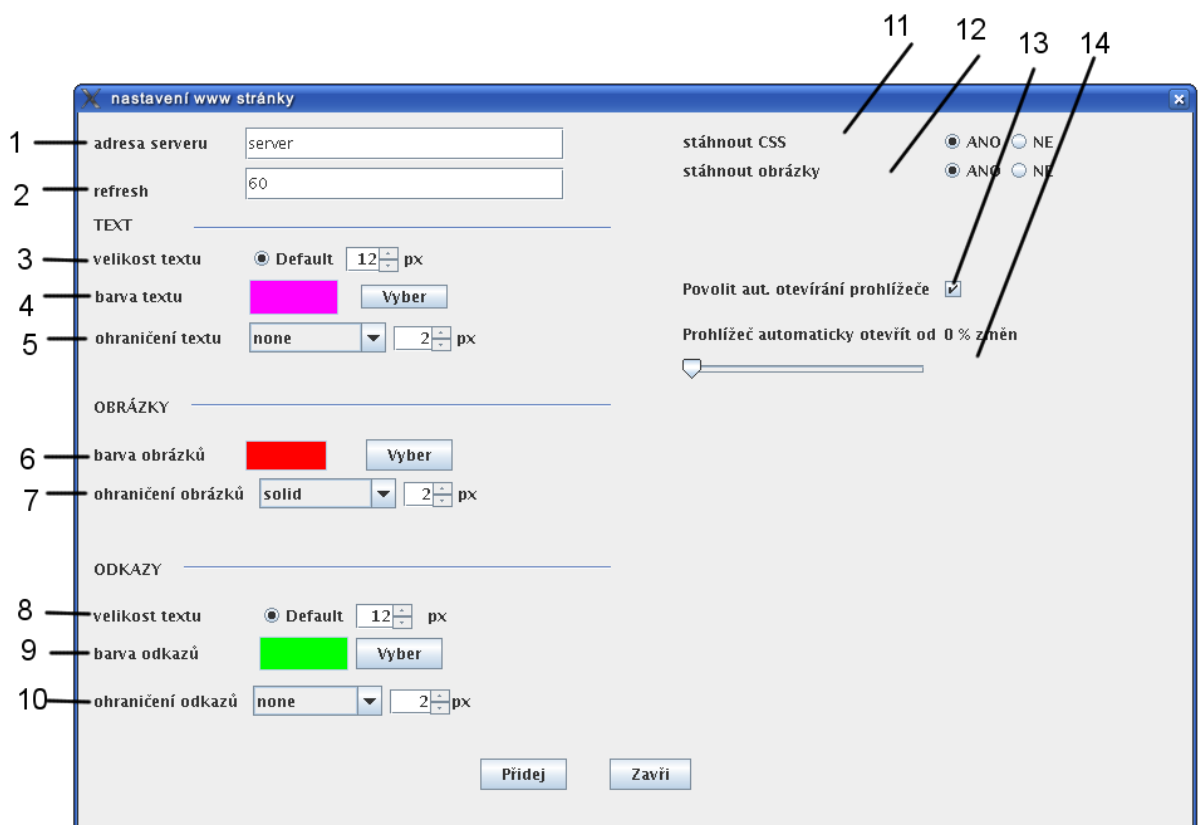
Aplikace v system tray - nastavení zobrazení aplikace v oblasti system tray (nutný restart aplikace)

Tlačítko OK uloží změny

Tlačítko Zavři zahodí všechny provedené změny

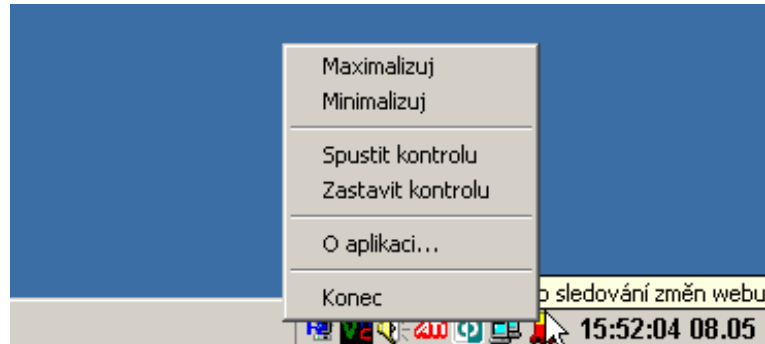
### **Přidání/editace kontrované stránky**

- 1 - internetová adresa kontrované stránky
- 2 - čas jak často se má stránka kontrolovat
- 3 - nastavení velikosti změněného textu
- 4 - nastavení barvy změněného textu
- 5 - nastavení ohraničení změněného textu
- 6 - barva ohraničení změněných obrázků
- 7 - styl a tloušťka rámečku kolem změněných obrázků



- 8 - nastavení velikosti textu ve změněných odkazech
- 9 - nastavení barvy textu změněných odkazů
- 10 - nastavení rámečku kolem změněných odkazů
- 11 - nastavení zdali se mají se stránkou stahovat také CSS soubory
- 12 - nastavení zdali se mají se stránkou stahovat také obrázky
- 13 - nastavení zdali se má při změně automaticky otevírat stránka v prohlížeči
- 14 - nastavení od kolika procent změn na stránce se má stránka automaticky otevřít v prohlížeči

## Přístup přes menu v system tray



Maximalizuj - zobrazení hlavního okna aplikace

Minimalizuj - minimalizace hlavního okna do system tray

Spustit kontrolu - spuštění kontroly internetových stránek

Zastavit kontrolu - pozastavení kontroly internetových stránek

O aplikaci... - zobrazení informací o aplikaci

Konec - ukončení aplikace

Aktualní verze nápovědy je dostupná na adrese

<http://www.stud.fit.vutbr.cz/~xsvabe02/napoveda/>.

## A.2 Médium CD

Na CD se v kořenovém adresáři nalézá technická zpráva v souboru `techzprava.pdf` obsahující text bakalářské práce. Adresář `zdrojove_soubory_zprava` obsahuje zdrojové soubory technické zpravy. Dále ve složce `/src` jsou umístěny zdrojové soubory aplikace. Aplikace je přeložitelná z hlavního root adresáře, který obsahuje soubor `build.xml`, pomocí aplikace Ant. Vlastní skripty pro překlad jsou umístěny v `/nbproject`. Přeložená aplikace je uložena v adresáři `/dist`. Knihovny potřebné pro správnou funkčnost aplikace v system tray jsou umístěny v adresáři `/system_tray`. Další knihovny nutné pro běh jsou v adresáři `/lib`. Adresář `/demo` obsahuje vzorový seznam internetových stránek včetně různých nastavení, který je možno přes menu aplikace soubor/`otevři...` načíst. Nápověda k aplikaci je uložena v adresáři `/napoveda`. Programová dokumentace je umístěna v adresáři `/doc`.