VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

APLIKACE PRO SROVNÁVÁNÍ CEN PRODUKTŮ

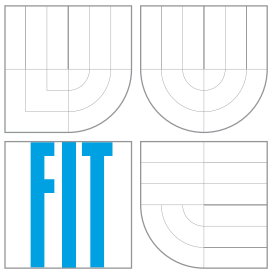
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAKUB VARADINEK

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

APLIKACE PRO SROVNÁVÁNÍ CEN PRODUKTŮ

ANALYSIS OF PRODUCTS PRICE COMPARISON

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAKUB VARADINEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PAVEL OČENÁŠEK, Ph.D.

BRNO 2015

Abstrakt

Tato diplomová práce se zabývá vývojem aplikace pro automatické srovnávání cen zboží. Aplikace funguje na principu, kdy uživatel zadá vstup v podobě seznamu produktů a výstupem je seznam nabídek sledovaných prodejců pro všechny zadané produkty. Uživateli je dána možnost vytvořit si individuální nastavení pro jednotlivé webové stránky prodejců nebo využít vyhledávání na již existujících agregátorech cenových nabídek. Cílová skupina uživatelů jsou prodejci, distributoři a výrobci, kteří využijí získaná data pro sledování konkurence a vývoje cen na trhu.

Abstract

This master's thesis deals with the development of the application for automatic price comparison of goods. The application works on the principle that the user enters inputs as a list of products and output is a list of offers from monitored dealers for all entered products. There is possibility for the user to create his individual settings for each website or use searching on existing comparison shopping website. Target group of users are dealers, distributors and manufacturers who use the data for monitoring competitors and changes in market prices.

Klíčová slova

srovnávání cen, cenová strategie, analýza trhu, web robot, sběr dat

Keywords

price comparison, pricing strategy, market analysis, web robot, web harvesting

Citace

Jakub Varadinek: Aplikace pro srovnávání cen produktů, diplomová práce, Brno, FIT VUT v Brně, 2015

Aplikace pro srovnávání cen produktů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Pavla Očenáška Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jakub Varadinek
25. května 2015

Poděkování

Rád bych poděkoval vedoucímu mé práce Ing. Pavlu Očenáškovi, Ph.D. za cenné rady a konzultace.

Obsah

Kapitola 1

Úvod

V této diplomové práci bude popsán vývoj aplikace pro získání základních dat o požadovaných produktech z webů prodejců. Jedná se především o data jako je prodejní cena a skladová dostupnost. Využít je mohou převážně výrobci, distributoři a velcí prodejci pro stanovení prodejní strategie u jednotlivých výrobků. Výstupem aplikace je přehled, ve kterém je vidět námi stanovenou cenu produktu a cenu případně i skladovou dostupnost u konkurence. Takto vytvořený přehled je poté možné využít pro další analýzy a sledování chování konkurentů.

Aplikace bude podporovat možnost mít uložené nastavení pro stahování dat. Především se jedná o adresu webu a obecného označení dat pro stažení pomocí např. jazyka XPath. Nastavení pro jednotlivé weby a seznam produktů, ke kterým se mají získat data, budou poté vstupem při automatickém spouštění aplikace např. na serveru. Uživatelé pak dostanou výsledky v podobě přehledu cen a skladové dostupnosti.

Aplikace podobného typu jsou v dnešní době realizovány převážně formou předplacené služby. Firma, používající takovou službu, si nadefinuje weby a produkty, které chce sledovat. Celkový přehled je pak k dispozici na webu dané služby nebo se přímo zasílá dohodnutým způsobem. Cena, která se za takovou službu platí, se odvíjí zejména od počtu sledovaných produktů a webů. Aplikace z této diplomové práce dává možnost si vytvořit vlastní nastavení a získávat data bez nutnosti jejich sdílení se třetí stranou. Řeší také problém při sledování velkého množství produktů, kde může být cena předplatného u webové služby neúměrně vysoká.

1.1 Struktura práce

Práce je členěna do jednotlivých kapitol, které pořadím odpovídají postupnému vývoji aplikace. V první fázi byla provedena analýza požadavků na aplikaci a nalezena již existující řešení. Celá analýza je popsána v kapitole 2. Volbou technických prostředků a jejich základního popisu pro stahování dat z webů se zabývá kapitola 3. V další kapitole 4 je popsán samotný návrh aplikace, což znamená soupis základních podporovaných funkcí, konceptuální diagram tříd a návrh uživatelského grafického rozhraní.

V kapitolách následujících po návrhu aplikace, je popsána samotná realizace a testování. Implementace je popsána v kapitole 5, kde jsou přítomny i hlavní snímky grafického rozhraní. V předposlední kapitole 6 jsou k nalezení základní analýzy, které je možné nad získanými daty provést. Poslední kapitola 7 rozebírá testování na webech velkých prodejců a popisuje případné problémy při stahování nebo párování dat.

Kapitola 2

Analýza požadavků

Kapitola rozebírá požadavky na vytvářenou aplikaci a možnosti, které nabízejí již existující komerční řešení. Existujících řešení je více a jsou vylepšována podle toho, jak postupně rostou požadavky firem na stále detailnější analýzu a přesnější data. Zároveň je zde rozebrána důležitost informací o prodejních cenách konkurence pro zvolení prodejní strategie u daného výrobku.

V podkapitole 2.1 obsahuje samotnou analýzu cen konkurence z pohledu dodavatele zboží (výrobce nebo distributor) a prodejce pro koncové zákazníky. V další části 2.2 je soupis základních požadavků na aplikaci z hlediska nabízené funkcionality. Jedná se spíše o rámcový přehled požadovaných klíčových vlastností. Na to navazuje v další podkapitole 2.4 popis existujících řešení, která jsou nabízena formou předplacené služby, a jejich možnosti. Poslední část 2.5 je věnována možným zdrojům dat. Je zde proveden výběr největších českých agregátorů nákupních nabídek a webů prodejců. Tyto zdroje dat budou poté využity při vývoji a testování aplikace.

2.1 Analýza cen konkurence

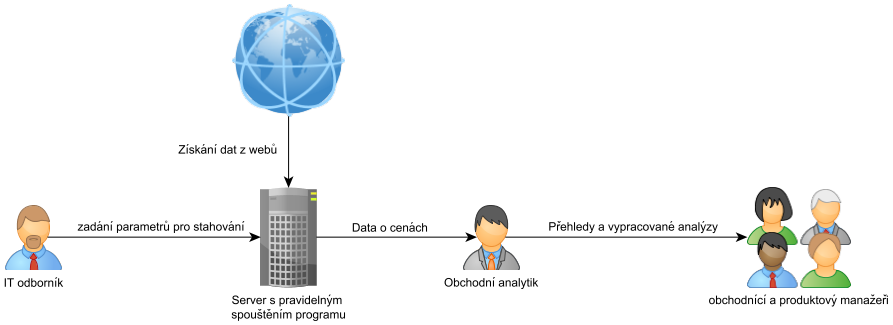
Získávání dat o cenách z webů prodejců a jejich další využití při stanovení cenové strategie je v dnešní době důležité pro konkurenční boj. Ze získaných dat je možné vyčíst mnoho údajů a také odhalit strategii, kterou používá úspěšnější konkurence. Využití získaných dat se liší podle konkretních subjektů. Pohled dodavatelů je rozebrána v části 2.1.1 a pohled prodejců v části 2.1.2.

2.1.1 Pohled dodavatele

Dodavatelem obecně rozumíme výrobce nebo distributora. Jedná se o subjekt, který se nespecializuje na prodej produktů přímo koncovým zákazníkům. Zbožím zásobuje prodejce a poskytuje jim obratové a reklamní bonusy. Data o cenách tedy využívá nejen ke konkurenčnímu boji, ale i pří sledování způsobu prodeje u svých zákazníků.

Z dat lze vytvořit potřebné analýzy pro kontrolování dohod s odběrateli. Přesně se určí, kdo a kdy porušil stanovenou dohodu. Přehled může být využit jako podklad při jednání s vybranými odběrateli. Rychlá detekce nežádoucího stavu, může zamezit případným problémům s dalšími odběrateli.

Díky možnosti získání dat ze standardních srovnávačů, popsaných v kapitole 2.5.1, lze objevit i nové potenciální zákazníky. Informace o zboží, které ve svých obchodech prodávají,



Obrázek 2.1: Schéma využití aplikace v podniku

pomohou vytvořit nabídku přímo na míru pro oslovení daného prodejce. Je tak možné přebrat konkurenčním firmám jejich stávající zákazníky a dosahovat lepších výsledků.

2.1.2 Pohled prodejce

Prodejce využívá datový přehled pro přímé porovnání svých cen s konkurencí. Pomáhá mu určit optimální cenu u jeho produktů. Zároveň získá informace o produktech, které je vhodné vyřadit z portfolia produktů, ať už kvůli nemožnosti konkurovat cenou nebo zastaralosti daného produktu.

Pomocí přehledu může prodejce také sledovat vývoj na trhu. Ze získaných dat je možné zjistit, které produkty konkurence nenabízí, a vyhledat produkty, unichž lze dosáhnout nejvyššího zisku. Po provedení analýzy dat v delším časovém horizontu lze odhalit způsob přeceňování, který používá konkurence.

2.2 Požadavky na aplikaci

Informace o cenách zboží a cenové strategii konkurence jsou klíčové pro úspěch firmy. O data mají zájem především velké společnosti z oblasti elektronického obchodu, distributoři a také velcí výrobci. Důležité je především, aby data byla aktuální a správné. Aplikace tedy nebude určena pro koncové zákazníky, kteří dnes mohou využít portálů jako heuréka[1] nebo zboží.cz[2]. Jednoduchý diagram využití aplikace ve firmě je možné vidět na obrázku 2.1.

Aplikace musí splňovat několik kritérií, aby byla použitelná v praxi. Především se jedná o aktuálnost a správnost dat. Obecně preferovaná varianta je získání přesných a aktuálních dat v okamžiku, kdy jsou potřeba. Z reálného hlediska stahování dat zabere určitý čas, který je dán výpočetním výkonem, rychlostí internetu, rychlostí odpovědi od serverů s daty a také technickým provedením aplikace. Pro praktické účely se jeví jako rozumná a dosažitelná doba jednoho dne. Data tedy budou aktualizována pravidelně jednou denně.

Vytvořená aplikace by měla být spustitelná automaticky s připraveným nastavením a vygenerovat žádaný datový výstup. Pro každý web bude nutné mít evidované vlastní nastavení navázaných odkazů a XPath výrazů. Pro vytváření takového nastavení bude aplikace disponovat uživatelským rozhraním, kde půjde připravené nastavení vyzkoušet. U uživatele, který bude připravovat nastavení, se přepokládá znalost HTML, XPath a pokročilejší zna-

[1] http://www.heureka.cz/
[2] http://www.zbozi.cz/

losti fungování webů. Aplikace při spuštění využije toto nastavení a vstupní data v podobě seznamu produktů s informacemi pro výsledný přehled.

2.3 Struktura výstupního souboru

ID produktu	PartNo	Název produktu	Vlastní cena	Název prodejce	Cena prodejce	Zdroj dat
5245896	NT.L15EE.003	Acer B1-A71	3 050	SG comp	3 143	heureka.cz
5245896	NT.L15EE.003	Acer B1-A71	3 050	Alza	3 255	alza.cz
5412355	EY.JEG04.001	Acer P1223	10 099	Alza	10 052	alza.cz
5412355	EY.JEG04.001	Acer P1223	10 099	CZC	10 129	czc.cz
5412355	EY.JEG04.001	Acer P1223	10 099	Mall	10 174	mall.cz

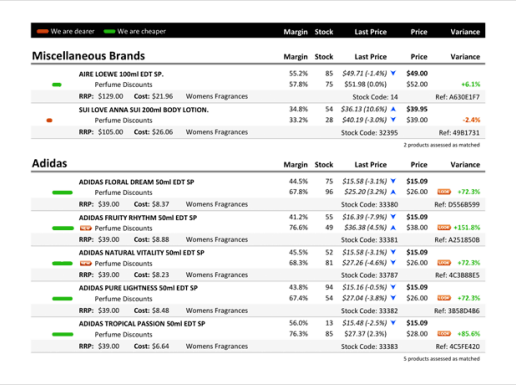
Tabulka 2.1: Ukázka možné podoby přehledu ve formě tabulky

Přehled bude vytvořen z dat získaných z webu a dat dodaných při spuštění aplikace. Data budou zapsána ve výsledném souboru tak, aby je bylo možné dále využít jako zdroje pro analýzy např. v programu Microsoft Excel. Předpokladem je, že samotná aplikace nebude provádět složité analýzy. V uživatelském rozhraní bude možné zvolit jaké data a v jakém pořadí mají být ve výstupním souboru.

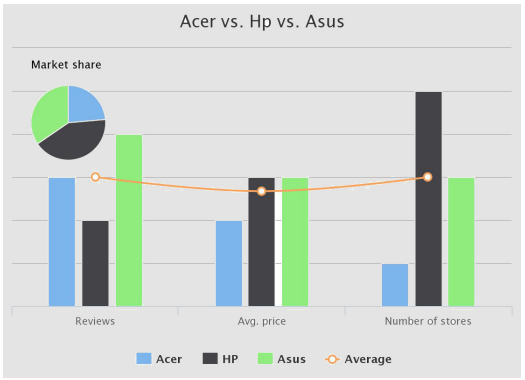
Tabulka 2.1 ukazuje možnou podobu přehledu. Jedná se pouze o variantu, kdy jsou obsaženy základní údaje. Předpokládá se, že v přehledu budou minimálně údaje uvedené v následujícím seznamu:

- **PartNo**: označení, které výrobce dal produktu. Tento údaj je možné použít při vyhledání produktu na standardních srovnávačích cen.
- **Název produktu prodejce**: název, pod kterým je produkt evidován u prodejce.
- **Název prodejce**: jméno, pod jakým prodejce vystupuje.
- **Cena prodejce**: cena produktu u prodejce.
- **Měna**: označení měny, ve které je cena prodejce.
- **Skladová dostupnost prodejce**: skladová dostupnost, kterou uvádí prodejce.
- **Fulltext**: pouze hodnoty 0 a 1. Příznak, zda nabídka z daného řádku byla získána fulltextovým vyhledáváním (menší spolehlivost pravdivosti dat).
- **Zdroj dat**: označení zdroje, kde byla data získána, např. heureka.cz nebo alza.cz.

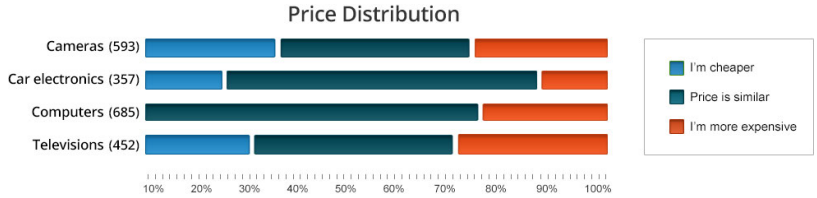
Jediný zadaný údaj od uživatele aplikace je v takovém případě part number pro identifikaci produktu. Ve vstupním souboru je možné zadat i další údaje, které má uživatel o produktu k dispozici a chtěl by je také mít i ve výstupním přehledu. Navíc mohou být ve vstupním souboru obsaženy také údaje využívané i pro jiné účely v rámci organizace. Uživatel má možnost zvolit, které vstupní údaje se budou přebírat do výstupu programu a které ne. V následujícím seznamu jsou uvedeny některé předpokládané údaje dodané uživatelem:



(a) Price Mentor převzato ze zdroje [1]



(b) Karsa Monitor převzato ze zdroje [2]



(c) Upstream Commerce převzato ze zdroje [3]

Obrázek 2.2: Ukázky z existujících nástrojů

- **ID produktu**: jedná se o ID, přidělené produktu uživatelem programu. Důležité je pro možnost provázat výsledky s databází uživatele.
- **Název produktu**: název, pod kterým produkt eviduje uživatel.
- **Vlastní cena**: cena produktu u uživatele.
- **Měna**: označení měny, ve které je **Vlastní cena**.

Počet řádků v takto definovaném výstupním souboru může být neúměrně vysoký. Velká organizace může mít v databázi vedeny i desetitisíce produktů. Předpokladem je, že se stahování údajů bude týkat menšího množství v řádu stovek významných produktů. V případě využití standardních srovnávačů může k jednomu produktu nalezeno i více než sto různých nabídek. Ve výstupu by tedy bylo řádově desetitisíce řádků, které by obsahovaly spoustu redundantních informací.

Program bude umožňovat, aby se data rozdělila i do více souborů s uvedenými společnými údaji. Může se jednat o rozdělení po prodejcích, produktech nebo podle zdrojů dat. Cílem je, aby výstup aplikace byl bezproblémový pro další využití.

2.4 Nástroje podobného typu

V posledních několika letech se nástroje pro automatickou analýzu konkurence na trhu neustále rozvíjí a vylepšují. Jedná se především o nástroje formou služby, tedy předplacení si přístupu k požadovaným informacím. Firmám jsou předávána požadované data a analýzy o cenách zboží na trhu. Výstupy mohou být jak ve formě samotných dat, tak i souhrnných analýz.

Firmám je umožněno dynamicky reagovat na změny na trhu. Mají u jednotlivých produktů přesné informace o cenách, které nastavila konkurence. Je možné také z dat vyčíst aktuální cenové trendy pro jednotlivé kategorie zboží. Pro firmu může být také důležité identifikovat produkty, jejichž prodejní ceny u konkurence jsou dlouhodobě nižší, než je nákupní cena u dodavatele.

2.4.1 Karsa monitor

Jeden z předních nástrojů pro monitorování českého a slovenského trhu je Karsa Monitor[3]. Tento software vyvíjí firma se sídlem v České republice. Celkově je podle údajů z webových stránek pokryto 3 221 948 produktů v 324 e-shopech z deseti různých evropských zemí.

Karsa Monitor funguje formou služby, kdy zákazník zaplatí za počet produktů a e-shopů, které chce sledovat. K dispozici je aktuální denní přehled včetně analýz. V nabídce jsou jak analýzy určené pro dodavatele, tak i pro prodejce. Službu využívají velcí výrobci jako Acer, Toshiba, Epson a další. Mezi významné prodejce využívající tuto službu patří např. Alza.

2.4.2 Řešení od Upstream Commerce

Jedná se o americkou společnost[4], která nabízí kompletní analýzy a sledování cen. Od Karsa monitor se odlišuje především rozsáhlejší nabídkou služeb. Kromě standardního přehledu a analýz cen konkurence je nabídnuta i možnost implementace automatického reagování na ceny konkurence. Obchodník si může vytvořit vlastní pravidla, jak se má cena měnit podle získaných údajů.

Řešení také umožňuje sledovat způsob zařazení zboží u konkurence. Jednoduše zjistí, zda je způsob rozřazení zboží podobný konkurenci nebo odlišný. Poskytnuty jsou také údaje k produktům podle jejich doby života. Obchodník zjistí, zda má produkt stále nabízet nebo ho z nabídky vyřadit.

2.4.3 Price mentor

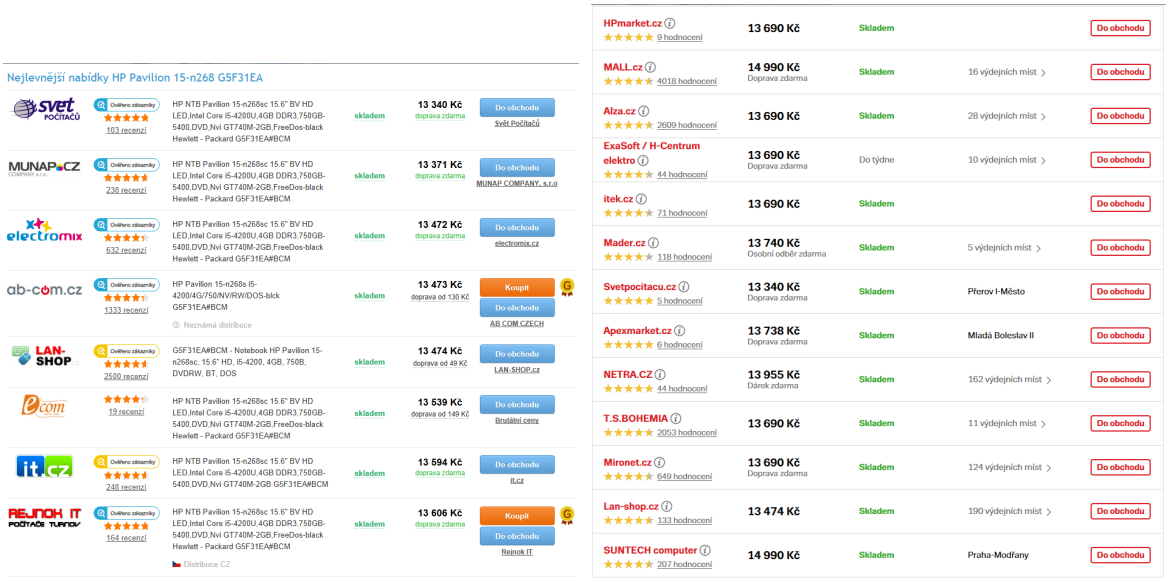
Price mentor[5] je řešení od australské společnosti pro zjišťování cen. Poskytuje velice podobné možnosti jako dříve popsaný Karsa Monitor. Zákazníkovi je k dispozici souhrnný výstup o cenách konkurence s údaji o změně ceny v čase. Zároveň jsou dodávány údaje o cenové strategii sledovaných konkurentů.

2.5 Možné zdroje dat

Obecně by aplikace měla umožnit nadefinovat si nastavení pro libovolný web. Pro účely testování a vytvoření analýzy výsledků budou zvoleny vhodné zdroje dat. Mělo by se jednat o největší prodejce a portály pro srovnání cen produktů v rámci českého trhu.

Výhodné je v první řadě využít standardních srovnávačů, jako je zboží.cz od Seznamu nebo ještě populárnější heuréka. Více o možnostech využití těchto srovnávačů pro získání dat a případné výhody a nevýhody, které jako zdroj dat přinášejí, jsou popsány v podkapitole 2.5.1. Druhým zdrojem jsou přímo stránky prodejců. Nejvhodnějšími jsou známí

[3] http://www.karsa-monitor.cz/
[4] http://upstreamcommerce.com/
[5] http://www.pricementor.com/



(a) Heureka převzato ze zdroje [4]

(b) Zboží převzato ze zdroje [5]

Obrázek 2.3: Srovnávače heureka.cz a zboží.cz

prodejci, kteří mají vysoký obrat a široké produktové portfolio. Jedná se o velké elektronické obchody jako Alza nebo Internet Mall.

Zdroje dat budou pro účely této diplomové práce a prvotní vývoj aplikace výhradně z českého trhu. Výsledná aplikace nebude ovšem limitována a bude možné nastavit i libovolné weby ze zahraničí. Pro účely testování a vývoje je však vhodné zaměřit se v první fázi pouze na místní trh. Vybírány budou také přednostně obchody a srovnávače, které nabízí zboží z oblasti IT, fototechniky a elektra.

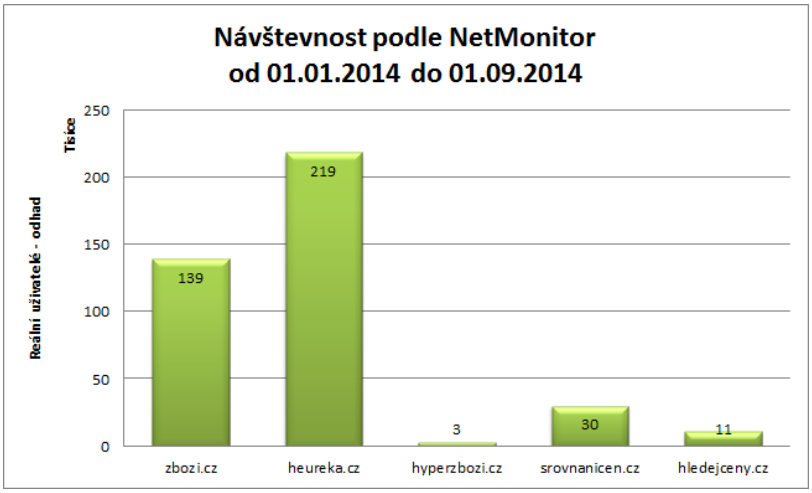
2.5.1 Standardní srovnávače cen

V této kapitole jsou popsány jedny z největších agregátorů nákupních nabídek v rámci českého trhu. Využívá je mnoho prodejců a jsou tedy vhodné pro získávání dat o cenách. Nejvíce navštěvované jsou podle NetMonitoru heuréka a zboží.cz. Jde pouze o odhady, ale lze vidět poměrně velký rozdíl v návštěvnosti mezi zmíněnými srovnávači a zbytkem.

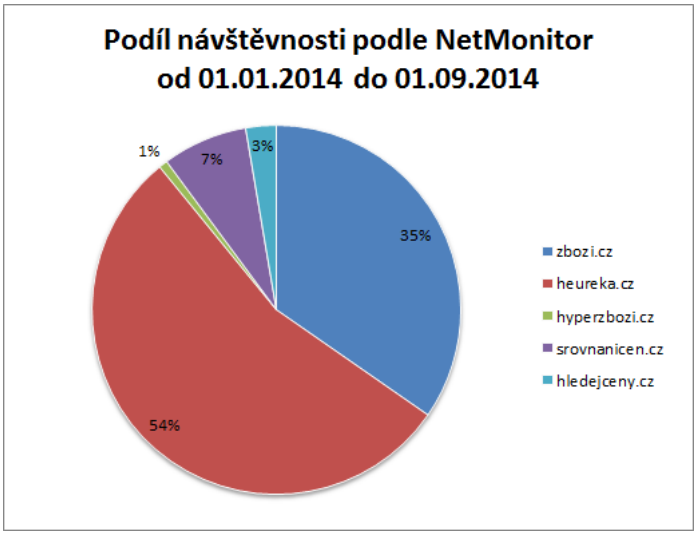
Struktura webu heuréka i zboží.cz je velmi podobná. Produkty mají na srovnávací stránku s popisem produktu a jednotlivými nabídkami od prodejců. Je tedy možné bez větších problémů takový výstup z webu zpracovat a zjistit ceny od jednotlivých prodejců, kteří data do srovnávače dávají.

Využívání srovnávačů cen jako zdroje dat má velkou výhodu v podobě rychlosti získání dat. Nemusí se procházet jednotlivé weby prodejců a při stažení dat z jedné stránky se dá rychle zjistit velké množství nabídek. Nevýhody spočívají převážně v případné neaktuálnosti a chybně zadaných údajích. Obchodník nemusí mít správně aktualizovaná data o ceně nebo svoji nabídku špatně do srovnávače zařadí. Druhým problémem může být skladová dostupnost, kterou někteří obchodníci na srovnávači neudávají.

Obecně je využití standardních srovnávačů jako zdrojů dat vhodné spíše pro informativní využití. Získaná data nemusí být přesná a ani není zaručeno, že jsou zahrnuty všechny obchody, které chceme sledovat. Přehled ze srovnávače je především vhodný pro další zpracování v případě, že nás zajímá spíše sledování trhu jako celku.



(a) Návštěvnost největších srovnávačů cen v ČR



(b) Podíl srovnávačů podle návštevnosti

Obrázek 2.4: Porovnání předních českých srovnávačů cen podle dat z NetMonitoru[6]

2.5.2 Weby elektronických obchodů

Využívání přímo webů prodejců je preferovaná varianta, pokud chceme získat co nejpřesnější data. Je možné si navázat jednotlivé stránky produktů u prodejců a stahovat požadovaná data. Odpadají tak problémy, které byly popsány v předchozí části při využívání agregátorů nákupních nabídek. Stažené informace budou přesně odpovídat webové stránce prodejce a navíc můžeme případně získat další informace, které stránka poskytuje.

Nevýhodou je nutnost nastavení velkého množství stránek. Kromě časové náročnosti přípravy nastavení pro každou stránku je nutné také brát do úvahy i možné změny na stránkách. Každou změnu bude nezbytné detekovat a případně upravit nastavení pro danou stránku. To může být při velkém množství kontrolovaných stránek poměrně časově náročné. Další technické problémy se mohou ukrývat v použité technologii pro realizaci webu. Problémové mohou být pro knihovny stahující data především technologie jako AJAX[6] nebo

[6] http://en.wikipedia.org/wiki/Ajax_(programming)

Společnost	**Tržby 2012** [tisíce Kč]	**Obchodní marže 2012** [tisíce Kč]	**Tržby 2013** [tisíce Kč]	**Obchodní marže 2013** [tisíce Kč]
HP TRONIC	5 532 277	210 276	5 419 064	46 280
Alza	7 473 639	437 802		
DATART INTERNATIONAL	4 561 887	994 733	4 535 676	92 063
Internet Mall	3 238 152	326 452	4 454 077	507 857
T.S.BOHEMIA a.s.	1 283 940	108 623		
CZC.cz s.r.o.	1 228 377	102 657	1 686 514	137 399

Tabulka 2.2: Porovnání předních českých e-shopů podle obratu z dat portálu justice[7]

obecně využití nestandardních JavaScript funkcí. U stránek velkých prodejců, které jsou vytvořeny profesionálně, takové problémy většinou nenastávají. Je ovšem nutné do úvahy vzít i variantu, kdy bude problém získat obsah stránky.

Pro účely této diplomové práce je vhodné se zaměřit hlavně na velké prodejce. V tabulce 2.2 je možné vidět přehled největších prodejců, kteří se zaměřují převážně na zboží z kategorie IT, foto a elektro. Podle obratu je největším prodejce Alza, následovaná firmou HP Tronic (provozovatel e-shopů jako Kasa nebo ePROTON), Datart a Internet Mall.

Kapitola 3

Stahování dat a web roboti

Cílem této kapitoly je rozebrat způsob přístupu k webu z pohledu webových robotů. Obecně jsou v dnešní době roboti pro web využíváni ve velkém množství. Slouží k různorodým činnostem od pouhého monitorování webu po automatické zadávání velkého množství dat a automatizaci časově náročných činností. Je možné je vyvíjet víceméně v libovolném programovacím jazyce, nutná je pouze možnost posílat a přijímat zprávy v protokolu HTTP. Pro získání znalostí o HTTP, které jsou zejména vhodné při práci s knihovnami pro procházení webů a při řešení složitějších problémů je vhodná knížky [8] a [9].

Aplikace pro srovnávání cen popisovaná v této diplomové práci bude vyvíjena v jazyce Java. Podkapitola 3.1 rozebírá knihovny, které slouží pro přístup k webům a zpracování HTML kódu. Další podkapitola 3.2 se zabývá problémy při hledání dat a možnosti jazyka XPath pro práci s HTML dokumenty. V poslední části 3.3 jsou popsány některé problémy a ochranná opatření, na které je možné narazit při hromadném stahování dat.

3.1 Java knihovny pro stahování z webu

Při volbě knihoven pro samotný vývoj aplikace byl využit zejména zdroj [10]. Pro samotné stahování dat slouží knihovna HtmlUnit, která je uživatelsky přívětivá a podporuje navíc technologie jako je Javascript a AJAX, které jsou na webech využívány.

Druhou knihovnou je Jsoup, která slouží jako syntaktický analyzátor jazyka HTML. Knihovna je celkově orientovaná na efektivnější práci s HTML kódem. Nabízí prostředky pro rychlé procházení elementů a má rozsáhlé možnosti pro provádění složitějších dotazů pro výběr dat.

3.1.1 Jsoup

Knihovna Jsoup[1] slouží pro práci s HTML kódem. Provádí převod do reprezentace DOM (Document Object Model) podobným způsobem, jako to dělají moderní webové prohlížeče podle specifikace WHATWG HTML[2] . Její výhoda je především v orientaci na reálné prostředí webu a zvládnutí převodu HTML kódu obsahujícího chyby. Vydána je pod svobodnou licencí MIT. Vývoj knihovny stále probíhá a jsou vydávány nové verze.

Jedná se o poměrně malou knihovnu, která je vhodná k rychlé práci s HTML. Má připravené metody pro posílání HTTP požadavků a dokáže jednoduše načíst webou stránku.

[1] http://jsoup.org/

[2] https://html.spec.whatwg.org/multipage/

Díky jednoduchému rozhraní a minimu speciálních funkcí dokáže knihovna velice rychle získat HTML kód ze zadané stránky. Absence JavaScript interpretu neumožňuje knihovnu použít na všech stránkách. Pokud by důležité data byla stahována po načtení stránky např. pomocí technologie AJAX, tak je použití této knihovny poměrně problematické.

3.1.2 HtmlUnit

HtmlUnit[3] je jedna z nejrozsáhlejších Java knihoven pro práci s weby. Nabízí spoustu možností a díky podpoře vykonávání JavaScript kódu dokáže zpracovat i poměrně technicky složité stránky. Vydána je pod svobodnou licencí Apache 2. Využívána je v projektech pro automatické testování webů jako je Canoo WebTest, JSFUnit nebo asi nejznámější nástroj Selenium[4].

Pro uživatele knihovny je připraveno velké množství tříd a předpřipravených možností. Prakticky všechny HTML elementy mají připravenou svoji třídu, která obsahuje základní i pokročilé metody. Práce s HTML v jazyku Java je tedy díky Htmlunit značně usnadněná. Předpřipravená nastavení především umožňují, aby se knihovna při práci s weby chovala stejným způsobem jako klasické prohlížeče typu Firefox, Chrome nebo Internet Explorer.

Čím knihovna především vyniká, je vestavěný interpret jazyka JavaScript a podpora AJAX knihoven. Z tohoto důvodu je právě HtmlUnit použita pro WebDriver[5] u nástroje Selenium. Přímo v dokumentaci Selenium je implementace WebDriveru pomocí HtmlUnit označovaná jako nejrychlejší. Problémem knihovny a podpory JavaScriptu je použití Rhino JavaScript Engine[6]. Jedná se o projekt, který implementuje JavaScript čistě pomocí jazyka Java. Bohužel není zaručena plná kompatibilita s interpretem JavaScriptu, který používají moderní prohlížeče. Mohou tedy nastat problémy při zpracování JavaScript kódu, který by moderní prohlížeče zvládly bez problému.

Na rozdíl od Jsoup tak HtmlUnit nabízí přímo koncept webového prohlížeče a stránek při práci s knihovnou. Je možné si vytvořit objekt, který simuluje webový prohlížeč, a dají se jeho pomocí získat objekty představující webové stránky. Na těch je možné poté simulovat akce jako zadání textu do formuláře, vybrání určitého prvku nebo kliknutí. Knihovna se sama stará o všechny procesy, které mají probíhat na pozadí, stejně jako tomu je u standardního webového prohlížeče s grafickým rozhraním. Koncepty a přístup použitý v HtmlUnit tedy odstiňuje programátora od technických detailů a dovoluje rychlejší a pohodlnější vývoj aplikace.

3.2 Vyhledávání a párování dat

Vyhledání a párování dat se v aplikaci bude řešit jak v případě vyhledání stránky produktu, tak v případě stahování konkrétních informací o produktu. Vyskytují se zde podobné problémy, jaké jsou více z teoretického pohledu popsány v knize o dolování dat z webu [11]. Využity budou především vhodné identifikační údaje produktu a v případě konkrétních stránek poté použity připravené XPath výrazy. Pomocí XPath výrazů budou ze stránky získávána konkretní data, jako je cena a skladová dostupnost.

Problém vyhledávání dat podle zadaných údajů se v aplikaci bude řešit především v případě, kdy uživatel bude využívat jako zdroj dat ostatní srovnávače cen, viz část 2.5.1. Je

[3] http://htmlunit.sourceforge.net/
[4] http://www.seleniumhq.org/
[5] http://www.seleniumhq.org/docs/03_webdriver.jsp
[6] https://developer.mozilla.org/en-US/docs/Mozilla/Projects/Rhino

možné využít přímo vyhledávače na webu srovnávače a produkty vyhledat podle identifikačního čísla. Uživatel při takovém použití bude muset počítat s horší přesností a z výstupních dat poté vyfiltrovat špatně zařazené výsledky hledání.

Pro vyhledávání zboží je obecně špatným postupem použít pouze název, který nemusí být unikátní a navíc vyhledávače vrací velké množství různorodých výsledků. Je zde nutné využití unikátního identifikačního čísla pro daný produkt. V případě zboží z IT se nejčastěji využívá identifikace produktu podle part number. Další možností je využití mezinárodního čísla obchodní položky (známé pod zkratkou EAN). EAN je univerzálnější při vyhledávání u různých typů produktů.

3.2.1 Možnosti XPath

Jedná se o jazyk pro adresování elementů v XML dokumentu. Rozebraný popis jazyka XPath včetně příkladů je možné nalézt např. ve zdroji [12] a [13] Způsob používání XPath (XML Path Language) je velice podobný popisu cest k souborům v operačním systému. Navíc umožňuje zadávat do výrazů podmínky a používat připravená klíčová slova pro navigaci v rámci zpracovávaného dokumentu.

Výrazy v jazyku XPath se skládají z kroků, které jsou oddělené lomítkem. Pohyb v daném XML souboru se děje v modelu XML, který je ve formě stromu. Obsahuje celkem sedm typů uzlů, jejichž vazby jsou reprezentovány pomocí orientovaných hran. Kořen této stromové reprezentace pro jazyk XPath není totožný s uzlem kořenového elementu. Uspořádání uzlů odpovídá uspořádání dokumentu.

Jazyk XPath lze využít také pro HTML dokumenty. Vytváření výrazů, které získají data ze zpracovávaného HTML dokumentu je snadnější, než vytváření algoritmu založeného na bázi metod pro získávání elementů z dokumentu. Navíc je možné využít relativních cest a podmínek, za pomoci kterých lze vytvořit více obecný dotaz. Dobře vytvořený XPath dotaz poté může fungovat, i když se později přetvoří části zpracovávané stránky. Výrazy mohou být předkompilované, čímž se výrazně zrychlí jejich opakované provádění.

3.3 Problémy při stahování dat

Hromadné stahování dat sebou přináší některé problémy, které při běžné uživatelské práci s webovou stránkou nenastávají. Weby srovnávačů cen a obchodníků jsou standardně přizpůsobeny koncovému uživateli, který používá jeden z několika hlavních webových prohlížečů. Weby nejsou po obsahové stránce ani po stránce technické vytvořeny pro možnost automatického stahování dat. Pro vlastníky stránek je naopak často nežádoucí, aby bylo jednoduché data ze stránek získat. Je tedy možné se setkat i s aktivní ochranou a omezením přístupu.

3.3.1 Špatná struktura stránky

Tento problém je zejména spojen s weby, které nejsou profesionálně vytvořeny. Web nemusí být z hlediska struktury natolik univerzální, že je složité nalézt obecné XPath výrazy, kterými by bylo možné z webu získávat data. Na webu mohou vypadat jednotlivé stránky rozdílně podle typu produktu. Struktura se také může lišit podle data zadání produktu na web, např. obchodník začal používat novou šablonu, ale nepřevedl do ní všechny stávající produkty. U velkých obchodníků takové problémy většinou nenastávají, protože je pro ně důležité držet jednotnou strukturu stránky.

Pokud XPath výrazy nebudou moci být dostatečně obecné kvůli struktuře stránky, může později vznikat problém při změnách v rozložení na dané stránce. Nastavení pro stahování z takové stránky bude muset být častěji opravováno, aby bylo funkční. Tento problém je v rámci aplikace prakticky neřešitelný. Zkušený uživatel pozná při zkoumání dané stránky, zda je její struktura vhodná. Je poté na rozhodnutí uživatele, zda chce ze stránky data stahovat i za cenu možných problémů a náročnější údržby nastavení aplikace.

3.3.2 Omezení počtu přístupů

Provozovatel stránky může z důvodu omezení možností webových robotů pro stahování dat omezit počet přístupů ke stránce za určitý čas. Pokud dojde v krátkém čase k nezvykle velkému množství požadavků z jedné IP adresy, tak se ochranný systém aktivuje. Web potom vrací na všechny dotazy pouze výstražnou stránku a není možné nadále přistupovat k požadovaným datům.

Často je forma této ochrany spojena se zobrazením CAPTCHA (viz další část této kapitoly). Pokračování v procházení webu je tedy umožněno až po správně zadané odpovědi na kontrolní otázku. Ideální je nastavit omezení počtu přístupů na danou stránku za jednotku času, aby se ochrana vůbec neaktivovala. Takové omezení samozřejmě povede k poklesu rychlosti zpracování stránky. Případně je vhodné se webům s takovou ochranou vyhnout úplně nebo omezit množství dat, která se mají stahovat.

3.3.3 Ochrana formou CAPTCHA

Jedná se o standardní ochranu proti webovým robotům, která se na internetu používá. Samotný akronym CAPTCHA[7] v češtině znamená plně automatický veřejný Turingův test[8] k odlišení počítačů od lidí. Používá se převážně při registraci, posílání příspěvků a obecně u všech akcí, kde je cílem zamezit webovému robotu, aby ji mohl vykonat. Pro uživatele CAPTCHA představuje úkol, který dokáže splnit člověk, ale nikoliv stroj. Může jít o opsání textu z obrázku nebo o jednoduchou otázku, často matematického typu.

Data o cenách jsou ovšem většinou veřejně přístupné bez registrace. Běžný uživatel se tedy s tímto ochranným opatřením při procházení stránky nesetká. CAPTCHA se na stránce aktivuje většinou jako bezpečnostní opatření, pokud si systém myslí, že stránku prochází webový robot. Jedná se právě o ochranné opatření proti hromadnému stahování z daného webu.

Prolomení správně použité ochrany CAPTCHA je problematické. Je možné vytvořit naoř. databázi obrázků, které se pro ochranu na daném webu používají, a manuálně jim přiřadit řešení. Automatickým prostředkům s takovou databází poté stačí pouze vyhledat řešení k danému obrázku v databázi. Další možností je mít automatické řešení a předpokládat, že CAPTCHA nebude příliš složitá a použitá metoda OCR (optical character recognition) bude schopná znaky rozpoznat.

Ochranu tohoto typu nelze v rámci možností aplikace vyvíjené v této diplomové práci jednoduše obejít. Snahou tedy je, aby se daná ochrana na stránce v prvé řadě vůbec neaktivovala. V rámci vývoje této aplikace nebude přímá snaha automaticky obejít takové ochranné opatření stránky. Důležitá bude hlavně detekce, že k problému došlo a není možné pokračovat ve stahování dat.

[7] http://www.captcha.net/

[8] Informace o Turingovu testu je možné naleznout na http://loebner.net/Prizef/TuringArticle.html

3.3.4 Ostatní technické problémy

Jedná se především o problémy s technologiemi jako je JavaScript a AJAX, které se na webech standardně vyskytují. Web je odladěný pro použití na uživatelský prohlížeč a problém nastává při přístupu skrze programové knihovny. Tyto knihovny nemusí mít podporu pro zmíněné technologie vůbec nebo nejsou kompatibilní se způsobem, jak jsou na daném webu použity. Nepodporované jsou technologie jako je Flash, jehož používání je na prodejních webech v dnešní době spíše minimální a netýká se přímo stránek s produkty.

Do úvahy je tedy nutné především vzít možnost, že skrze knihovny v aplikaci nemusí jít web správně načíst. Některé technologie, které dnešní prohlížeče bez problému zvládají, představují těžko řešitelný problém, pokud jde o automatický přístup k datům. Řešení je poté nutné hledat pro konkrétní technologii nebo přímo pro konkrétní technický problém, který nastane.

Kapitola 4

Návrh aplikace

Tato kapitola rozebírá návrh aplikace, který bude využit při samotné implementaci. Návrh je udělán pro platformu Java SE s grafickým uživatelským rozhraním realizovaným pomocí technologie Java Swing[1]. Zvolené programovací nástroje umožňují využívat velkého množství připravených funkcionalit a výslednou aplikaci bude možné používat na více platformách.

V první části 4.1 této kapitoly je samotný konceptuální návrh celé aplikace v podobě UML diagramů. Jsou zde diagramy případů použití, balíčků a také konceptuální diagram tříd. Ve druhé části 4.2 je popsáno grafické uživatelské rozhraní. Kromě seznamu funkcí, které má rozhraní plnit, je zde i nastíněna jeho základní podoba.

4.1 Konceptuální návrh

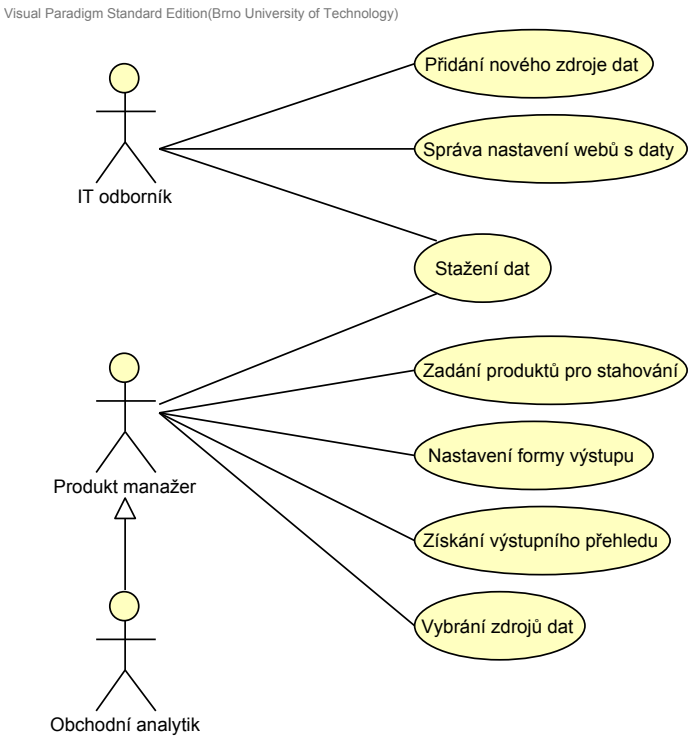
V této části jsou UML diagramy popisující vytvářenou aplikaci. První je diagram případů použití 4.1.1, který ukazuje vnější pohled na vytvářenou aplikaci a definuje způsob, jakým bude aplikace jednotlivými uživateli používána. Dále je zde konceptuální diagram tříd 4.1.2 pro část web robota, který bude stahovat data ze zadaných webů. Poslední je diagram balíčků 4.1.3, který zachycuje celkové rozdělení aplikace na jednotlivé části a jejich vzájemné vazby.

Cílem této kapitoly je pomocí zmíněných UML diagramů vytvořit základní návrh aplikace, který bude využit při její realizaci. Návrh se nesoustředí na úplné detaily, ale spíše popisuje samotný koncept aplikace a dává prostor pro případné úpravy a rozšíření při samotné implementaci.

4.1.1 Diagram případů použití

Diagram případů použití na obrázku 4.1 je vytvořen na základě provedené analýzy v části 2.2. Vystupují v něm celkem tři aktéři. Prvním je IT odborník, který připravuje nastavení programu, přidává nové zdroje dat a případně může data stáhnout. Při stahování dat se předpokládá automatické stahování v určených intervalech. Další aktéři jsou produkt manažer a obchodní analytik, kteří s programem pracují především při určování podoby výstupního přehledu. Původní záměr v části analýzy byl takový, že s programem aktivně pracuje pouze obchodní analytik a poté předává již kompletní analýzy produktovým manažerům. Pro praktické účely je ovšem vhodné počítat v návrhu s tím, že produkt manažeři

[1] http://www.oracle.com/technetwork/java/architecture-142923.html



Obrázek 4.1: Diagram případů použití aplikace

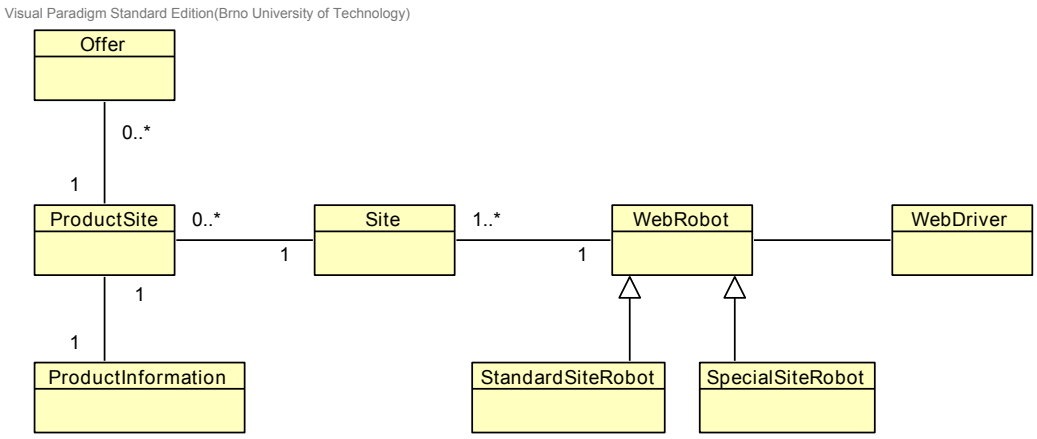
budou moci program přímo používat nezávisle na obchodním analytikovi.

Případ užití s názvem `Správa nastavení webů s daty` je pro zjednodušení diagramu ve formě CRUD akce. Tento případ tedy pokrývá možnosti vytvoření, čtení, změny a smazání.

4.1.2 Konceptuální diagram tříd

Obrázek 4.2 ukazuje konceptuální diagram tříd pro webového robota. Jedná se o část, která provede pouze stažení dat podle zadaného nastavení ze zvolených webů. Nejsou zde zachyceny závislosti na zbyteku aplikace ani konkrétní metody nebo atributy. Cílem tohoto diagramu je připravit rozvržení hlavních tříd při implementaci. Základní popis a význam jednotlivých tříd je v následujícím seznamu:

- `WebRobot`: třída zastřešující jednotlivé implementace webového robota. Pravděpodobně se bude jednat o rozhraní nebo abstraktní třídu.
- `StandardSiteRobot` a `SpecialSiteRobot`: konkretní třídy pro webové roboty. Předpokládá se minimálně implementace zvládající standardní stránky obchodů a stránky agregátorů cenových nabídek.
- `WebDriver`: třída bude určena pro samotnou práci s webem.
- `Site`: třida představující celý web prodejce (např. mall.cz). Budou zde uložena základní nastavení pro práci s daným webem.
- `ProductSite`: v této třídě budou uloženy adresy a nastavení pro konkrétní stránky produktů.



Obrázek 4.2: Konceptuální diagram tříd pro část web robota

- `ProductInformation`: třída bude obsahovat společné údaje o jednotlivých produktech.
- `Offer`: tato třída reprezentuje právě jednu nabídku na daný produkt. Standardně se předpokládá, že pro stránky prodejců, zde bude právě jedna nebo žádná nabídka. U stránek standardních srovnávačů zde může být nabídek více.

4.1.3 Diagram balíčků

Diagram balíčků na obrázku 4.3 zachycuje rozdělení aplikace na jednotlivé části. Hlavní jednotkou je `Controller`, který bude řídit veškerou spolupráci mezi jednotlivými částmi aplikace. Uživatelské rozhraní v balíčku `User Interface` je vnitřně navázáno na třídy z balíčku Java Swing a navázáno je také na metody v `Controller`. S aplikací bude možné pracovat i nezávisle na uživatelském rozhraní při automatickém stahování dat podle předvoleného nastavení.

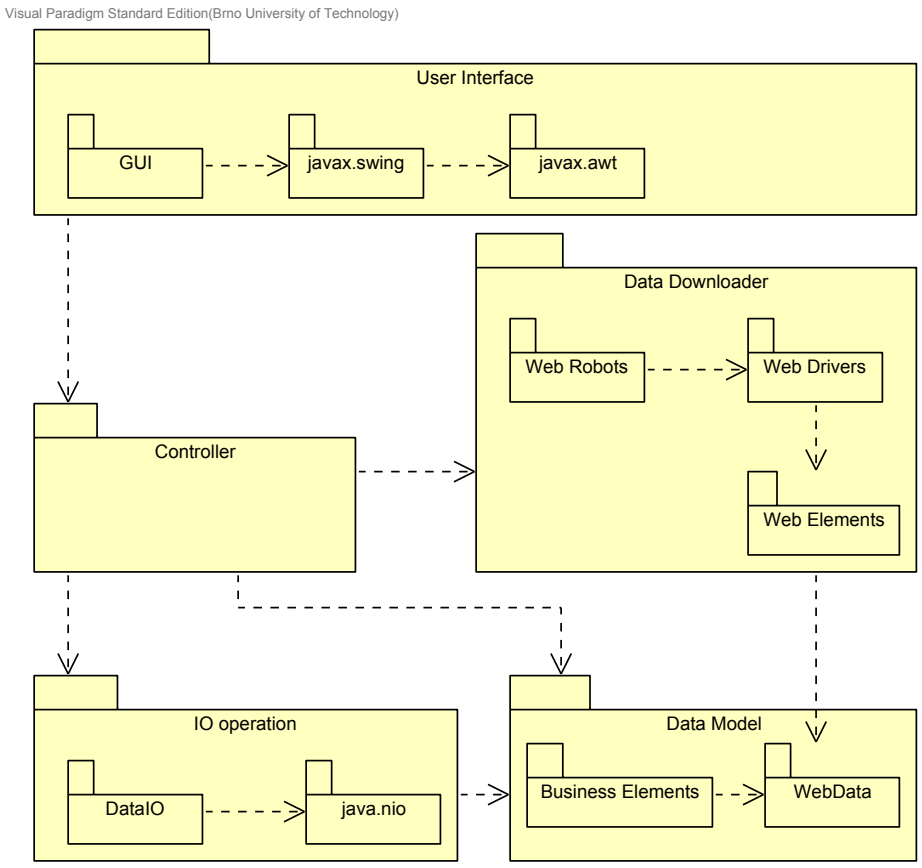
Balíček `Data Downloader` slouží pro část stahování dat pomocí web robota. Jsou zde uloženy jednotlivé části, které realizují procházení webů a stahování dat. Využity zde jsou také dříve zmíněné knihovny HtmlUnit a JSoup.

Poslední balíčky `IO operations` a `Data Model` slouží pro realizaci nízkoúrovňových operací a tříd, které budou sloužit pro předávání dat. Využito bude především standardních knihoven z platformy Java. Pro vytváření výstupů do formátu XLSX a případně dalších budou zvoleny vhodné knihovny, které danou činnost dokáží realizovat.

Z diagramu balíčků je patrné využití návrhového vzoru model-view-controller a snaha maximálně oddělit jednotlivé vrstvy od sebe. Zároveň je oddělena i samotná část pro web robota, aby ji bylo možné využívat i nezávisle.

4.2 Grafické uživatelské rozhraní

Grafické uživatelské rozhraní bude sloužit pro zadávání a zkoušení nastavitelných parametrů aplikace. Při reálném používání se předpokládá automatizované spouštění a ukládání stažených dat do souborů. Využívat ho budou tedy zejména technicky znalí uživatelé, kteří jsou schopni připravit konfiguraci. Nepředpokládá se denní používání ani nutnost přizpůsobit rozhraní pro běžné pracovníky.



Obrázek 4.3: Diagram balíčků pro celou aplikaci

Uživatelské rozhraní bude tedy navrhováno, aby bylo po stránce designu jednoduché a obsahovalo všechny důležité možnosti. Cílem je realizovat uživatelské rozhraní se standardními funkcemi a rozložením, jaké je známé z jiných aplikací. Rozhraní by nemělo obsahovat grafické prvky, které se nevztahují k samotné funkcionalitě.

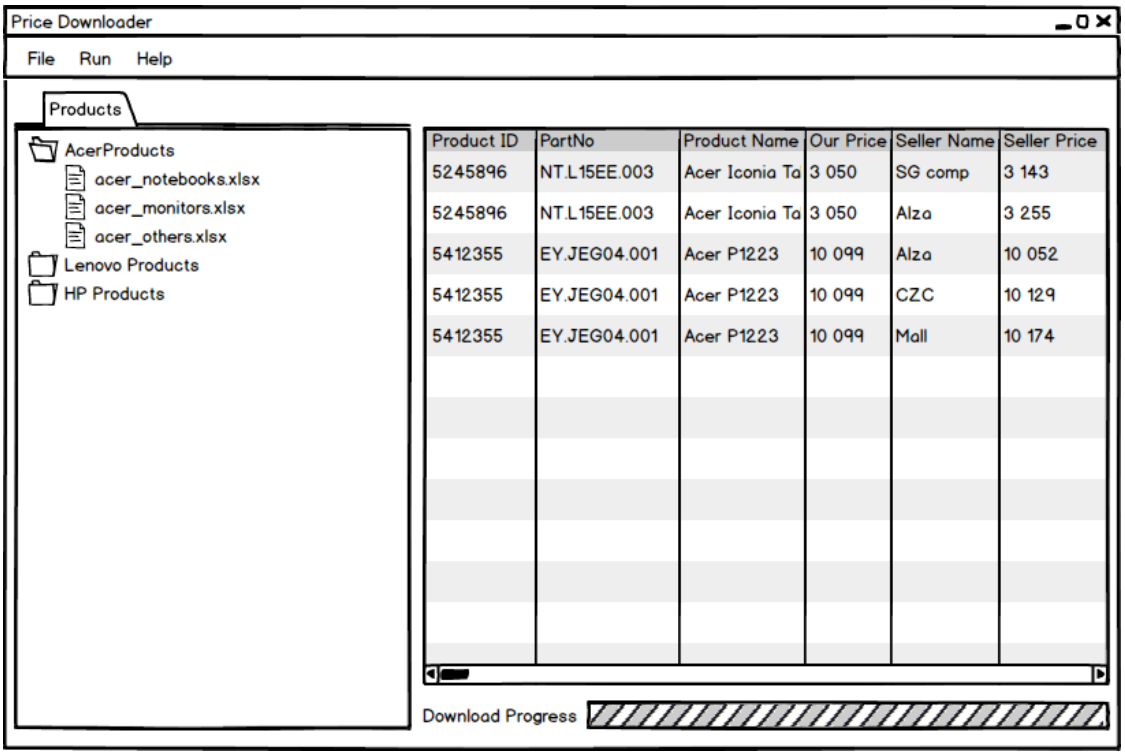
4.2.1 Funkce uživatelského rozhraní

- Otevření vstupních dat a uložení přehledu srovnání cen. Počítá se s podporou formátů jako XLSX a CSV.
- Spuštění stahování dat z webů se zadanou konfigurací.
- Zobrazení výsledků stahování – přehled srovnání cen.

4.2.2 Základní koncept

Na obrázku 4.4 je možné vidět návrh, jak bude přibližně vypadat uživatelské rozhraní. Základní menu programu nabídne standardní funkce jako jiné aplikace podobného typu. Jedná se zejména o možnosti výběru vstupních souborů a ukládání výstupů. V menu bude také možnost spuštění stahování z webů.

Rozložení prvků rozhraní odpovídá standardně používanému způsobu rozložení u aplikací podobného typu. V levé části má uživatel možnost si vybrat soubory, které chce ote-



Obrázek 4.4: Základní návrh grafického uživatelského rozhraní

vřít. V záložce Products bude výběr vstupních dat v podobě souborů s údaji o produktech. V pravé části je poté zobrazená tabulka s daty a ukazatel postupu při stahování.

Jedná se pouze o základní náčrt. Reálná podoba rozhraní se bude odvíjet od nastaveného vzhledu prostředí a možností použitých technologií. Z tohoto důvodu nejsou také řešeny všechny detaily a předpokládá se, že při realizaci se bude postupovat v souladu se standardními postupy při vytváření uživatelských rozhraní pomocí Java Swing.

Kapitola 5

Implementace

Tato kapitola rozebírá samotnou implementaci aplikace pro srovnávání cen. Podle návrhu provedeného v předchozí kapitole je implementace rozdělena do dvou částí. První částí je webový robot, který dokáže získávat data z požadovaných webových stránek. Druhou částí je samotná aplikace využívající webového robota. Ta obsahuje metody pro vstupy a výstupy, objekty pro nastavení ukládané do XML a zajišťuje celou práci s webovým robotem. V rámci aplikace je vytvořeno také uživatelské rozhraní pro možnosti přípravy nastavení a testování jeho správnosti.

UML diagramy v této kapitole jsou tvořeny s cílem ukázat hlavně vazby mezi jednotlivými třídami a rozhraními. V rámci přehlednosti jsou vynechány některé méně podstatné metody a parametry metod. U tříd implementujících rozhraní nejsou metody z rozhraní znovu uváděny, protože třída musí vždy implementovat všechny metody z daného rozhraní.

Kapitola je členěna do čtyř podkapitol, kde první podkapitola 5.1 popisuje použité knihovny a další převzatá řešení. Ve druhé podkapitole 5.2 je rozebírán samotný webový robot a jeho jednotlivé části. Je zde popsán způsob získávání dat, třídy pro nastavení a také třídy produktů, které reprezentují jeden produkt a obsahují všechny zadané odkazy na nabídky. Podkapitola 5.3.2 poté rozebírá řídící část celé aplikace. Popsány jsou zejména realizace čtení a zápisu dat pro zadané formáty, ukládání nastavení a nejdůležitější metody při práci s webovým robotem. Rozebráno je zde také uživatelské rozhraní, které je vytvořeno pomocí technologie Java Swing.

5.1 Použité knihovny

V této kapitole budou popsány knihovny, které byly využity při implementaci aplikace. Jedná se o knihovny pro práci se soubory, s webem a pro tvorbu uživatelského rozhraní. Všechny použité knihovny jsou součástí Java standard edice nebo jsou vydány pod licencí pro svobodný software. Kromě knihoven byla do aplikace převzata i řešení v podobě zdrojových kódů. Všechny takto převzaté zdrojové kódy jsou uvedeny v této kapitole. Ve zdrojových kódech aplikace z této diplomové práce jsou označeny převzaté části komentářem s odkazem na stránku, odkud byly převzaty.

5.1.1 Práce se soubory

Aplikace podporuje vstup i výstup do formátů XLSX a CSV. Pro práci s těmito formáty byly vybrány knihovny Apache POI[1] a Opencsv[2]. Obě knihovny jsou vydány jako svobodný software pod licencí Apache 2.0[3]. Stále mají aktivní vývoj a jsou pro ně vydávány nové aktualizace. Zároveň se jedná o knihovny, které mají za sebou delší historii a je možné předpokládat jejich další vývoj i v budoucnu. Aplikace také pracuje se soubory ve formátu XML, které slouží primárně pro ukládání nastavení případně pro i zadání stahovaných dat.

První zmíněnou knihovnu Apache POI je možné popsat jako programové rozhraní v jazyce Java pro dokumenty z nástrojů Microsoft Office. Knihovna obsahuje podporu pro většinu starších i nových typů dokumentů. Převážně je zaměřená na práci s dokumenty XLS respektive XLSX z programu Excel a dokumenty DOC respektive DOCX z programu Word. Obsahuje také více způsobů, jak s danými soubory pracovat. Varianty práce se soubory se liší především rychlostí zpracování a nároky na paměť. Celkově je knihovna poměrně robustní a obsahuje velké množství připravených tříd i metod, které usnadňují práci s jednotlivými dokumenty.

Druhou knihovnou je Opencsv, která slouží pro práci se soubory ve formátu CSV. Je napsaná pro jazyk Java a svým rozsahem se jedná o menší knihovnu. Obsahuje pouze malý počet tříd a metod, které jsou nutné při zpracovávání CSV souborů. Protože k formátu CSV přistupují jednotlivé aplikace různým způsobem (např. volbou oddělovače) jsou v knihovně připraveny metody, které umožňují nastavit jednotlivé parametry pro vstupní i výstupní soubory. První verze této knihovny se objevily už v roce 2005 a práce na knihovně stále pokračují.

Pro ukládání nastavení o webech a robotech byla využita implementace JAXB[4], která je přímo obsažena ve standardní edici jazyku Java. JAXB obsahuje implementaci převodu objektů z jazyka Java do XML a zpět. V aplikace je tedy využita pro uložení nastavení pro stahování z webů a nastavení web robota. Při práci s JAXB se vycházelo především z oficiálního návodu ze zdroje [14].

5.1.2 Práce s webem

Možné knihovny pro práci s webem byly popsány již v kapitole 3.1. Jednalo se o knihovny Jsoup a HtmlUnit. Při implementaci nenastal případ, kdy bylo nutné nebo výhodné použít knihovnu Jsoup a tedy v rámci aplikace z této diplomové práce byla použita pouze knihovna HtmlUnit.

Druhou knihovnou pro práci s webem je Java WebDriver vytvořený pro nástroj Selenium právě pomocí knihovny HtmlUnit. WebDriver má již vložené nastavení, které je možné upravovat pouze v menším rozsahu. V aplikaci je možné využít standardního klienta pro web nebo právě implementaci od Selenium.

5.1.3 Knihovny pro uživatelské rozhraní

Uživatelské rozhraní aplikace je vytvořeno pomocí Java Swing. K dispozici jsou připravené implementace pro rozložení jednotlivých prvků v uživatelském rozhraní, komponenty, kontejnery, akce a další připravené možnosti pro vytváření uživatelského rozhraní. Java Swing

[1] https://poi.apache.org
[2] http://opencsv.sourceforge.net
[3] http://www.apache.org/licenses/LICENSE-2.0
[4] http://www.oracle.com/technetwork/articles/javase/index-140168.html

také obsahuje implementaci pro práci s vlákny v rámci uživatelského rozhraní. Při práci s komponenty Java Swing se vycházelo především z oficiální dokumentace [15].

V rámci aplikace byly také využity již zdrojové soubory pro DynamicWizardDialog[5]. Jedná se o implementaci dialogů, které v takové podobě Java Swing neposkytuje. Pomocí této implementace je možné realizovat pokročilé dialogy, které nabízejí uživateli postupný výběr v několika obrazovkách dialogu. Druhou využitou komponentou je Java Swing File Browser[6], ve kterém se zobrazují soubory pro zadanou složku. V obou použitých řešení byly provedeny změny pro lepší začlenění do vyvíjené aplikace.

5.1.4 Knihovny pro obecné použití

Další knihovny, které byly při vývoji aplikace použity, jsou Apache Commons[7] a Apache Log4j 2[8]. Obě knihovny jsou vydány pod licencí Apache 2.0. První zmíněná knihovna obsahuje jednotlivé komponenty, které realizují určitou činnost. Případně se jedná o pomocné třídy a metody, které je možné využit např. pro zjištění koncovky souboru atd. Plně využívanou je v rámci aplikace např. komponenta CLI[9], která slouží pro zpracování parametrů při spuštění z příkazové řádky.

Druhá zmíněná knihovna Log4j 2 slouží pro vytváření souborů s výpisy o běhu programu. Na základě výpisů je možné odhalit chyby a odladit celý program. Použita je v rámci této aplikace především pro webového robota.

5.2 Webový robot

Webový robot je v popisované aplikaci samostatná část implementující stahování dat ze zadaných webů. Implementace je provedena s ohledem na možnost jednoduchého rozšiřování a přidávání dalších stránek, ze kterých je možné získávat data. Realizace je provedena pomocí knihovny HtmlUnit, která byla popsána v části 3.1.2, a významné třídy jsou ještě rozebrány v podkapitole 5.2.1.

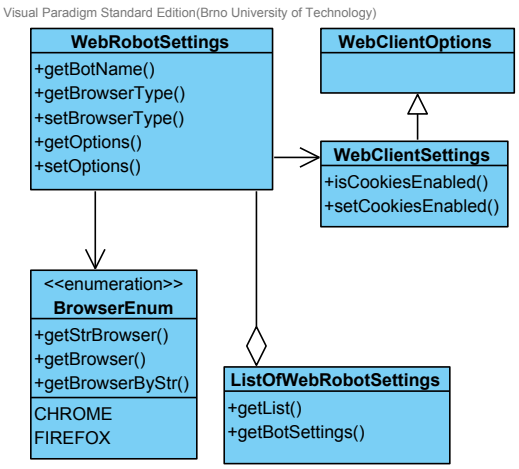
Robot pracuje na principu, kdy přijímá nastavení způsobu stahování i nastavení pro jednotlivé stránky. Stahování probíhá po produktech. Každý produkt obsahuje informace o stránkách, kde se nacházejí nabídky daného produktu. Každý objekt představující informace o stránce obsahuje kromě adresy stránky také seznam XPath výrazů. Pomocí XPath výrazů se extrahují data ze stažené stránky.

5.2.1 Hlavní třídy knihovny HtmlUnit

Použitá knihovna HtmlUnit obsahuje několik významných tříd, které byly použity v rámci vytvoření webového robota. Níže popsané třídy mohou být uvedeny v popisu částí webového robota i v UML diagramech.

`WebClient`
: Nejdůležitější část celé knihovny HtmlUnit. Tato třída slouží přímo pro otevírání webových stránek a emuluje chování zadaného prohlížeče.

[5] http://www.java2s.com/Code/Java/Swing-Components/DynamicWizardDialog.htm
[6] http://java-articles.info/articles/?p=637
[7] https://commons.apache.org
[8] http://logging.apache.org/log4j/2.x
[9] https://commons.apache.org/proper/commons-cli/



Obrázek 5.1: Diagram tříd pro nastavení webového robota

`WebClientOptions`
: Třída pro uložení nastavení pro `WebClient`. Tato třída obsahuje pouze základní nastavení a je pro to nutné ji případně rozšířit o další možnosti.

`HtmlPage`
: Jedná se o jednu z implementací rozhraní `Page`. Tato třída představuje HTML stránku. Poskytuje možnosti pro procházení a vyhledávání HTML elementů na stránce. Podporuje také použití výrazů XPath.

5.2.2 Třídy nastavení webového robota

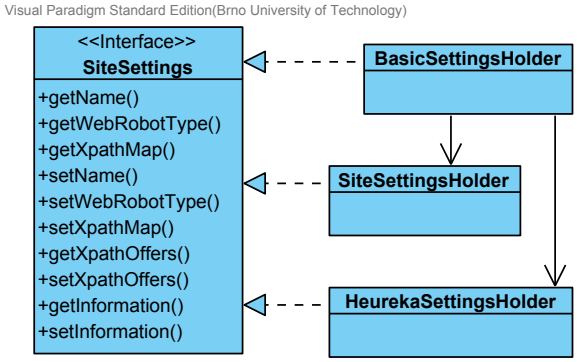
Třídy pro nastavení webového robota je možné vidět na diagramu 5.1. Tyto třídy slouží pro popis, jak se bude robot chovat při stahování dat na dané stránce. Možnosti nastavení vychází především ze třídy `WebClientOptions` z knihovny HtmlUnit, která v sobě nese základní nastavení používaného `WebClienta`.

`WebClientSettings`
: Třída zastřešuje jednotlivé nastavení pro webového robota. Vychází ze třídy `WebClientOptions` a rozšiřuje možnosti o vypnutí a zapnutí používání cookies. Třídu je možné jednoduše rozšířit o další požadované volby pro nastavení.

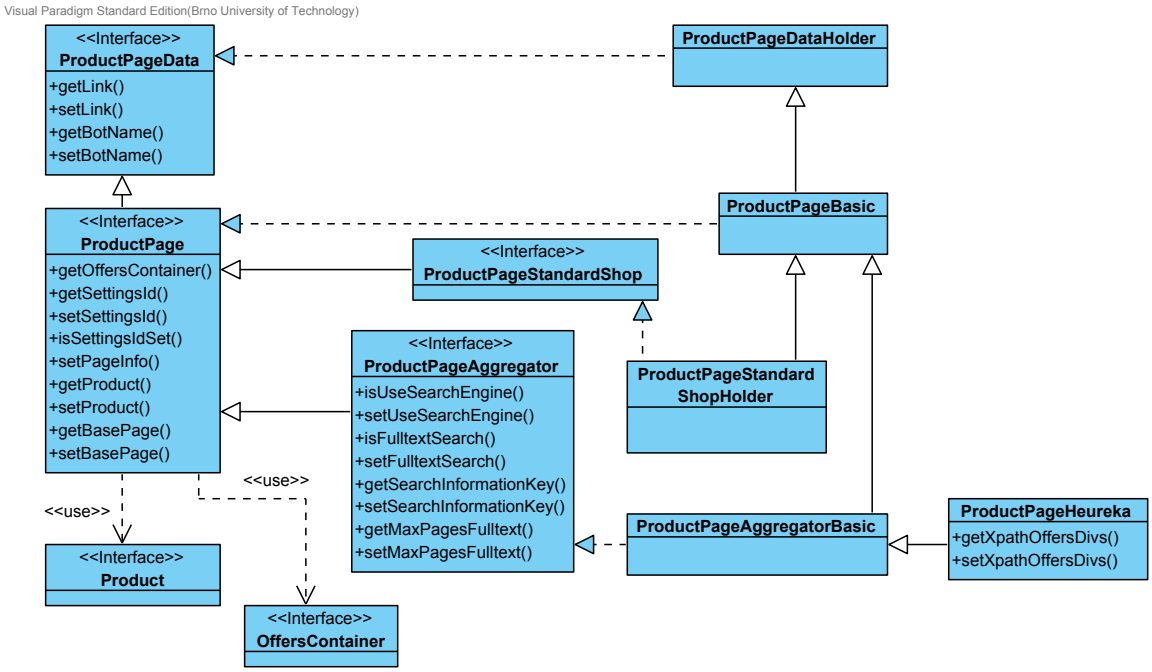
`BrowserEnum`
: Jedná se o výčet prohlížečů, jejichž chování je možné emulovat pomocí knihovny HtmlUnit.

`WebRobotSettings`
: Třída představující nastavení pro robota. Nese v sobě zejména nastavení v podobě instance třídy `WebClientSettings` a také typ prohlížeče z `BrowserEnum`. Každé nastavení je spojené s názvem, který se používá jako identifikátor pro volbu, které nastavení robota chceme při stahování použít.

`ListOfWebRobotsSettings`
: Jedná se o třídu, která v sobě nese seznam vytvořených nastavení pro roboty a poskytuje případně pokročilé operace nad tímto seznamem.



Obrázek 5.2: Diagram tříd pro nastavení stahovaných stránek



Obrázek 5.3: Diagram tříd pro stránky u webového robota

5.2.3 Třídy nastavení stránek

Nastavení stránek slouží pro možnost zadávat informace pro stahování nezávisle na implementaci stránky a označovat údaje, které se z nich mají stáhnout. Pro standardní stránky obchodníků, jejichž struktura odpovídá formátu jedna nabídka na jedné stránce, je možné využití již implementovaných tříd. Pro agregátory cenových nabídek je nutné použit více individuální přístup, a proto je nutné pro ně naprogramovat řešení podle struktury daného webu.

Cílem implementace pro nastavení stránek, zobrazené na diagramu 5.2, bylo dát uživateli maximální možnosti pro individuální nastavení při zachování jednoduchosti. Uživateli je umožněno jak nastavení konkrétních dat pro každou stránku, tak i zadání identifikátoru společného nastavení např. pro všechny stránky na daném webu. Protože každá stránka obsahuje individuální nastavení, není v diagramu 5.3 ani naznačena vazba mezi rozhraními `ProductPage` a `SiteSettings`. O správné aplikování nastavení se stará přímo hlavní třída

webového robota `WebRobot`, která je popsaná v další části.

`SiteSettings`
: Rozhraní, které zastřešuje nastavení pro označení dat, která se mají získávat. Je možné zadat identifikátor nastavení. Nastavení se skládá především z XPath výrazů, zadaného identifikátoru robota a dalších informací, které budou ve výstupu.

`ProductPageData`
: Rozhraní pro uložení základních dat o stránce jako je webová adresa a typ používaného robota.

`ProductPage`
: Základní rozhraní přestavující stránku jednoho produktu. Stránka obsahuje informace o produktu i o stránce na webu. Do `ProductPage` je možné zadat také identifikátor nastavení pro danou stránku.

`ProductPageStandardShop`
: Jedná se o rozhraní, které pod sebe zastřešuje všechny standardní stránky obchodů. Předpokládaná struktura těchto stránek je ve formátu, kde jedna stránka představuje jednu nabídku pro daný produkt.

`ProductPageAggregator`
: Rozhraní pro stránky agregátorů cenových nabídek. Povoluje specifikovat některé volby, které standardně takové portály obsahují. Zároveň rozhraní deklaruje metody pro použití vyhledávání na daném portálu.

Ostatní třídy v diagramu implementují popsaná rozhraní. Pro standardní stránky se využívá třída `ProductPageStandardShopHolder` a pro stránky agregátorů poté konkrétní třídy pro daný agregátor, např. `ProductPageHeureka`.

5.2.4 Třídy produktů a nabídek

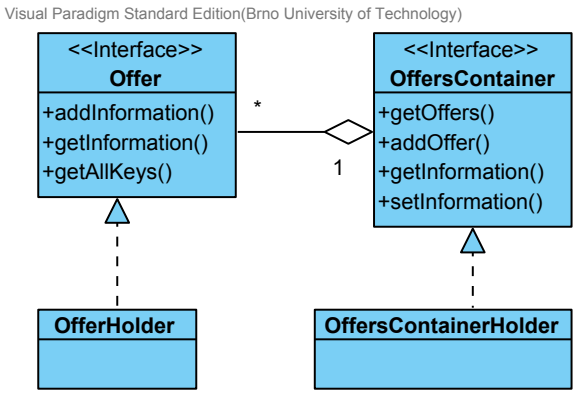
V této části jsou popsány třídy, které se starají o informace o nabídkách a produktech. Každá nabídka nese vlastní doplňující informace. Třídy pro nabídky zobrazené v diagramu 5.4 jsou popsány v následujícím seznamu.

`Offer`
: Rozhraní reprezentující jednu nabídku. Může obsahovat libovolné informace v podobě klíče a hodnoty. Nejčastěji se jedná o údaje jako je cena, jméno produktu uvedené na stránce nabídky, skladová dostupnost atd.

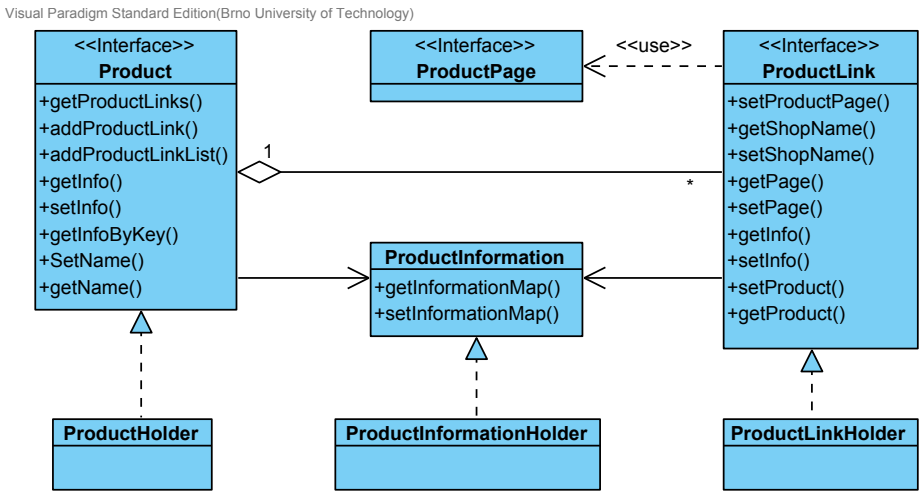
`OffersContainer`
: Rozhraní reprezentující skupinu nabídek, které pocházejí ze stejného zdroje. Do OffersContainer je možné zadat společné informace všem nabídkám.

Druhou část tvoří třídy produktů zobrazené v diagramu 5.5. Ty jsou určené pro držení popisu produktu a všech odkazů nabídek, které se k produktu stahují. Každý produkt obsahuje odkazy na stránky nabídek v podobě objektu třídy ProductLink. Informace o produktu je možné přidat jak individuálně pro jednotlivé odkazy, tak i pro produkt jako celek.

`ProductInformation`
: Třída pro udržování informací o produktu.



Obrázek 5.4: Diagram tříd reprezentující nabídky z webů



Obrázek 5.5: Diagram tříd reprezentující produkty

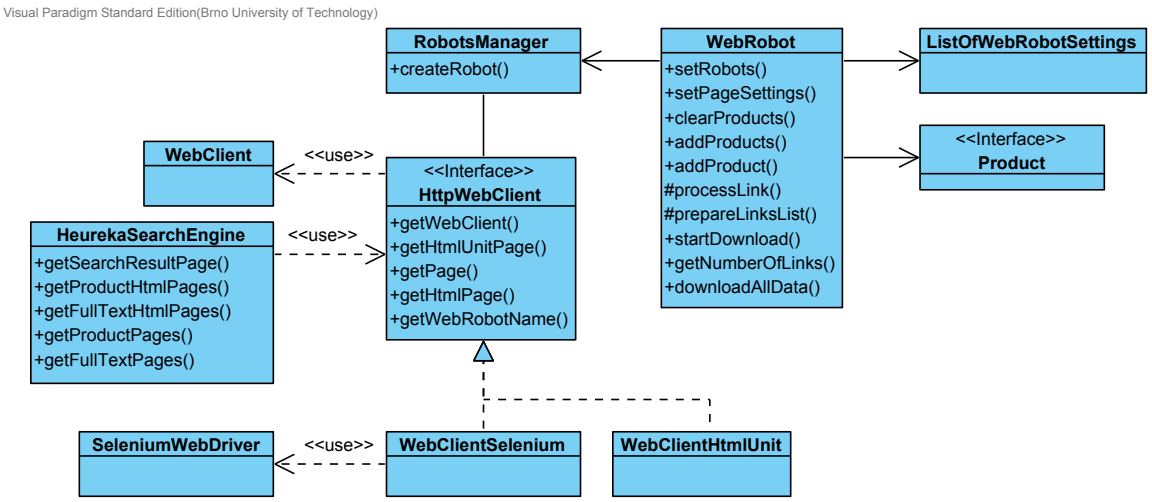
`ProductLink`
: Rozhraní představující právě jeden odkaz na nabídku produktu. Nese v sobě informace o obchodu a stránce, kde se nabídka nachází.

`Product`
: hlavní třída pro udržování informací o produktu. Jsou v ní obsažené všechny odkazy na produkty a společné informace o produktu. `Product` se používá jako hlavní nositel informací o datech, která se mají stahovat. Webový robot přijímá právě instance tříd implementující rozhraní `Product`, protože obsahují všechny informace pro stažení dat.

5.2.5 Hlavní třídy robota

V této části jsou rozebrány hlavní třídy webového robota. V UML diagramu 5.6 je možné vidět i vazby na dříve popsané třídy a rozhraní jako je `Product` nebo `WebClient`.

`RobotsManager`
: Třída pro vytvoření a správu jednotlivých objektů s rozhraním `HttpWebClient`. Poskytuje možnost pro ušetření zdrojů a znuvupoužitelnost již existujících objektů v rámci běhu aplikace.



Obrázek 5.6: Diagram hlavních tříd webového robota

`HeurekaSearchEngine`
: Jedná se o třídu, která implementuje možnost použití vyhledávání na serveru heureka. Vlastní třídy pro jednotlivé agregátory cenových nabídek jsou nutné z důvodu odlišných struktur stránek.

`WebRobot`
: Hlavní třída zastřešující webového robota. Jedná se o třídu, kterou budou primárně využívat aplikace při práci s robotem. Předpokládá se, že instanci této třídy budou předána nastavení, která se mají použít při získávání dat. Poté jí budou předány objekty třídy `Product`, které obsahují všechny potřebné údaje pro stažení informací z webů. Třída podporuje kromě celkového stažení všech dat najednou i možnost získávat výsledky průběžně.

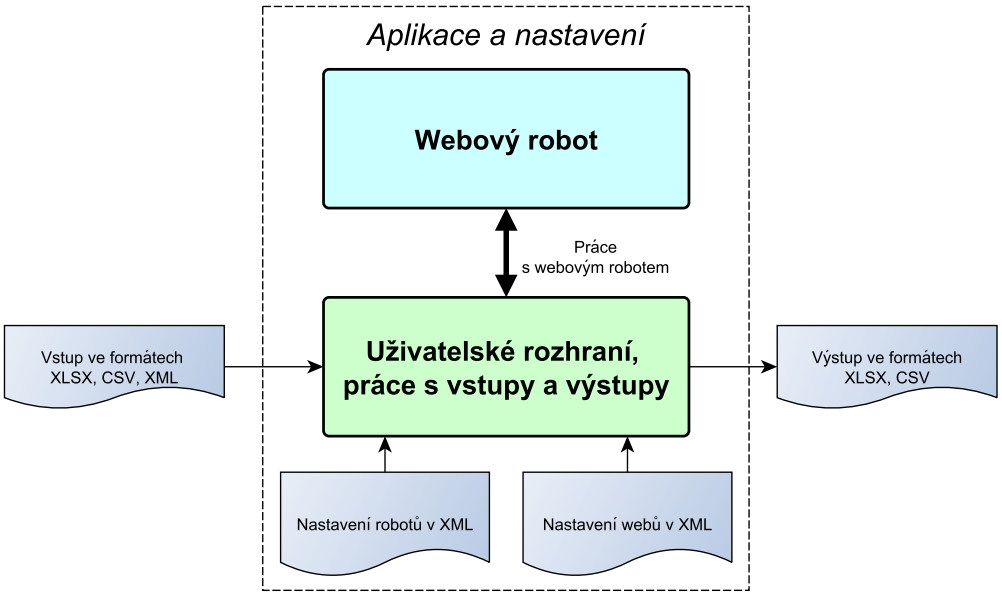
5.3 Řídící část aplikace

Druhou částí vytvářené aplikace je řidící část a uživatelské rozhraní. Řešená je zde především část implementace spojená s ovládáním aplikace a možnostmi pro vstupy a výstupy. Na obrázeku 5.7 je přiblížen způsob, jak aplikace funguje. Celé propojení jednotlivých částí zajišťuje třída `WebDownloaderController`, přes kterou dochází k předávání dat.

5.3.1 Systém pro nastavení

Jednou z předností popisované aplikace jsou možnosti nastavení. Nastavení je uloženo v samostatných souborech ve formátu XML. Možnosti nastavení webů a produktů částečně vychází z implementace tříd webového robota popsaného v části 5.2.2. Pro systém nastavení jsou však vytvořeny nezávislé třídy, které obsahují potřebné anotace elementů z JAXB. Již vytvořené třídy z části webového robota nebyly využity z důvodu lepšího rozdělení implementace do jednotlivých částí. Další výhodou zvoleného řešení je možnost rozšiřovat nastavení bez nutnosti změn zdrojového kódu webového robota.

Následuje popis pro podstatné třídy, které nejvíce reprezentují možnosti nastavení.



Obrázek 5.7: Způsob fungování aplikace

`XmlSummary`
: Jedná se o třídu pro uložení produktů, které chceme sledovat. Třída obsahuje seznam zadaných odkazů na XML soubory, které představují třídu `WebRobotXmlProduct`. Tento soubor může tedy obecně obsahovat libovolný počet produktů, pro které chceme stahovat data.

`WebRobotXmlProduct`
: Třída obsahující údaje pro převedení do dříve popsané třídy `Product`. Obsahuje seznam tříd `WebRobotXmlProductLink` a společné informace o produktu. Každý produkt tedy může obsahovat libovolné množství odkazů na různé weby obchodníků.

`WebRobotXmlProductLink`
: Třída představující jeden odkaz na stránku zadaného produktu. Nese základní informace podobně jako dříve popsaná třída `ProductLink`. Je svázaná právě s jednou stránkou v podobě třídy `WebRobotXmlProductPage`.

`WebRobotXmlProductPage`
: Nastavení pro danou stránku podobně jako `ProductPage`. Jsou zde uloženy veškeré informace pro získání dat z odkazované stránky. Nastavení je zde možné zadávat přímo nebo formou identifikátoru nastavení pro daný web.

Uživatel má tedy možnost vytvořit si nezávislé XML soubory s informacemi a odkazy pro sledovaný produkt. Poté si vytvoří jednotlivé vstupní soubory, kde jsou uvedeny pouze odkazy na soubory s produkty. Tento způsob uživateli umožňuje vytvoření více seznamů produktů (např. podle výrobce, typu, výše ceny, ...) bez nutnosti znovu zadávat informace pro každý produkt.

Nastavení robotů je uloženo v samostatném XML souboru, jehož strukturu reprezentuje třída `WebRobotXmlBotsSettings`. Další třída `WebRobotXmlSettings` zase reprezentuje

nastavení pro weby. Ukázky souborů s nastavením pro roboty, weby a vstupní data jsou uvedeny v příloze C.

5.3.2 Stahování z webů

Stahovaní dat je prováděno v nezávislém vláknu řešeném přes standardní třídu `SwingWorker`[10]. Ta umožňuje průběžně předávat výsledky do vlákna pro uživatelské rozhraní. Tabulka s daty je tedy aktualizována po každém zpracovaném produktu a uživatel vidí stažená data okamžitě. Zároveň jsou zde i předávány informace pro zobrazování průběhu stahování.

V případě, že je aplikace zapnutá pouze pro jednorázové získávání dat, výše zmíněné činnosti nejsou potřebné. Není tedy zobrazené uživatelské rozhraní a není potřebná indikace průběhu stahování a ani postupné zobrazování dat. Využijí se metody pro přímé stahování, aby bylo dosaženo maximální rychlosti aplikace.

Následuje popis některých podstatných tříd pro stahování dat v řídící části.

`WebDownloaderBot`
: Nástavba nad třídou `WebRobot`. Rozšiřuje původní třídu hlavně o metody pro získávání stahovaných dat postupně produkt po produktu.

`DownloadWorker`
: Třída, která realizuje stahování ve vlákně nezávislém na uživatelském rozhraní. Jedná se o rozšíření třídy `SwingWorker`. Jsou zde pravidelně předávány informace o průběhu stahování zpět do uživatelského rozhraní.

`WebDownloader`
: Hlavní třída. Zde je umístěn vstupní bod při startu aplikace. Vyhodnocuje se zejména, zda dojde ke spuštění uživatelského rozhraní nebo zda bude provedeno stahování podle zadaných parametrů.

5.3.3 Import a export dat produktů

Importovat je možné data ve formátech XLSX, CSV a XML. V případě XML se jedná o dříve popsanou strukturu, která reprezentuje třídu `XmlSummary`. Soubory ve formátu XLSX i CSV jsou požadovány ve struktuře, kdy první čtyři sloupce představují: odkaz na stránku s produktem, identifikátor nastavení pro daný web, název obchodu a označení produktu (název, part number případně interní identifikátor). Další sloupce v souboru se poté berou jako společné informace o produktu (např. nákupní a prodejní cena, měna).

Vstupy v podobě XLSX a CSV obsahují často stejné údaje zadané opakovaně. Tomuto problému se nelze v případě použití těchto formátů jednoduše vyhnout. Při implementaci byla upřednostněna jednoduchá struktura zadávaných souborů i za cenu redundance některých dat. V případě využití vstupních souborů ve formátů XML je možné mít společné údaje zapsané jen na jednom místě. V následujících třídách je implementace čtení, zápisu a převodu vstupních dat do požadované podoby v aplikaci.

`ExcelIO, CsvIO, WebRobotXmlFileUtils`
: Třídy realizují samotné čtení a zápis do formátu XLSX, CSV a XML.

`TextTableFileData`
: Třída, která převádí tabulkovou strukturu vstupních dat (formáty XLSX, CSV) do jednotlivých objektů webového robota.

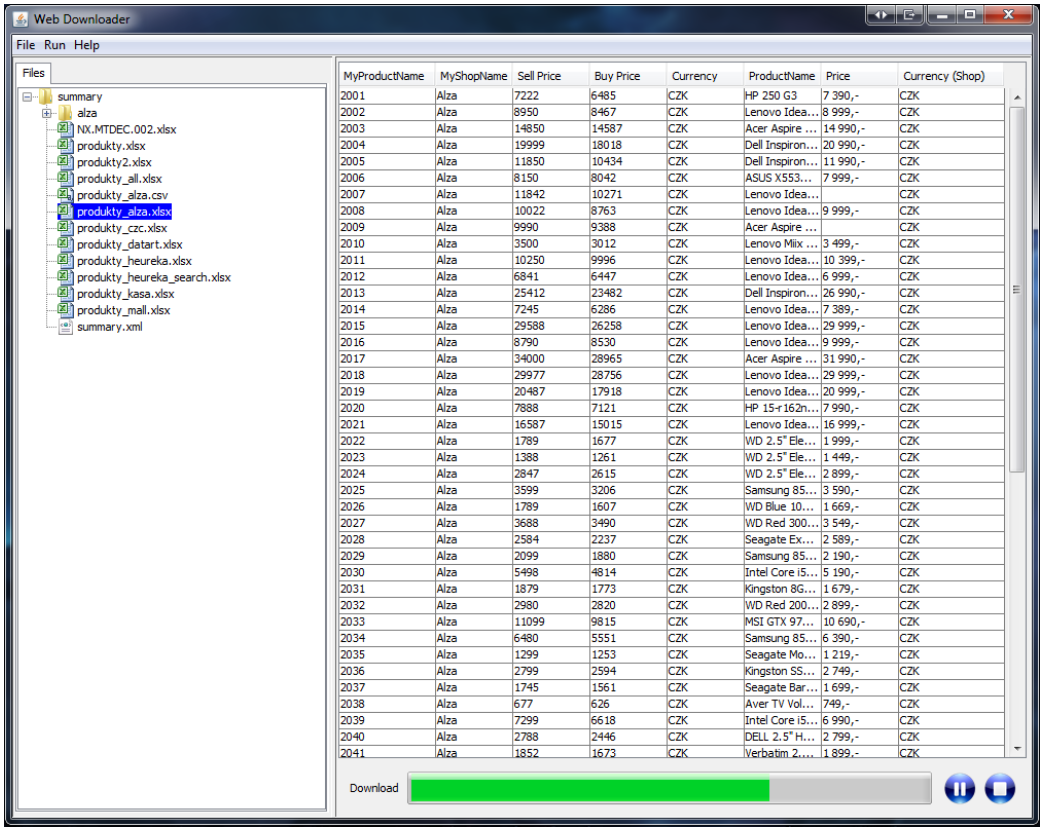
[10] http://docs.oracle.com/javase/7/docs/api/javax/swing/SwingWorker.html

5.3.4 Uživatelské rozhraní

Uživatelské rozhraní v aplikaci slouží pro testování vytvořeného nastavení pro jednotlivé weby obchodníků. Při tvorbě se vycházelo z provedeného návrhu v kapitole 4.2. Na obrázku 5.8 je poté možné vidět výslednou podobu uživatelského rozhraní. Hlavní třídy jsou popsány v následujícím seznamu:

`ActionManager`
: Třída, která obsahuje všechny implementované akce pro uživatelské rozhraní v aplikaci. Jedná se o akce jako otevření vstupního souboru, uložení dat do výstupního souboru, zahájení stahování, generování produktů, ... V aplikaci je pouze jedna instance této třídy, která umožňuje přistupovat k jednotlivým akcím z libovolné části programu.

`WebDownloaderMainFrame`
: Třída představující hlavní okno uživatelského rozhraní. Je zde implementace pro umístění jednotlivých komponent rozhraní.



Obrázek 5.8: Ukázka aplikace při stahování dat

Kapitola 6

Statistiky a přehledy

Tato kapitola se zabývá zpracováním získaných dat o cenách. Jsou zde popsány některé analýzy, které lze nad daty provádět. Pro vytváření analýz byl zvolen program Microsoft Excel. Důvody volby zmíněného nástroje a jeho výhody a případné omezení jsou uvedeny v kapitole 6.1.

Vytvořené analýzy můžeme rozdělit na časově závislé a časově nezávislé. U časově nezávislých analýz využíváme pouze jeden soubor dat při provádění analýzy. Z praktického pohledu můžeme takovou analýzu provést vždy nad nejaktuálnějšími daty bez potřeby jakýchkoliv předchozích výsledků. Časově nezávislé analýzy jsou rozebrány v podkapitole 6.2.

Druhým typem analýz popsaných v této kapitole jsou časově závislé analýzy. U tohoto typu analýz potřebujeme pracovat s vícero datovými výstupy získanými v průběhu určitého období. Jedná se např. o sledování vývoje ceny u určitého produktu po dobu několika měsíců. Časově závislé analýzy jsou rozebrány v podkapitole 6.3 a 6.4.

Všechny popsané analýzy v této kapitole vycházejí pouze z údajů získaných z aplikace realizované v rámci této diplomové práci. Pro časově závislé analýzy je rovněž využita znalost datumů stažení jednotlivých přehledů. U popsaných analýz může být potřebné získaná data o cenách převést do určité formy, která je použitelná při dalším zpracování v programu Excel.

Cílem popsaných analýz v této kapitole je dát více informací lidem, kteří musí rozhodovat o nastavení cen u svých produktů. Důležitá je tedy jednoduchá forma nejlépe s výsledky v podobě tabulek a grafů. Platí ovšem, že při rozhodování vždy bude záležet na konkrétní situaci a nelze stanovovat nové ceny zcela automaticky.

6.1 Volba nástrojů

Uvažovalo se o dvou způsobech realizace analýz. Prvním uvažovaným způsobem bylo umístit analýzy přímo do aplikace a v uživatelském rozhraní ukazovat jejich výsledky. Výhodou by bylo především možnost získat analýzy bez nutnosti použití dalších externích nástrojů. Zároveň by byla značně zjednodušená práce s daty při vytváření analýz. Nebylo by potřeba data převádět do jiných formátů a upravovat do použitelné podoby pro externí aplikaci.

Nevýhodou jsou zejména malé možnosti změn a úprav analýz, které by aplikace uživateli poskytovala. Změny a další analýzy by se musely vždy doprogramovat. Pro reálné použití by aplikace potřebovala mít připravené velké množství analýz s možností zadat parametry. Případně by aplikace musela nabízet možnost kompletního vytvoření analýzy vlastní. Náročnost tvorby takového řešení je však nad rámec této diplomové práce, která se zabývá

především získáváním daných dat.

Druhou možností je použití externího programu, ve kterém lze analýzy provádět. Realizovaná aplikace tedy slouží čistě pro stahování dat z webu a dává uživateli možnost s takto staženými daty dále pracovat. Nevýhodou je menší komfort, kdy uživatel musí vlastnit daný externí program. Navíc je nutné vyřešit úpravu dat do tvaru, se kterým je možné dále pracovat. Zvláště v případě časově závislých analýz, kdy budou potřebná data uložena ve více souborech podle datumu jejich stažení.

Pro realizaci byl tedy zvolen program Microsoft Excel, který se běžně používá pro analýzu dat. Výhodou Excelu je především velká uživatelská základna. Poskytuje množství funkcí pro práci s daty včetně možnosti předzpracování dat pomocí maker[1]. Zároveň obsahuje nástroje pro práci s daty v tabulkách a vytvoření přehledných grafů. Více o možnostech programu Excel je rozebráno v následující části 6.1.1.

6.1.1 Možnosti programu Excel pro analýzy

Způsob použití a základní možnosti Excelu jsou obecně známé. V této části budou rozebrány spíše pokročilejší možnosti, které lze využít při zpracování dat o cenách. V následujím seznamu jsou tedy uvedeny možnosti, které nabízí Excel 2010 pro analýzy a zpracování dat.

- **Datová spojení**:
  Excel umožňuje vytvořit datové spojení na soubor obsahující data v podporovaných formátech nebo přímo na databázi. Díky tomu je možné automaticky načítat vstupní data pro analýzy.

- **Podmíněné formátování**:
  Jedná se o funkci, která umožňuje zvýraznit buňky nebo celé řádky podle zadané podmínky. Je možné tímto způsobem upozornit na konkrétní produkty, které (ne)splňují určitá kritéria, která jsme zvolili.

- **Vlastní funkce**:
  Uživatel má možnost si vytvořit vlastní funkci, se kterou může později pracovat. Tato možnost je především důležitá z důvodu, že standardní funkce v Excelu již požadují data v určité podobě. Díky vlastním funkcím je tedy možné parametry předzpracovat bez nutnosti složitých konstrukcí.

- **Standardní a kontingenční tabulky**:
  Pokročilá podpora práce s tabulkami je jedním z rysů programu Excel. Standardní tabulky značně usnadňují práci s daty, ale pro soubory s velkým množstvím dat mohou být nepřehledné. Excel ovšem disponuje i kontingenční tabulkou, díky které je možné i z velkého množství neuspořádaných dat udělat přehledný výstup. Kontingenční tabulky se zejména hodí pro zobrazení dat z agregátorů cenových nabídek.

- **Makra**:
  Makra dovolují psaní funkcí a procedur ve VBA[2]. Je možné napsat vlastní způsob zpracování dat a realizovat celý výpočet analýzy. Pomocí VBA se také programují výše popsané vlastní funkce, které lze později použít v tabulce s daty.

[1] https://support.office.com/en-NZ/article/introduction-to-macros-a39c2a26-e745-4957-8d06-89e0b435aac3

[2] http://en.wikipedia.org/wiki/Visual_Basic_for_Applications

6.2 Časově nezávislé analýzy

Jedná se o jednoduché analýzy, které je možné použít na aktuální soubor s daty o cenách. Nepotřebujeme údaje z delšího časového období. V části 6.2.1 je popsána jednoduchá analýza založená na ohodnocující funkci.

Pro data z agregátorů cenových nabídek je využito kontingenční tabulky, která dovoluje realizovat pokročilé filtrování a podat vícero pohledů na data. Popis způsobu vytvoření takové tabulky je v části 6.2.2.

6.2.1 Konkurenceschopnost produktů

	A	B	C	D	E	F	G	H
1	MyProductName	ProductName	MyShopName	Sell Price	Buy Price	Price	Ranking	
2	2001	HP 250 G3	Alza	7222	6485	7 390,-	3	
3	2002	Lenovo IdeaPad G50-30 Black	Alza	8950	8467	8 999,-	3	
4	2003	Acer Aspire E15 Midnight Black	Alza	14850	14587	15 990,-	4	
5	2004	Dell Inspiron 13z Touch	Alza	19999	18018	20 990,-	3	
6	2005	Dell Inspiron 11z Touch	Alza	11850	10434	11 990,-	3	
7	2006	ASUS X553MA-SX376H černý	Alza	8150	8042	7 999,-	1	
8	2007	Lenovo IdeaPad G500 Black	Alza	11842	10271		-1	
9	2008	Lenovo IdeaPad G50-45 Black	Alza	10022	8763	9 999,-	2	
10	2009	Acer Aspire E15 Garnet Red	Alza	9990	9388	9 990,-	3	
11	2010	Lenovo Miix 3 8 Black 32GB	Alza	3500	3012	3 499,-	2	
12	2011	Lenovo IdeaPad B5400	Alza	10250	9996	10 399,-	3	
13	2012	Lenovo IdeaPad G50-30 Black	Alza	6841	6447	6 999,-	3	
14	2013	Dell Inspiron 13z Touch	Alza	25412	23482	26 990,-	4	
15	2014	Lenovo IdeaPad G50-30 Black	Alza	7245	6286	7 389,-	3	
16	2015	Lenovo IdeaPad Y50-70 Black	Alza	29588	26258	29 999,-	3	
17	2016	Lenovo IdeaPad G50-30 Black	Alza	8790	8530	9 999,-	4	
18	2017	Acer Aspire V17 Nitro Black Edition	Alza	34000	28965	31 990,-	2	
19	2018	Lenovo IdeaPad Y50-70 Black	Alza	29977	28756	29 999,-	3	
20	2019	Lenovo IdeaPad Z50-70 Black	Alza	20487	17918	20 999,-	3	
21	2020	HP 15-r162nc Flyer Red	Alza	7888	7121	7 990,-	3	
22	2021	Lenovo IdeaPad Yoga 2 13 Silver	Alza	16587	15015	16 999,-	3	
23	2022	WD 2.5" Elements Portable 1000GB černý	Alza	1789	1677	1 999,-	4	
24	2023	WD 2.5" Elements Portable 500GB černý	Alza	1388	1261	1 499,-	4	
25	2024	WD 2.5" Elements Portable 2000GB černý	Alza	2847	2615	2 899,-	3	
26	2025	Samsung 850 EVO 250GB	Alza	3599	3206	3 590,-	2	
27	2026	WD Blue 1000GB 64MB cache	Alza	1789	1607	1 669,-	2	
28	2027	WD Red 3000GB 64MB cache	Alza	3688	3490	3 549,-	2	
29	2028	Seagate Expansion Portable 2000GB	Alza	2584	2237	2 589,-	3	
30	2029	Samsung 850 EVO 120GB	Alza	2099	1880	2 190,-	3	
31	2030	Intel Core i5-4460	Alza	5498	4814	5 190,-	2	
32	2031	Kingston 8GB KIT DDR3 1600MHz CL10 HyperX Fury Blue Series	Alza	1879	1773	1 679,-	1	
33	2032	WD Red 2000GB 64MB cache	Alza	2980	2820	2 899,-	2	
34	2033	MSI GTX 970 GAMING 4G	Alza	11099	9815	10 690,-	2	
35	2034	Samsung 850 EVO 500GB	Alza	6480	5551	6 390,-	2	
36	2035	Seagate Momentus SpinPoint M8 500GB	Alza	1299	1253	1 219,-	1	
37	2036	Kingston SSDNow V300 240GB 7mm	Alza	2799	2594	2 749,-	2	
38	2037	Seagate Barracuda 7200.14 1000GB s Advanced Format	Alza	1745	1561	1 699,-	2	
39	2038	Aver TV Volar HD	Alza	677	626	749,-	4	
40	2039	Intel Core i5-4690K	Alza	7299	6618	6 990,-	2	
41	2040	DELL 2.5" HDD 2TB černý	Alza	2788	2446	2 799,-	3	

Přehled

Obrázek 6.1: Ukázka provedené analýzy pro ohodnocení ceny produktů

Cílem analýzy popsané v této části je nalézt produkty, kde naše cena není konkurenceschopná. Produkty, pro které nebyla nalezena cena (např. produkt se přestal prodávat), do zpracování nebereme, ale ponecháme je zobrazené v seznamu. Pro uživatele je vhodné, aby problematické produkty byly barevně odlišené od zbytku.

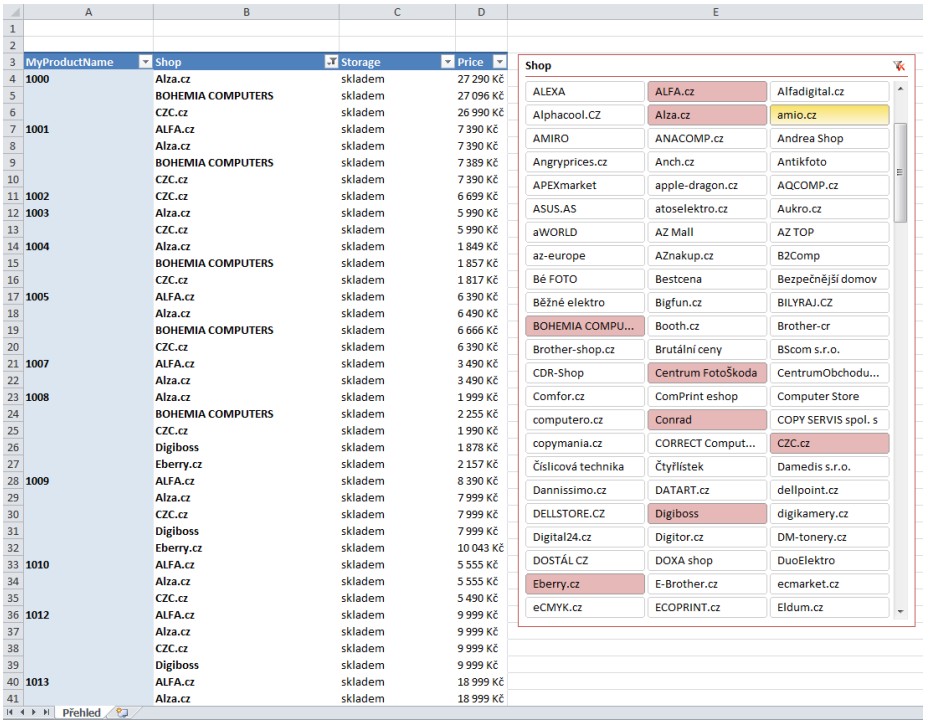
Pro realizaci je vhodné vytvořit ohodnocující funkci, která podle zadaných parametrů ohodnotí danou nabídku. Jakým způsobem bude funkce ohodnocení provádět, je většinou závislé na konkrétní situaci v dané firmě. Je však možné sestavit jednoduchou ohodnocující funkci, která porovnává naše ceny s konkurencí. Cílem je nalézt především produkty, které prodáváme za vyšší cenu než konkurence, ale máme určitý prostor pro úpravu ceny. Druhým typem hledaných produktů jsou ty, kde nemůžeme být beze změny ceny u dodavatele konkurenceschopní.

Ohodnucující funkce byla v tomto případě realizována pomocí odstupňovaného hodnocení 1 až 5, kde 1 je nejhorší a 5 nejlepší hodnocení. Funkce hodnotí podle následujících kriterií:

- **Hodnocení 5**: Prodejní cena konkurence je o více jak 15% vyšší než naše prodejní cena.
- **Hodnocení 4**: Prodejní cena konkurence je o více jak 5% vyšší než naše prodejní cena.
- **Hodnocení 3**: Prodejní cena konkurence je vyšší než naše prodejní cena nebo je jí rovna.
- **Hodnocení 2**: Prodejní cena konkurence je nižší než naše prodejní cena a zároveň vyšší než naše nákupní cena.
- **Hodnocení 1**: Prodejní cena konkurence je nižší než naše nákupní cena.
- **Hodnocení -1**: Produkt nemá přiřazenou cenu.

Možná podoba výsledku je na obrázku 6.1. Tabulka poskytuje možnost filtrace podle výsledků ohodnocující funkce. Navíc jsou všechny produkty, které mají oproti konkurenci vyšší cenu (hodnocení 1 a 2) automaticky barevně odlišeny odstínem červené a žluté barvy. Naopak produkty, kde máme velký prostor pro zvýšení ceny (hodnocení 5) jsou zvýrazněny zeleně.

6.2.2 Kontingenční tabulka nad daty ze srovnávačů



MyProductName	Shop	Storage	Price
1000	Alza.cz	skladem	27 290 Kč
	BOHEMIA COMPUTERS	skladem	27 096 Kč
	CZC.cz	skladem	26 990 Kč
1001	ALFA.cz	skladem	7 390 Kč
	Alza.cz	skladem	7 390 Kč
	BOHEMIA COMPUTERS	skladem	7 389 Kč
	CZC.cz	skladem	7 390 Kč
1002	CZC.cz	skladem	6 699 Kč
1003	Alza.cz	skladem	5 990 Kč
	CZC.cz	skladem	5 990 Kč
1004	Alza.cz	skladem	1 849 Kč
	BOHEMIA COMPUTERS	skladem	1 857 Kč
	CZC.cz	skladem	1 817 Kč
1005	ALFA.cz	skladem	6 390 Kč
	Alza.cz	skladem	6 490 Kč
	BOHEMIA COMPUTERS	skladem	6 666 Kč
	CZC.cz	skladem	6 390 Kč
1007	ALFA.cz	skladem	3 490 Kč
	Alza.cz	skladem	3 490 Kč
1008	Alza.cz	skladem	1 999 Kč
	BOHEMIA COMPUTERS	skladem	2 255 Kč
	CZC.cz	skladem	1 990 Kč
	Digiboss	skladem	1 878 Kč
	Eberry.cz	skladem	2 157 Kč
1009	ALFA.cz	skladem	8 390 Kč
	Alza.cz	skladem	7 999 Kč
	CZC.cz	skladem	7 999 Kč
	Digiboss	skladem	7 999 Kč
	Eberry.cz	skladem	10 043 Kč
1010	ALFA.cz	skladem	5 555 Kč
	Alza.cz	skladem	5 555 Kč
	CZC.cz	skladem	5 490 Kč
1012	ALFA.cz	skladem	9 999 Kč
	Alza.cz	skladem	9 999 Kč
	CZC.cz	skladem	9 999 Kč
	Digiboss	skladem	9 999 Kč
1013	ALFA.cz	skladem	18 999 Kč
	Alza.cz	skladem	18 999 Kč

Shop

ALEXA	ALFA.cz	Alfadigital.cz
Alphacool.CZ	Alza.cz	amio.cz
AMIRO	ANACOMP.cz	Andrea Shop
Angryprices.cz	Anch.cz	Antikfoto
APEXmarket	apple-dragon.cz	AQCOMP.cz
ASUS.AS	atoselektro.cz	Aukro.cz
aWORLD	AZ Mall	AZ TOP
az-europe	AZnakup.cz	B2Comp
Bé FOTO	Bestcena	Bezpečnější domov
Běžné elektro	Bigfun.cz	BILYRAJ.CZ
BOHEMIA COMPU...	Booth.cz	Brother-cr
Brother-shop.cz	Brutální ceny	BScom s.r.o.
CDR-Shop	Centrum FotoŠkoda	CentrumObchodu...
Comfor.cz	ComPrint eshop	Computer Store
computero.cz	Conrad	COPY SERVIS spol. s
copymania.cz	CORRECT Comput...	CZC.cz
Číslicová technika	Čtyřlístek	Damedis s.r.o.
Dannissimo.cz	DATART.cz	dellpoint.cz
DELLSTORE.CZ	Digiboss	digikamery.cz
Digital24.cz	Digitor.cz	DM-tonery.cz
DOSTÁL CZ	DOXA shop	DuoElektro
Eberry.cz	E-Brother.cz	ecmarket.cz
eCMYK.cz	ECOPRINT.cz	Eldum.cz

Obrázek 6.2: Ukázka kontigenční tabulky pro data z webu Heureka

Data o cenách získaná z webu Heureka obsahují standardně desítky nabídek pro jeden produkt. Při použití běžné tabulky by zobrazení dat bylo poměrně nepřehledné. Pro taková

data je velice výhodné použití kontingenční tabulky[3]. U té má uživatel možnost zvolit, jaké údaje chce zobrazovat v tabulce a jak má být tabulka strukturovaná. Zároveň je možné určit, ze kterých údajů se mají udělat souhrny a které údaje se mají použít pouze pro filtrování. Navíc je možné v rámci kontingenční tabulky přidávat průřezy.

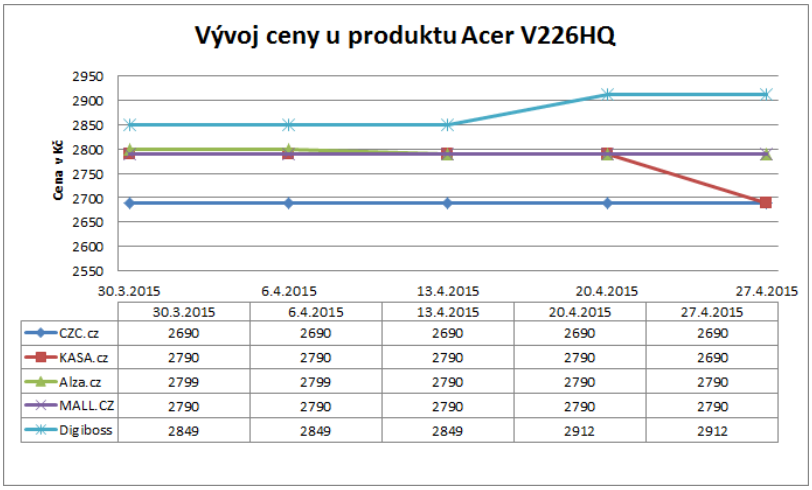
Na obrázku 6.2 lze vidět příklad takové tabulky. Uživateli je dána možnost si jednoduše zvolit pro jaké obchodníky se mají ukazovat nabídky. Pro úpravy v kontingenční tabulce Excel nabízí přímo grafické uživatelské rozhraní (na obrázku 6.2 není ukázané), kde lze nastavit její podobu. Navíc je možné při použití OLAP[4] přímo nad kontingenční tabulkou provádět citlivostní analýzy dat.

6.3 Časově závislé analýzy

Analýzy popsané v této kapitole nelze realizovat pouze nad daty z jednoho přehledu. Je nutné mít k dispozici více přehledů stažených v průběhu určitého časového období. Data z přehledů spojíme přímo dohromady nebo zpracujeme každý přehled zvlášť. Nad výsledky zpracování přehledů poté můžeme provádět jednotlivé analýzy.

Cílem analýz v této kapitole je dát uživateli informace pro určení optimální ceny produktu. Vstupem pro analýzy jsou soubory přehledů získané v průběhu jednoho měsíce z webu Heureka a webů obchodníků. První analýza popsaná v části 6.3.1 zkoumá vývoj cen pro vybrané obchodníky u produktu. V části 6.3.2 je popsána analýza, která se zabývá vývojem ceny pro daný produkt od všech obchodníků. Sleduje se vývoj minimální i maximální ceny a také ceny, která představuje medián.

6.3.1 Sledování vývoje ceny pro vybrané obchodníky



	30.3.2015	6.4.2015	13.4.2015	20.4.2015	27.4.2015
CZC.cz	2690	2690	2690	2690	2690
KASA.cz	2790	2790	2790	2790	2690
Alza.cz	2799	2799	2790	2790	2790
MALL.CZ	2790	2790	2790	2790	2790
Digiboss	2849	2849	2849	2912	2912

Obrázek 6.3: Graf vývoje ceny produktů u vybraných obchodníků

Tato analýza poskytuje možnost sledování vývoje ceny u vybraných konkurentů. Ze získaných dat za určité období vybereme nabídky sledovaných obchodníků pro daný produkt.

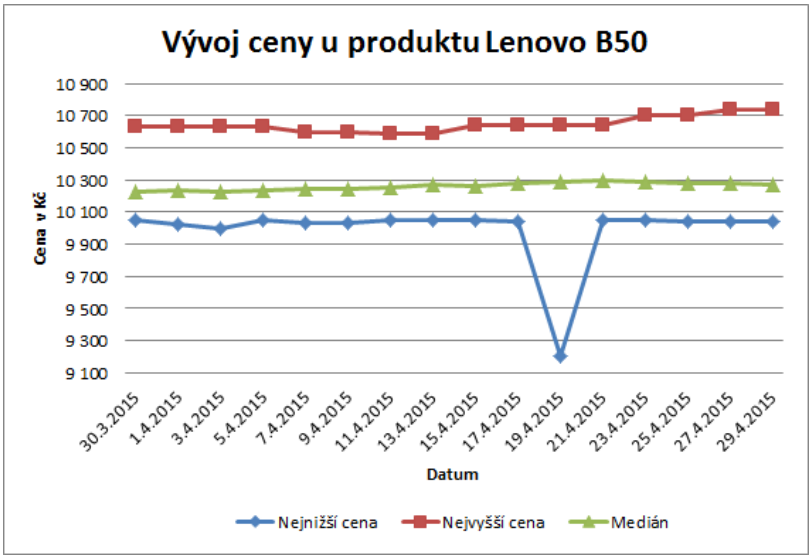
[3] http://en.wikipedia.org/wiki/Pivot_table

[4] https://support.office.com/en-US/article/Overview-of-Online-Analytical-Processing-OLAP-15d2cdde-f70b-4277-b009-ed732b75fdd6

Poté z vybraných dat vytvoříme tabulku a pro přehlednost i graf. Podle těchto podkladů je možné provést rozhodnutí o volbě optimální ceny produktu při prodeji.

Na obrázku 6.3 je možné vidět graf pro data získaná od 30.3.2015 do 29.4.2015 pro monitor Acer V226HQ[5]. Zachyceny jsou vývoje cen pro obchody CZC.CZ, Kasa, Alza, Mall a Digiboss. V grafu lze pozorovat, že v průběhu měsíce nedocházelo u obchodníků k velkým změnám ceny. Zdražení bylo zaznamenáno u firmy Digiboss a naopak zlevnění u firmy Kasa. Ostatní sledovaní obchodníci měnili cenu pouze v zanedbatelné míře.

6.3.2 Sledování vývoje ceny u produktu



Obrázek 6.4: Graf vývoje ceny minimální, maximální a mediánu

Jedná se o časovou analýzu cen, díky které získáme přehled o vývoji na trhu pro jednotlivé produkty. Analýza vychází z nabídek stažených z webu Heureka. Pro praktické účely je vhodné vynechat nabídky, které by výsledky analýzy zkreslily.

Jedná se zejména o nabídky, které mají velmi nízkou cenu oproti průměru, ale jejich skladová dostupnost je neurčitá. Často jde o chybné nebo neaktualizované nabídky, které reálně neplatí. Podobný problém se vyskytuje i u maximálních cen. Na produktových stránkách webu Heureka je podobný přehled také k dispozici. Jeho praktická použitelnost je však spíše malá z důvodu, že zahrnuje kompletně všechna data.

Obrázek 6.4 ukazuje výsledný graf pro data získaná od 30.3.2015 do 29.4.2015 stahovaná každé tři dny. Zkoumaný produkt byl v tomto případě notebook od společnosti Lenovo B50[6]. Je možné pozorovat poměrně stabilní vývoj ceny. Pouze v jednom časovém okamžiku došlo ke změně minimální ceny pravděpodobně z důvodu doprodání skladových zásob nebo akce u jednoho z obchodníků.

[5] http://lcd-monitory.heureka.cz/acer-v226hq/
[6] http://notebooky.heureka.cz/lenovo-b50-59-421971/

6.4 Další možné analýzy

V této kapitole budou nastíněny další možné analýzy nad získanými daty. Jedná se především o pokročilejší analýzy, kde bereme v úvahu všechna získaná data najednou. Hledáme spojitosti a kolerace mezi daty a jednotlivými produkty a subjekty na trhu.

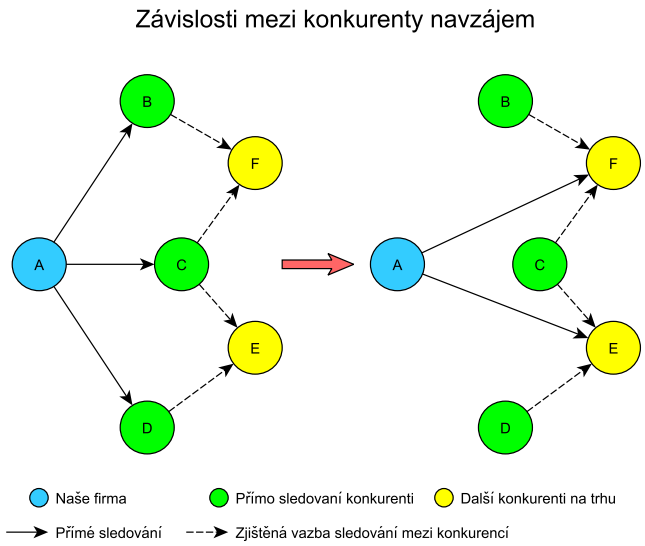
Jsou popsány analýzy, které se týkají predikce cen u konkurence. Cílem je s předstihem odhadnout cenu konkurence a nastavit správně cenu vlastní. V části 6.4.1 je popsána možná analýza pro zjištění systému přeceňování zboží u konkurence. Na tuto analýzu navazuje část 6.4.2, kde je rozebrána analýza při hledání spojitostí úpravy cen mezi konkurenty navzájem.

6.4.1 Systém přeceňování u konkurence

Při této analýze vycházíme ze změn cen v čase u vybraného obchodníka. Cílem je odhalení způsobu, jakým daný konkurent upravuje své ceny. Z dat je možné odhalit např. plošné zvednutí nebo snížení cen u produktů.

Zároveň se můžeme zaměřit i na skladovou dostupnost. Je možné tak odhalit např. systém, kdy se u zboží automaticky snižuje cena podle doby od jeho naskladnění. Že zboží bylo vyprodáno, je možné odhalit skrze absenci v přehledu případně skrze parametr skladová dostupnost. Datum dalšího přehledu, kde se nabídky znovu objeví jako dostupné, je možné považovat za datum naskladnění nových produktů.

6.4.2 Predikce reakcí konkurentů



Obrázek 6.5: Nalezené vazby mezi obchodníky v grafu

Tato analýza se podobně jako předchozí zaměřuje na určení změny cen u konkurence. V tomto případě však hledáme spojitosti mezi upravováním cen u sledovaných konkurentů a cenami u všech obchodníků, o kterých máme údaje. Nalezené spojení můžeme použít pro predikci budoucích cen.

Obrázek 6.5 ukazuje graf, kde vrcholy jsou jednotliví obchodníci a hrany představují nalezené vazby. Obchodník označený písmenem **A** je naše firma, která chce získat údaje

o obchodnících **B**, **C**, **D**. Hrany plnou čarou značí, že chceme sledovat dané firmy a vždy vycházejí z vrcholu **A**. Čárkované hrany značí, že podle stažených dat bylo nalezeno spojení mezi danými obchodníky. U jednoho obchodníka může být detekováno i více vazeb. Protože máme k dispozici časové údaje, můžeme rozhodnout o tom, která změna byla příčinou a která je pouze následkem.

V popisovaném obrázku 6.5 mají na obchodníka **F** vazbu obchodnící **B** a **C**. Na obchodníka **E** mají vazbu obchodníci **C** a **D**. Ze zjištěných údajů se můžeme při sledování zaměřit na obchodníky **E**, **F** a predikovat změny u zbývajících obchodníků podle těchto dvou.

Kapitola 7

Testování

Tato kapitola se zabývá testováním výsledné aplikace na vybraných webech. Cílem kapitoly je popsat provedené testy a vyhodnotit, zda aplikace splňuje požadavky uvedené v návrhu. Protože se jedná o aplikaci, která je závislá na externích faktorech, je nutné tomu přizpůsobit i metodiku testování. Celé testování bylo rozvrženo do časového období přibližně jednoho měsíce, aby výsledky testování (zvláště při meření rychlosti) byly co nejvíce reprezentativní pro realné použití aplikace. Testování aplikace je popsáno takovým způsobem, aby bylo případně možné testy znovu provést na jiném počítači.

Pro testování byly vybrány weby velkých českých prodejců Alza, CZC.cz, Mall, Datart, Kasa a agregátor cenových nabídek Heureka. Stahovaly se informace především k IT produktům jako jsou notebooky, tiskárny a externí pevné disky. Pro web Heureka se testovalo jak stahování informací ze zadaných stránek, tak i vyhledávání podle zadaného parametru. Primárním cílem testování aplikace je zjistit rychlost stahování a ověřit správnost získaných dat.

Kapitola je členěna do jednotlivých podkapitol, které popisují metodiku testování a vlastní testy zmíněných webů. Popis metodiky je možné nalézt v podkapitole 7.1. Samotné testování na srovnávači cen Heureka je v podkapitole 7.2. Testování webů velkých prodejců je popsáno v podkapitole 7.3. Souhrnné testování, které má odhalit případné nedostatky při nutnosti zpracovávat velké množství produktů a ověřit správnost jednotlivých testů je rozebráno v podkapitole 7.3.1. Závěr této kapitoly tvoří část 7.4, kde jsou shrnuty výsledky testování a popsány některé chyby, co byly odhaleny a odstraněny.

7.1 Metodika testování

Pro testování bylo nutné zvolit určitou metodiku, díky které bude možné otestovat rychlost programu i správnost dat. Bylo nutné vzít především v úvahu externí faktory, které nelze ovlivnit. Externími faktory se myslí především rychlost a odezva serveru, na který přicházejí požadavky o data, a rychlost připojení místa, kde je program spuštěný. Metodika byla uzpůsobena tak, aby se co nejvíce eliminoval vliv těchto faktorů na výsledky měření. Postup testování je uveden v následujícím seznamu:

1. Zvolit vzorek dat pro každý testovaný web. Předpokládaná velikost vzorku je přibližně sto a více produktů pro jeden testovaný web. Z vybraných vzorků je vytvořen soubor, kde jsou uloženy informace nutné pro stahování dat o produktech. Každý produkt je navázán právě na jednu webovou stránku pro dané testování.

2. V průběhu jednoho dne provést opakovaně stahování dat a zaznamenat výsledky a rychlost stahování. Rychlost stahování je měřena čistě při zpracování ve webovém robotu. Nejsou zahrnuty časy načítání ani ukládání dat. Z jednotlivých měření je vypočítán průměr, který je zaznamenán do výsledné tabulky. U každého webu je zachycena také průměrná odezva serveru k testovacímu počítači.

3. Pro vyhodnocení je podstatný především celkový čas a z toho vypočítaná doba pro získání jednoho produktu. Údaj o počtu nabídek a vypočítaný čas potřebný na jednu nabídku je především podstatný u webů agregátorů cenových nabídek jako je Heureka.

V rámci testování byl také ověřován očekávaný výstup. Jednalo se především o ověření, zda jsou stažená data z webu správná. První výsledky pro jednotlivé vzorky byly ověřeny zcela na shodu vůči webu, kde byly získány. Ověřování probíhalo bez použití automatických nástrojů, pouze za pomoci standardního webového prohlížeče. Při opakování testů byly vždy ověřovány pouze hodnoty, u kterých došlo ke změně.

Testování je možné označit jako úspěšné, pokud bude aplikace schopna v rozumném čase vytvořit požadovaný přehled nabídek. Přehled musí také obsahovat správná data pro jednotlivé nabídky a být použitelný do statistik a analýz dat.

7.2 Web Heureka

Pořadí testu	**Počet produktů**	**Počet nabídek**	**Celkový čas** [ms]	**Čas na produkt** [ms]	**Čas na nabídku** [ms]
1.	205	16 020	195 323	953	12
2.	205	16 152	210 489	1 027	13
3.	205	16 881	200 158	976	12
4.	205	16 712	209 589	1 022	13
5.	205	15 975	208 699	1 018	13
6.	205	16 213	195 147	952	12
7.	205	15 736	190 255	928	12
8.	205	15 600	215 964	1 053	14
9.	205	15 952	270 154	1 318	17
10.	205	16 087	198 148	967	12
Průměr	205	16 133	209 393	1 021	13

Tabulka 7.1: Výsledky testování webu Heureka

Testování webu Heureka přinášelo určité problémy. Především se jedná o velké množství získaných nabídek. Těchto nabídek může být řádově desítky až stovky na jeden produkt. Při použití vzorku, který obsahoval přes dvě stě produktů, je nutné počítat s řádově tisíci až desetitisíci nabídkami ve výsledném přehledu. Navíc může docházet v řádu dnů nebo i hodin ke zmeně počtu nabídek u jednotlivých produktů.

Při oveřování správnosti dat nebylo reálně možné procházet všechny nabídky a kontrolovat, zda jsou správné. Byl tedy zvolen postup, kdy se po každém testování vybralo několik produktů, u kterých byly zkontrolovány načtené nabídky z webu. V každém dalším testování byly vždy kontrolovány produkty, které ještě v dřívejších testech kontrolovány nebyly. Případné chyby by se také odhalily při provádění analýz popsaných v kapitole 6.

Tabulka 7.1 obsahuje výsledky testování pro případ, kdy jsou u jednotlivých produktů přímo zadány odkazy na jejich stránky. Průměrná odezva serveru byla při testování 19 až 22 ms. Z výsledků je možné pozorovat, že aplikace potřebuje přibližně jednu sekundu na načtení informací o jednom produktu, u kterého je přibližně 80 nabídek.

7.2.1 Testy vyhledávání dat

Pořadí testu	**Počet produktů**	**Nalezených produktů**	**Počet nabídek**	**Celkový čas** [ms]	**Čas na produkt** [ms]	**Čas na nabídku** [ms]
1.	100	92	3 515	87 138	871	25
2.	100	92	3 612	86 002	860	24
3.	100	92	3 608	86 224	862	24
4.	100	92	3 560	90 125	901	25
5.	100	92	3 677	85 100	851	23
6.	100	92	3 623	86 578	866	24
7.	100	92	3 641	87 004	870	24
8.	100	92	3 623	87 879	879	24
9.	100	92	3 589	88 523	885	25
10.	100	92	3 599	87 627	876	24
Průměr	100	92	3 605	87 220	872	24

Tabulka 7.2: Výsledky testů vyhledávání na webu Heureka

V této části jsou rozebrány testy vyhledávání produktů na webu Heureka. Jako údaj pro vyhledávání byl použit part number produktu. Nabídky byly získávány pouze z nalezených produktových stránek.

Do testovaného vzorku byly zahrnuty i produkty, které nemají nabídky na webu Heureka. Výsledky testů pro vzorek 100 produktů jsou obsaženy v tabulce 7.2. Počet nalezených produktových stránek byl pro každé testování jen 92. Ověření výstupu potvrdilo, že pro osm nenalezených produktů neexistuje odpovídající produktová stránka.

Při srovnání s předchozím testem je zejména zajímavý rozdíl v čase potřebném na zpracování produktu. Obecně by tento čas měl být delší pokud aplikace musí navíc vyhledat i produktovou stránku. Nameřené hodnoty ovšem ukázaly výsledek opačný. Tento stav nastal ze dvou důvodů.

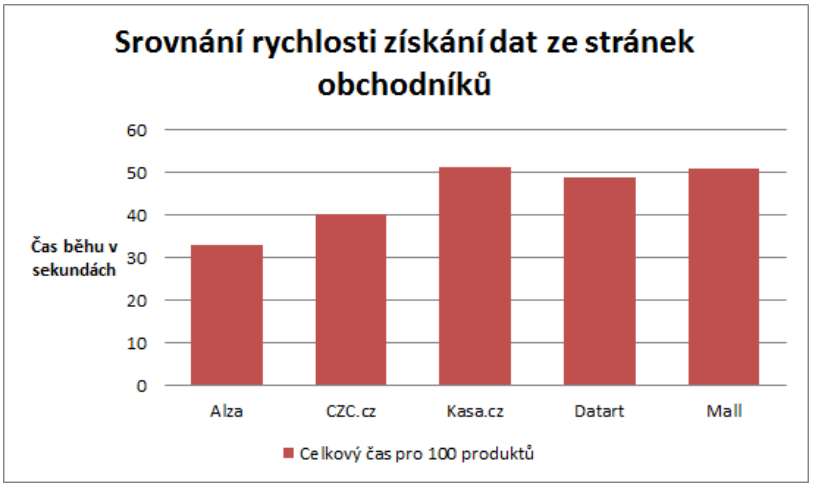
Prvním důvodem jsou již zmíněné nenalezené produkty. Pro ty nedocházelo ke zpracování produktové stránky, ale čas na produkt je vypočítán podle počtu produktů v daném testovacím vzorku. Pokud bychom brali do úvahy jenom nalezené produkty, tak by výsledný průměrný čas na jeden produkt vyšel 948 ms, což je stále menší čas než u prvního testování. Je proto také nutné vzít do úvahy počet nabídek k daným produktům.

U testovacího vzorku pro vyhledávání dat se vyskytovaly produkty, ke kterým bylo nalezeno průměrně asi poloviční množství nabídek oproti vzorku se zadanou produktovou stránkou. Důležitý parametr, podle kterého je možné porovnávat, je tedy čas potřebný na jednu nabídku. Ten je při testech vyhledávání delší než při testech se zadanou produktovou stránkou. Obecně bude metoda s vyhledáváním vždy delší minimálně o získání stránky s výsledky vyhledávání a jejího zpracování.

7.3 Weby velkých obchodníků

Obchod	Počet Produktů	Celkový čas [ms]	Průměrný čas na produkt [ms]	Průměrná odezva serveru [ms]
Alza	100	33 019	330	24
CZC.cz	100	40 240	402	18
Kasa.cz	100	51 237	512	22
Datart	100	48 963	490	13
Mall	100	50 971	510	19

Tabulka 7.3: Souhrná tabulka testování webů obchodníků



Obrázek 7.1: Srovnání rychlosti pro weby obchodníků z tabulky 7.3

Druhá část testování se zaměřila na získávání dat od největších tuzemských elektronických obchodů. Jejich výběr, popsaný v kapitole 2.5.2, byl založený na celkovém obratu dané firmy. Vybrány pro testování byly weby obchodníků: Alza, CZC, Kasa, Datart a Mall. Všichni disponují profesionálním webem, který nemá zásadní nedostatky ve struktuře, jež by znemožňovaly automatické stahování.

Postup testování byl stejný jako v případě webu Heureka popsaného v předchozí části této kapitoly. V tabulce nejsou již uvedeny kompletní testy, ale pouze výsledné průměrné hodnoty ze všech testů. Z hlediska rychlosti není mezi weby prodejců výrazný rozdíl. V grafu porování rychlosti stahování 7.1 je možné vidět, že nejrychleji se stahují produkty z webu Alza.cz, nejpomaleji potom z webu Kasa.cz.

Na webech prodejců je potřebný přibližně poloviční čas na zpracování stránky jednoho produktu oproti webu Heureka. Stránky prodejců vždy obsahují pouze jednu nabídku na daný produkt a jejich zpracování je tedy rychlejší.

Další testované weby

Kromě zmíněných webů v předchozí části jsou navíc připravena nastavení pro stahování dat z webů obchodníků: Digiboss[1], ePROTON[2], Euronics[3] a T.S. Bohemia[4]. Testování těchto webů probíhalo pouze na menším množství testovacích produktů při tvorbě nastavení. Jejich funkčnost tedy není zcela ověřena. Při kompletním testování těchto webů by se postupovalo obdobně jako v předchozích případech.

7.3.1 Souhrnné testování

Pořadí testu	**Počet produktů**	**Počet nabídek**	**Celkový čas** [ms]	**Čas na produkt** [ms]	**Čas na nabídku** [ms]
1.	705	16 750	448 929	637	27
2.	705	16 258	421 325	598	26
3.	705	16 960	435 258	617	26
4.	705	16 354	451 487	640	28
5.	705	16 999	443 458	629	26
6.	705	16 454	415 807	590	25
7.	705	16 202	451 478	640	28
8.	705	16 645	439 587	624	26
9.	705	16 954	454 789	645	27
10.	705	16 240	425 879	604	26
Průměr	705	16 582	438 800	622	26

Tabulka 7.4: Souhrnné testování všech produktů

Tabulka 7.4 ukazuje výsledky pro vzorek dat, který je sloučením všech dříve testovaných produktů. Cílem toho testovacího případu je vyzkoušet, zda v programu nedochází k chybám a velkému nárůstu času při zpracování většího vzorku dat. Z hodnot předchozích testů je možné dopočítat očekávaný průměrný čas zpracování jednoho produktu. Využijeme k tomu znalosti celkového počtu produktů a průměrných časů na zpracování produktu u jednotlivých scénářů. Při použití tohoto postupu vychází čas na zpracování jednoho produktu 615 ms. Průměrný čas získaný z testování je 622 ms.

Je vidět, že teoreticky předpokládaný čas je podobný s časem získaným při měření. Aplikace by tedy neměla mít problémy ani při zpracovávání rozsahem většího množství produktů. Oproti dřívějšímu testování webu Heureka je čas potřebný na získání jedné nabídky přibližně dvojnásobný.

Jedná se znovu o očekávaný výsledek, protože platí dříve popsaný stav, že na jednu stránku webu prodejce připadá právě jedna nabídka. Čas na získání jedné nabídky, je tedy u webů prodejců roven času na získání jednoho produktu. Při porovnání časů je vidět, že aplikace potřebuje přibližně 35x více času pro získání nabídky z webů prodejců oproti webu Heureka. Po zahrnutí nabídek z webů prodejců je tedy průměrný čas na získání jedné nabídky větší než z webu Heureka.

[1] http://www.digiboss.cz/
[2] http://www.eproton.cz/
[3] http://www.euronics.cz/
[4] https://www.tsbohemia.cz/

7.4 Výsledky testování

Výsledky testování ukázaly, že aplikace splňuje požadované vlastnosti popsané v kapitole 2.2. Z hlediska rychlost aplikace dokáže získat data k několika tisícům produktů za hodinu. Zároveň je pro praktické účely použitelné také vyhledávání podle zadaného parametru na webu Heureka. Je tedy možné denně aktualizovat přehled k produktům i u distributorů, jejichž produktové portfolio je obecně daleko vetší než u bežných obchodníků.

Objeveno bylo rovněž několik chyb, které byly ve výsledné aplikaci opraveny. Více o nalezených chybách je popsáno v následující části.

7.4.1 Nalezené chyby a jejich řešení

Nalezené chyby v průběhu testování byly neprodleně odstraněny. Jednotlivé chyby obvykle souvisí se špatným označením dat na webu, případně vynecháváním některých požadovaných údajů. V následujícím seznamu jsou uvedeny jednotlivé chyby, které byly odstraněny. Nemusí se jednat pouze o weby uvedené v této kapitole o testování.

- Datart – chyba nenačtení ceny pro některé produkty.
- Heureka – chyba načítání špatného parametru při použití vyhledávání. Problémy s načtením názvu obchodu u určitých nabídek.
- CZC – chyba při načítání ceny produktu.
- Digiboss – načítání špatné ceny (cena bez dph).
- Mall – vynechávání ceny u některých produktů.
- Chyby v programu – nezapisování první nabídky do seznamu a přehození dat ve sloupcích ve výsledném přehledu.

Většina chyb byla řešena úpravou XPath výrazů pro danou stránku. XPath výraz nezahrnoval všechny možnosti, případně byl napsán pro špatný údaj. Chyby s daty ve výsledném přehledu byly odstraněny úpravou metod, kterými byl přehled generován. V seznamu nebyly zahrnuty chyby, které se spíše týkaly technické části aplikace např. špatné použití metod z knihovny atd.

Kapitola 8

Závěr

V této diplomové práci byla provedena analýza, návrh a implementace aplikace pro hromadné srovnávání cen produktů. Práce rozebírá analýzu cen z pohledu dodavatelů produktů a prodejců, kteří poté produkty nabízejí koncovým zákazníkům. Obecně jsou popsány možné zdroje dat a také provedena analýza a vyhledání největších internetových obchodníků a srovnávačů cen v rámci České republiky. Výstupem bude soubor, ve kterém budou umístěny všechny nabídky z monitorovaných webů.

Popsány jsou také knihovny, které je možné využít při realizaci aplikace v jazyce Java. Do úvahy bylo nutné brát problémy spojené s použitím těchto knihoven, neboť zpracovávají stránky jinými prostředky než moderní prohlížeče. Rozebrány byly rovněž ochrany, které mohou weby využívat proti robotům získávajícím z webu data. Návrh aplikace popisuje základní rozdělení aplikace na hlavní části a obsahuje UML diagramy. Vytvořen byl diagram případů užití, konceptuální diagram tříd a diagram balíčků. Navrženo bylo uživatelské rozhraní se soupisem funkcí, které bude poskytovat.

Implementována byla aplikace, která umožňuje stahování informací o cenách, skladové dostupnosti, popřípadě dalších informací z webů podle zadaných parametrů. Aplikace poskytuje uživateli možnost vytvořit si vlastní nastavení pro stahované weby. Zároveň je umožněno využít systému vyhledávání, který byl implementován pro web Heureka. Uživatel může zadat parametr (např. part number), podle kterého se budou vyhledávat produktové stránky s nabídkami. Stahované produkty je možné zadávat ve formátu XML, XLSX a CSV. Výstupem je přehled, kde jsou uvedeny všechny nalezené nabídky. Pro výstup z aplikace je rovněž implementována podpora pro formáty XLSX a CSV.

Celá implementace aplikace byla provedena v jazyku Java s uživatelským rozhraním v knihovně Swing. Rozdělena je do dvou částí, a to na webového robota a nástavbu nad tímto robotem v podobě uživatelského rozhraní a správy nastavení. Webového robota je tedy možné použít i samostatně při řešení obdobných problémů. Nastavení webového robota i nastavení pro jednotlivé weby je uloženo ve formátu XML.

Aplikaci lze využít především pro sledování vývoje cen na trhu. Výstupní data z aplikace mohou být vstupy do dalších nástrojů, jako je např. program Microsoft Excel. Firma nad získanými daty může provádět analýzy podle vlastních potřeb, aniž by bylo nutné zasahovat do kódu samotné aplikace pro stahování dat. Může se jednat např. o zjištění produktů, jejichž nastavená cena není konkurenceschopná, nebo sledování vývoje cen u jednotlivých konkurentů.

8.1 Možnosti budoucích rozšíření

Aplikace disponuje v základu nastavením pro stahování z produktových stránek u přibližně deseti webů. Pro reálné nasazení je potřeba připravit nastavení i pro další weby obchodníků. Další vhodnou části pro rozšíření aplikace je uživatelské rozhraní. Jednalo by se o přidání dalších možností především pro úpravu nastavení webů a webového robota. Zvětšil by se tak značně komfort uživatele při práci s aplikací.

Literatura

[1] Obrázek z aplikace Price Mentor. *Price Mentor* [online]. [cit. 2014-12-10]. Dostupné z: http://www.pricementor.com/price-monitor/

[2] Obrázek z aplikace Karsa Monitor. *Karsa Monitor* [online]. [cit. 2014-12-10]. Dostupné z: http://www.karsa-monitor.cz/manufacturers-cz

[3] Obrázek z aplikace od Upstream Commerce. *Upstream Commerce* [online]. [cit. 2014-12-10]. Dostupné z: http://upstreamcommerce.com/products_/pricing-intelligence/

[4] Stránka produktu HP Pavilion 15-n268. *Heureka* [online]. [cit. 2014-12-10]. Dostupné z: http://notebooky.heureka.cz/hp-pavilion-15-n268-g5f31ea/porovnat-ceny/#offers

[5] Stránka produktu HP Pavilion 15-n268. *Zboží.cz* [online]. [cit. 2014-12-10]. Dostupné z: http://www.zbozi.cz/vyrobek/hp-pavilion-15-n268sc-g5f31ea-bcm/

[6] Online Aplikace NetMonitor. *NetMonitor* [online]. [cit. 2014-12-27]. Dostupné z: http://online.netmonitor.cz/

[7] Veřejný rejstřík a Sbírka listin. *eJustice* [online]. [cit. 2014-12-27]. Dostupné z: https://or.justice.cz/ias/ui/rejstrik

[8] GOURLEY, David a Brian TOTTY. *HTTP: the definitive guide*. 1st ed. Sebastopol, CA: O'Reilly, 2002, xviii, 635 s. ISBN 15-659-2509-2.

[9] HEATON, Jeff. *HTTP programming recipes for Java bots*. 1st ed. St. Louis, MO: Heaton Research, Inc, 2007. ISBN 09-773-2066-9

[10] MITCHELL, Ryan a James HOLMES. *Instant web scraping with Java*. Online-Ausg. Birmingham: Packt Publishing, 2013, xi, 370 s. ISBN 978-184-9696-883.

[11] LIU, Bing. *Web data mining: exploring hyperlinks, contents, and usage data*. Berlin: Springer, c2007, xix, 532 s. ISBN 978-3-540-37881-5.

[12] KAY, Michael. *XSLT 2.0 and XPath 2.0 Programmer's Reference*. 4th ed. Indianapolis, IN: Wiley Publishing, 2008, 1316 s. ISBN 978-0-470-19274-0.

[13] RICHTA, Karel. *Jazyky XQuery a XPath* [online]. [cit. 2014-12-29]. Dostupné z: www.ksi.mff.cuni.cz/~richta/publications/RichtaMD2006.pdf

[14] Introduction to JAXB. *Oracle Documentation* [online]. [cit. 2015-03-10]. Dostupné z: https://docs.oracle.com/javase/tutorial/jaxb/intro/index.html

[15] The Swing Tutorial. *Oracle Documentation* [online]. [cit. 2015-03-22]. Dostupné z: http://docs.oracle.com/javase/tutorial/uiswing/

Příloha A

Obsah CD

Jako příloha je k diplomové práci přidáno CD, které obsahuje položky uvedené v následujícím seznamu:

- Textová podoba diplomové práce v souboru **DIP_xvarad01.pdf**.
- Soubory pro textovou práci ve složce **latex**.
- Přeložená aplikace ve složce **WebDownloader**. Přiloženy jsou i soubory s produkty, které byly použity při testování.
- Vygenerovaná programová dokumentace ve složce **doc**.
- Provedené analýzy ve složce **statistiky_a_prehledy**. U některých souborů je nutné přenastavit cesty pro datové spojení.

Příloha B

Manuál

Manuál popisuje překlad, spuštění a způsob použití implementované aplikace.

Překlad a spuštění aplikace

Pro překlad je nutné mít nainstalováno JDK[1] ve verzi 1.8. Překlad probíhá pomocí nástroje Ant[2] (příkaz **ant jar**). Poskytnuta je také možnost vygenerování programové dokumentace (příkaz **ant doc**). Výsledný jar soubor lze standardně spustit pomocí programu java. Případně je možné pro spuštění užít překládací skript (příkaz **ant run**).

Práce s aplikací

Aplikaci je možné používat ve dvou režimech. V prvním režimu je aktivní jednoduché uživatelské rozhraní. V tom uživatel nalezne možnosti pro otevření souborů s produkty ve formátech XLSX, CSV, XML a export do formátů XLSX a CSV. Otevřený soubor je zobrazen v tabulce a uživatel má možnost spustit stahování dat. Tabulka se průběžně aktualizuje po každém zpracovaném produktu. Dále je uživateli nabídnuta možnost vygenerování XML souborů s produkty pro zadaný vstup ve formátu XLSX.

Druhou možností je spuštění přes příkazovou řádku se zadanými parametry. V takovém případě nedochází k aktivaci uživatelského rozhraní. Aplikace pro zadaná data stáhne nabídky a uloží je do požadovaného výstupního souboru. Zadávané parametry jsou:

- **-input** filePath Jméno vstupního souboru
- **-output** filePath Jméno výstupního souboru

[1] http://www.oracle.com/technetwork/java/javase/downloads/index.html
[2] http://ant.apache.org/

Příloha C

Ukázky souborů s nastavením

V této příloze jsou ukázky XML souborů s nastavením pro aplikaci.

Nastavení pro roboty

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<webRobotXmlBotsSettings>
    <robot>
        <botName>ROBOT_A</botName>
        <browserType>CHROME</browserType>
        <options>
            <activeXNative>false</activeXNative>
            <appletEnabled>false</appletEnabled>
            <cssEnabled>true</cssEnabled>
            <doNotTrackEnabled>false</doNotTrackEnabled>
            <geolocationEnabled>false</geolocationEnabled>
            <homePage></homePage>
            <javaScriptEnabled>false</javaScriptEnabled>
            <popupBlockerEnabled>false</popupBlockerEnabled>
            <printContentOnFailingStatusCode>true</printContentOnFailingStatu
            <redirectEnabled>true</redirectEnabled>
            <throwExceptionOnFailingStatusCode>true</throwExceptionOnFailing
            <throwExceptionOnScriptError>true</throwExceptionOnScriptError>
            <timeout>90000</timeout>
            <useInsecureSSL>false</useInsecureSSL>
            <cookiesEnabled>false</cookiesEnabled>
        </options>
    </robot>
</webRobotXmlBotsSettings>
```

Nastavení pro produkt

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<webRobotXmlProduct>
    <productLinks>
        <information>
            <map>
                <entry>
                    <key>Currency</key>
                    <value>CZK</value>
                </entry>
            </map>
        </information>
        <webRobotXmlHeurekaPageHolder>
            <link>http://notebooky.heureka.cz/apple-macbook-air-md760cz-b/</
            <settingsId>Heureka</settingsId>
            <fulltextSearch>false</fulltextSearch>
            <maxPagesFulltext>0</maxPagesFulltext>
            <useSearchEngine>false</useSearchEngine>
        </webRobotXmlHeurekaPageHolder>
        <shopName>Heureka.cz</shopName>
    </productLinks>
    <productName>Apple MacBook Air MD760CZ/B</productName>
</webRobotXmlProduct>
```

Soubor pro stahování v XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xmlSummary>
    <products>.\products\product1.xml</products>
    <products>.\products\product2.xml</products>
    <products>.\products\product3.xml</products>
    <products>.\products\product4.xml</products>
    <products>.\products\product5.xml</products>
    <products>.\products\product6.xml</products>
    <products>.\products\product7.xml</products>
    <products>.\products\product8.xml</products>
    <products>.\products\product9.xml</products>
    <products>.\products\product10.xml</products>
</xmlSummary>
```

Soubor pro stahování v XLSX/CSV

Link	Shop Settings	Shop Name	PartNo	Sell Price	Buy Price
https://www.alza.cz/ hp-250-g3-d2167929.htm	alza	Alza	2001	7222	6485
https://www.alza.cz/lenovo-ideapad -g50-30-black-d2173179.htm	alza	Alza	2002	8950	8467
https://www.alza.cz/acer-aspire-e15 -midnight-black-d2368575.htm	alza	Alza	2003	14850	14587
https://www.alza.cz/dell-inspiron -13z-touch-d2414679.htm	alza	Alza	2004	19999	18018
https://www.alza.cz/dell-inspiron -11-touch-d2147910.htm	alza	Alza	2005	11850	10434
https://www.alza.cz/asus-x553ma -sx376h-cerny-d2221457.htm	alza	Alza	2006	8150	8042
https://www.alza.cz/lenovo-ideapad -g500-texture-black-d1939996.htm	alza	Alza	2007	11842	10271