



**VYSOKÁ ŠKOLA CHEMICKO-TECHNOLOGICKÁ V PRAZE**  
**Fakulta chemické technologie**

---

**Laboratoř informatiky a chemie**

# DIPLOMOVÁ PRÁCE

**Webová aplikace pro vizualizaci krystalových struktur  
anorganických sloučenin**

**Web application for vizualization of crystal structures of  
inorganic compounds**

**Bc. Tomáš Müller**

Prohlašuji, že jsem předloženou diplomovou prací vypracoval samostatně a použil jsem jen pramenů, které cituji a uvádím v seznamu použité literatury.

V Praze dne .....

.....

Podpis



# SOUHRN

## **Webová aplikace pro vizualizaci krystalových struktur anorganických sloučenin**

Cílem práce bude vyvinout webovskou aplikaci pro vizualizaci krystalových struktur anorganických sloučenin. Jádrem aplikace bude systém pro zobrazování krystalů anorganických sloučenin s různou krystalovou symetrií. Krystalové struktury budou bud získány z vhodné krystalografické databáze, nebo bude uživateli dána možnost zobrazení vlastních souřadnic z formátu cif a dalších krystalografických datových formátů. Pro správnou vizualizaci bude třeba dopočítat polohy symetricky ekvivalentních atomů. Úkolem bude též spočítat polohy všech atomů ve výřezu zvoleném uživatelem a případně vykreslit koordinační polyedry. Aplikace bude vytvořena ve frameworku Django za použití javascriptové knihovny JQuery. Pro vlastní vizualizaci bude použit javovský molekulový prohlížeč JMol.

## **Web application for vizualization of crystal structures of inorganic compounds**

The aim of the work is to develop the web application for inorganic crystal structures vizualization. The system will be capable of displaying crystals of inorganic compounds with various crystal symmetry. Crystal structures will either be obtained from the appropriate crystallographic database or the user will be able to vizualize its own structures stored in cif and other data formats. The positions of symmetry equivalent atoms will have to be calculated for proper vizualization. The application will be implemented using the Django web framework and the Javascript library JQuery. The vizualization will be realized by the JMol Java applet.

Studijní obor: **Aplikovaná informatika v chemii**

Diplomová práce: **Bc. Tomáš Müller**

Vedoucí práce: **Mgr. Daniel Svozil, PhD.**

Konzultant: **prof. Dr. Ing. David Sedmidubský**

Práce odevzdána dne: .....

# Poděkování

Chtěl bych poděkovat svému vedoucímu práce Mgr. Danielu Svozilovi, PhD. za obětavý přístup, prof. Dr. Ing. Davidu Sedmidubskému za podnětné připomínky, ing. Petru Čechovi za ochotu a technickou podporu a mému kolegovi Ctiboru Škutovi za podpoření mé kreativity.

# Obsah

<b>1.</b>	<b>ÚVOD</b> .....	<b>- 1 -</b>
<b>2.</b>	<b>TEORETICKÁ ČÁST</b> .....	<b>- 2 -</b>
2.1	KRYSTALOGRAFIE .....	- 2 -
2.1.1	<i>Historický úvod</i> .....	- 2 -
2.1.2	<i>Krystalizace</i> .....	- 4 -
2.1.3	<i>Symetrie krystalů</i> .....	- 5 -
2.1.4	<i>Rentgenové záření</i> .....	- 11 -
2.1.5	<i>Strukturní analýza krystalů</i> .....	- 14 -
2.2	KRYSTALOGRAFICKÉ DATABÁZE .....	- 16 -
2.2.1	<i>Crystallography Open Database</i> .....	- 16 -
2.2.2	<i>American Mineralogist Crystal Structure Database</i> .....	- 16 -
2.2.3	<i>Databáze NIST</i> .....	- 17 -
2.3	DATOVÉ FORMÁTY .....	- 17 -
2.3.1	<i>STAR</i> .....	- 17 -
2.3.2	<i>CIF</i> .....	- 20 -
2.4	POUŽITÝ SOFTWARE .....	- 21 -
2.4.1	<i>Python</i> .....	- 21 -
2.4.2	<i>Django</i> .....	- 23 -
2.4.3	<i>PyCIFRW</i> .....	- 26 -
2.4.4	<i>Jmol</i> .....	- 27 -
2.4.5	<i>JavaScript</i> .....	- 28 -
2.4.6	<i>JQuery</i> .....	- 29 -
<b>3.</b>	<b>PRAKTICKÁ ČÁST</b> .....	<b>- 30 -</b>
3.1	ZÁKLADNÍ POPIS PROJEKTU PRSK (PROHLÍZEČ STRUKTUR KRYSTALŮ) .....	- 30 -
3.2	WEBOVÝ VÝSTUP .....	- 31 -
3.3	POPIS JEDNOTLIVÝCH FUNKCIONALIT PORTÁLU .....	- 32 -
3.3.1	<i>Novinkový systém</i> .....	- 32 -
3.3.2	<i>Krystalové struktury</i> .....	- 33 -

3.3.3	<i>Prohledávat databáze</i> .....	- 35 -
3.3.4	<i>Vizualizace</i> .....	- 40 -
3.3.4.1	Upload CIF souboru .....	- 41 -
3.3.4.2	Tvorba vlastní struktury .....	- 41 -
3.3.4.3	Zobrazení externího CIF souboru .....	- 47 -
3.3.4.4	Ovládání Jmol appletu .....	- 47 -
3.4	ADMINISTRACE PORTÁLU .....	- 48 -
3.4.1	<i>Novinkový systém</i> .....	- 49 -
3.4.2	<i>Krystalové struktury</i> .....	- 50 -
3.5	NÁPOVĚDY .....	- 51 -
<b>4.</b>	<b>ZÁVĚR</b> .....	<b>- 53 -</b>
<b>5.</b>	<b>POUŽITÁ LITERATURA</b> .....	<b>- 54 -</b>
<b>6.</b>	<b>PŘÍLOHY</b> .....	<b>- 59 -</b>

# 1. Úvod

Výuka krystalografie je velice náročná na prostorovou představivost. Pouze z textu si lze jen velice obtížně udělat představu o uspořádání prvků v třírozměrném prostoru. O něco lepší je použití obrázků z učebnic a skript, avšak tyto obrázky jsou stále z pochopitelných důvodů dvojrozměrné, a tedy mohou zkreslovat až mást. Nejlepší představu získáme pomocí fyzických třírozměrných modelů, které jsou ovšem omezené co do počtu, tak do velikosti a complexity. Dobrým kompromisem je počítačová vizualizace, která není nijak omezena počtem zobrazovaných krystalů a téměř není omezena jejich komplexitou. V zásadě se stále jedná o plochý dvojrozměrný obraz, avšak interaktivitou či simulací třetího rozměru můžeme získat představu o prostorovém uspořádání poměrně jasně. Vytvoření takového nástroje je předmětem této diplomové práce.

## 2. Teoretická část

### 2.1 Krystalografie

#### 2.1.1 Historický úvod

Slovo krystal pochází z řeckého *krystalus* (kus ledu), z kmene *kryos* (mráz)<sup>1</sup>. Původně se takto nazýval křišťál – průhledný křemen, o kterém se Řekové domnívali, že je to zkamenělý led. Později se tento termín rozšířil na všechny minerály pravidelných útvarů s hladkými, lesklými plochami. Lidé byli již od dávných dob fascinováni krystaly. Například bájný kámen mudrců měl být neznámá krystalická látka schopná měnit různé běžné kovy (obzvláště olovo na zlato) na drahé kovy. Také se věřilo, že tato látka je schopná lidem vrátit mládí, či dokonce dát jim nesmrtelnost. O jeho vyrobění se pokoušelo mnoho středověkých alchymistů, mezi které patřil například i Sir Isaac Newton. Krystaly vzácných minerálů se také používaly a používají jako ozdoby hodné i královských korun. Mezi ty nejznámější patří například rubín či safír (oxid hlinitý), smaragd (hlinitokřemičitan berylnatý) nebo ametyst (oxid křemičitý).

V 17. století se badatelé začali zabývat krystaly z hlediska jejich tvaru a struktury, a tak vznikla nová vědní disciplína – krystalografie. V tomto období často učinili pokrok vědci z jiných vědních oborů, jako například astronom J. Kepler, který v roce 1611 popsal strukturu sněhové vločky či kodaňský profesor anatomie N. Stensen, který roku 1669 objevil, že úhly mezi hranami v krystalech křemene jsou vždy konstantní.

Fyzikové R. Hooke roku 1665 a Ch. Huygens roku 1690 navrhli shodné vysvětlení krystalových struktur jako pravidelné ukládání kulovitých částíček hmoty. Za zakladatele krystalografie je považován R. J. Haüy, který popsal vnější tvar krystalů jako důsledek jejich vnitřní struktury. Jeho představa byla taková, že krystal

se sestává z pravidelně uspořádaných rovnoběžnostěnů, které se skládají do různých krystalových tvarů.

Roku 1809 C. S. Weiss rozdělil krystaly do 7 krystalografických soustav. Roku 1839 W. H. Miller zpopularizoval návrh S. Whewella z roku 1825 a G. Grassmana z roku 1829 pro popis krystalových ploch pomocí úseků, které tyto plochy vytínají na krystalografických osách, a zavedl notaci, kde se tyto poměry zapisují jako tři čísla v kulatých závorkách. Od té doby jsou indexy (hkl) nazývány Millerovy indexy nebo též indexy roviny<sup>2</sup>. V roce 1831 odvodil J. F. Hessel 32 symetrických oddělení. Nezávisle na něm začal roku 1867 A. Gadolin používat systematiku 32 krystalografických tříd. Roku 1950 dospěl A. Bravais pečlivou geometrickou analýzou k závěru, že body mohou být v prostoru vzhledem k symetrii uspořádány pouze 14 způsoby. Vznikl tak pojem 14 Bravaisových mřížek a byl tak položen základ mřížkové koncepce v krystalografii. Vrcholem geometrické krystalografie bylo odvození počtu možných uspořádání prvků souměrnosti v prostoru, kterých je 230. Toho docílili E. S. Fedorov (1890), A. Schoenflies (1891) a W. Barlow (1894).

Zhruba v tomto období se z krystalografie vyvinula krystalochemie. Ta si kladla za cíl dát do souvislosti tvar krystalu a jeho chemické složení. Prvního velkého úspěchu bylo dosaženo v roce 1912, kdy vědci Friedrich a Knipping dle návrhu M. von Laueho zrealizovali experiment, při kterém nechali dopadat úzký paprsek rentgenového záření se spojitým spektrem na krystal modré skalice. Výsledkem byl difrakční obrazec podobný difrakčnímu obrazci vzniknuvšímu průchodem světla rytou optickou mřížkou. Z tohoto byly vyvozeny následující dva závěry: 1. rentgenové záření je vlnové povahy a 2. krystalová struktura je nespojitá a periodická. Zřejmě kvůli podobnosti chování rentgenového záření při průchodu krystalem a světla při průchodu optickou mřížkou se krystalové struktuře začalo říkat krystalová mřížka, což může být poměrně matoucí, neboť pojem „mřížka“ se v tomto oboru používá v jiném významu.

W. H. Bragg a jeho syn W. L. Bragg si díky experimentu s rentgenovým

paprskem uvědomili, že v difrakčním obrazci musí být nějakým způsobem zakódovaná informace o struktuře krystalu. V letech 1912 – 1913 se jim podařilo nalézt první krystalové struktury chloridu sodného a chloridu draselného.

Postupem času byly metody difrakční analýzy dále vyvíjeny, modifikovány a zpřesňovány. Vývoje se dočkal i samotný pojem krystal. Jeho definice – pravidelný útvar s lesklými hranami - přestala být dostačující, přestože takovýto vzhled je pro krystal příznačný. Podobného vzhledu je však možné docílit i u nekrystalických látek pomocí broušení. Na přelomu 19. a 20. století začal být při definici krystalu kladen větší důraz na jeho fyzikální vlastnosti. P. Niggli jej například definoval jako homogenní anizotropní (závislé na směru) diskontinuum. Další charakteristická chemicko-fyzikální vlastnost, kterou se krystalická látka liší od látky amorfní, je ostrý bod tání.

### **2.1.2 Krystalizace**

Krystaly vznikají krystalizací. Vznikla tak i většina zemské kůry. Asi nejstarší cílevědomá krystalizace prováděná člověkem byla odpařování mořské vody, čímž lze snadno získat sůl. Fascinace lidské rasy drahokamy vedla k vývoji jejich umělé výroby. Obecný rys krystalů je jejich čistota, a proto se začalo využívat krystalizace při výrobě čistých látek. Monokrystalické materiály se zase používají pro výrobu polovodičových součástek (diody, tranzistory) pro elektroniku. Další chemicko-fyzikální vlastnosti krystalů jako piezoelektrický jev, magnetismus nebo optické vlastnosti jim zaručují široké uplatnění.

Stavební prvky krystalů mohou být atomy, molekuly, ale třeba i biomolekuly. Síly držící stavební prvky pohromadě mohou být povahy iontové, kovalentní nebo van der waalsovské. Nejdůležitějšími faktory při výstavbě monokrystalu jsou charakter vazby a souměrnost stavební jednotky. Tyto faktory více než kompenzují snahu o maximální neuspořádanost, krystaly jsou velice stabilní. Jedna látka může vykristalizovat do více tvarů, záleží na podmínkách krystalizace.

Vzájemná orientace stavebních prvků krystalu je řízena stejnými zákonitostmi jako stavba molekul, tedy elektronovou konfigurací a valenčním stavem. Velice důležitý faktor pro krystalizaci je krystalizační prostředí – zdroj stavebních prvků krystalu. To může být tuhé, kapalné i plynné. Zajistit dokonalý růst krystalů není snadné, v tomto úkonu figuruje mnoho proměnných. Obecně však platí, že čím pomaleji krystal roste, tím je dokonalejší.

### 2.1.3 Symetrie krystalů

Vnitřní struktura krystalů je vysoce uspořádaná a periodicky se opakující, a proto je rozumné ji stejným způsobem popisovat. V každé takové struktuře můžeme najít opakující se motiv podobně jako na tapetách, jen ve třírozměrném prostoru. Opakováním motivu v prostoru pak dostaneme celou krystalovou strukturu. Důvodem pro takto uspořádaný systém je energetická výhodnost.

Pokud bychom základní motiv nahradili jediným bodem a ten opakovali namísto motivu, dostáváme prostorovou mřížku, která zachovává původní geometrické vztahy struktury (vzdálenosti a úhly). Každá krystalová struktura má pouze jednu vlastní krystalovou mřížku, avšak dvě různé struktury mohou mít mřížku stejného typu. Existuje mnoho rozmanitých krystalových struktur, avšak jen omezené množství krystalových mřížek.

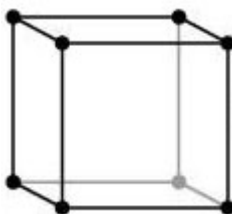
Mnohdy dochází k záměnám pojmů krystalová struktura a krystalová mřížka. Krystalová struktura má představovat určitou fyzikální realitu, zatímco krystalová mřížka je pouze pomůcka, jak tuto realitu popsat.

Z každého bodu krystalové mřížky lze vést řady translačních vektorů do jiných bodů mřížky, ale stačí nám pouze 3 vektory (neležící v jedné rovině) pro jednoznačný popis mřížky. Tato trojice vektorů  $(\vec{a}, \vec{b}, \vec{c})$  určuje tvar rovnoběžnostěnu – základní buňky. Délky hran základní buňky značíme  $a, b, c$  a úhly jimi sevřené  $\alpha, \beta$  a  $\gamma$ . Konvenčně je úhel  $\alpha$  svírán hranami  $b, c$ , úhel  $\beta$  hranami  $a, c$  a

úhel  $\gamma$  hranami  $a, b$ . Těchto šest hodnot ( $a, b, c, \alpha, \beta, \gamma$ ) tvoří parametry mřížky.

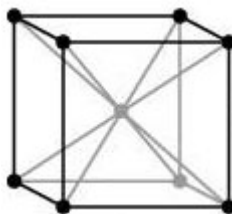
Buňky můžeme dělit podle toho, kolik mřížkových bodů připadá na jejich objem:

- Primitivní buňka (P, R) – v každém z osmi rohů je po jednom bodu, přičemž každý z těchto bodů náleží buňce z jedné osminy. Primitivní buňka tedy obsahuje jeden mřížkový bod.



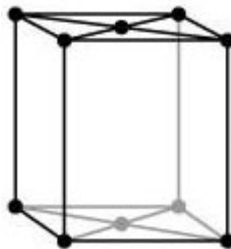
**Obrázek 1 - Primitivní buňka**

- Tělesově centrovaná (I) – má mřížkové body v rozích a navíc jeden na průsečíku tělesových úhlopříček, který do buňky patří celý. Buňka tedy obsahuje dva mřížkové body.



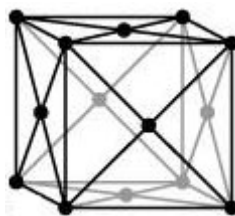
**Obrázek 2 - Tělesově centrovaná buňka**

- Bočně centrovaná (A, B, C) – má mřížkové body v rozích a ve dvou protilehlých stěnách. Tyto dva body buňce náleží z jedné poloviny, takže tato buňka dohromady obsahuje dva mřížkové body.



**Obrázek 3 - Bočně centrovaná buňka**

- Plošně centrovaná (F) – taková buňka krom bodů v rozích obsahuje body ve středech všech stěn. Těchto bodů je šest a buňce náleží z jedné poloviny, čili buňka celkem obsahuje čtyři mřížkové body.



**Obrázek 4 - Plošně centrovaná buňka**

Teoreticky existují i další centrace, ale ty mohou být převedeny na primitivní buňku nebo na výše uvedené centrace. O tom, zda je buňka primitivní či centrovaná, rozhoduje výhradně počet mřížkových bodů připadajících na její objem, ne počet atomů.

Krystalová mřížka je periodická a lze jí tedy prokládat soubory mřížkových rovin. Tyto roviny popisujeme třemi indexy ( $hkl$ ). Číselné hodnoty indexů označují, na kolik úseků dělí roviny osy  $a$ ,  $b$ ,  $c$  základní buňky. Délky těchto úseků jsou potom  $a/h$ ,  $b/k$ ,  $c/l$ . Je-li hrana dělena v záporném smyslu, je příslušný index označen pruhem. Pokud se rovina s některou z os neprotíná, je příslušný index nulový. Vzdálenost mezi rovinami nazýváme mezirovinnou vzdáleností. Kromě mřížkových rovin uvažujeme i roviny nemřížkové, které neprocházejí mřížkovými body.

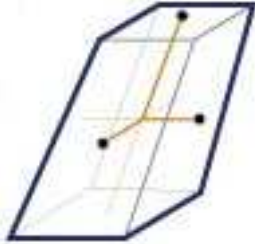
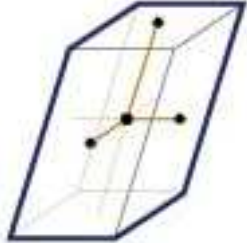
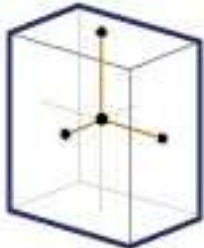
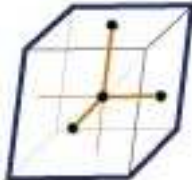
Každý vnější tvar ideálně vyvinutého krystalu a každá krystalová struktura vykazuje souměrnost, kterou popisujeme operacemi a prvky symetrie. Operací nazýváme každý pohyb, kterým dostaneme objekt do polohy nerozlišitelné od výchozí polohy. Operace symetrie provádíme pomocí geometrických prvků (přímka, rovina, bod), které nazýváme prvky symetrie. Konkrétně hovoříme o ose, rovině nebo středu symetrie a říkáme, že objekt je podle těchto prvků symetrický.

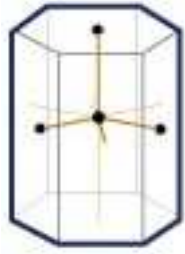


**Tabulka 1 - Operace a prvky symetrie**

Operace souměrnosti	Prvek souměrnosti	Symbol
Zrcadlení	Rovina (zrcadlo)	m
Otáčení (rotace)	Osa	2, 3, 4, 6
Inverze	Střed	$\bar{1}$
Rotační inverze	Rotačně inverzní osa	$\bar{3}, \bar{4}, \bar{6}$
Translace	Přímka	žádný
Sklužný pohyb (zrcadlení-translace)	Sklužná rovina	a, b, c, n, d
Šroubový pohyb (otáčení-translace)	Šroubová osa	$2_1, 3_1, 3_2, 4_1, 4_2, 4_3, 6_1, 6_2, 6_3, 6_4, 6_5$

Celkem existuje čtrnáct typů prostorových mřížek

Tabulka 2 - typy prostorových mřížek

Název	Varianty	Popis	Obrázek
Trojklonná (triklinická)	P	$a \neq b \neq c$ $\alpha \neq \beta \neq \gamma \neq 90^\circ$	
Jednoklonná (monoklinická)	P, C	$a \neq b \neq c$ $\alpha = \gamma = 90^\circ \neq \beta$	
Kosočtverečná (orthorombická)	P, C, I, F	$a \neq b \neq c$ $\alpha = \beta = \gamma = 90^\circ$	
Klencová (romboedrická)	R	$a = b = c$ $\alpha = \beta = \gamma < 90^\circ$	

Šesterečná (hexagonální)	P	$a = b \neq c$  $\alpha = \beta = 90^\circ; \gamma = 120^\circ$	
Čtverečná (tetragonální)	P, I	$a = b \neq c$  $\alpha = \beta = \gamma = 90^\circ$	
Krychlová (kubická)	P, I, F	$a = b = c$  $\alpha = \beta = \gamma = 90^\circ$	

Každý objekt v krystalografii má jeden nebo více prvků symetrie. Symetrii objektu pak popisujeme pomocí grupy operací symetrie. Grupa je matematický pojem – je to množina prvků, pro které platí, že kombinace dvou prvků grupy je také prvkem grupy. V krystalografii jsou symetrické objekty vnější tvar krystalu, skupina atomů (molekula nebo základní motiv), Bravaisova mřížka a krystalová struktura. V případě tvaru krystalu, Bravaisovy mřížky a skupiny atomů používáme kategorii bodová grupa, pro popis symetrie krystalové struktury používáme prostorové grupy. Bodová grupa znamená, že jde o grupu uzavřených operací souměrnosti (tj. zrcadlení, rotace, inverze a rotační inverze), po jejichž provedení zůstane alespoň jeden bod objektu nepohyblivý. Deset operací a 22 jejich kombinací představuje dohromady celkem 32 krystalografických bodových grup.

Pro popis krystalové struktury používáme prostorové grupy. V těch se uplatní

jak zavřené operace (zrcadlení, rotace, inverze a rotační inverze) tak operace otevřené (translace, skluzný pohyb, šroubový pohyb). Prostorová grupa vystihuje symetrii periodicky se opakujícího vzoru v prostoru. Na rozdíl od bodových grup, kde se prvky symetrie protínaly alespoň v jednom bodě, jsou prvky souměrnosti prostorových grup rozloženy v prostoru. Kombinacemi otevřených a zavřených operací souměrnosti spolu s translacemi čtrnácti Bravaisových mřížek dostaneme celkem 230 prostorových grup.

S pojmem prostorové grupy je úzce spojen pojem ekvivalentní polohy. Pokud chceme vyjádřit polohu nějakého atomu v buňce, používáme k tomu frakční souřadnice  $x; y; z$ , které představují zlomky hran buňky. Souřadnice  $0,5; 0,25; 1/3$ , znamenají, že popisovaný atom má souřadnici  $x$  v polovině hrany  $a$ , souřadnici  $y$  v jedné čtvrtině hrany  $b$  a souřadnici  $z$  v jedné třetině hrany  $c$ . Polohu bodu v buňce můžeme vyjádřit pomocí polohového vektoru  $\vec{r} = x\vec{a} + y\vec{b} + z\vec{c}$ . Operacemi symetrie dostáváme atomy, jejichž poloha je ekvivalentní k výchozí poloze  $x, y, z$ . Tuto novou polohu pak vyjadřujeme pomocí původní polohy. Např. vezmeme-li se prostorovou grupu  $P2/m$  (primitivní buňka, jedna dvojčetná osa a na ni kolmá rovina zrcadlení) a počátek souřadného systému umístíme do průsečíku osy a roviny, tak původní poloha je  $x, y, z$ , ekvivalentní polohy jsou  $\bar{x}, y, \bar{z}$ ,  $x, \bar{y}, z$  a  $\bar{x}, \bar{y}, \bar{z}$ .

#### 2.1.4 Rentgenové záření

Základní metodou pro určování struktury krystalů je rentgenová strukturní analýza. Rentgenové záření se používá proto, že jeho vlnová délka je řádově shodná s meziatomovými vzdálenostmi v krystalu. Díky tomu vzniká při průchodu rentgenového záření krystalem difrakční obrazec, ze kterého se při rentgenové strukturní analýze vychází. Rentgenový paprsek je elektromagnetické záření s vlnovou délkou řádově  $10^{-10} m$ . Vzniká v Roentgenových lampách (rentgenkách) při dopadu urychlených elektronů na kovovou překážku. Rentgenka obsahuje dvě elektrody. Katoda je žhavená wolframová spirála, která termoemisí uvolňuje

elektrony. Anoda je kovová destička vyrobená většinou z mědi, molybdenu, kobaltu, železa nebo chromu. Mezi elektrodami je udržován vysoký potenciálový rozdíl, čímž se dosahuje urychlení emitovaných elektronů. Jedním z výsledných efektů srážky elektronů s anodou je emise rentgenového záření, které vystupuje z rentgenky beryliovými okénky, které jsou pro rentgenové záření dobře propustné. Uvnitř rentgenky je vakuum, aby nedocházelo ke zbytečným srážkám elektronů s molekulami plynu. Účinnost celého procesu je velmi nízká (cca 1%), většina energie přechází v odpadní teplo, takže dochází k přehřívání anody, která tak musí být chlazena vodou.

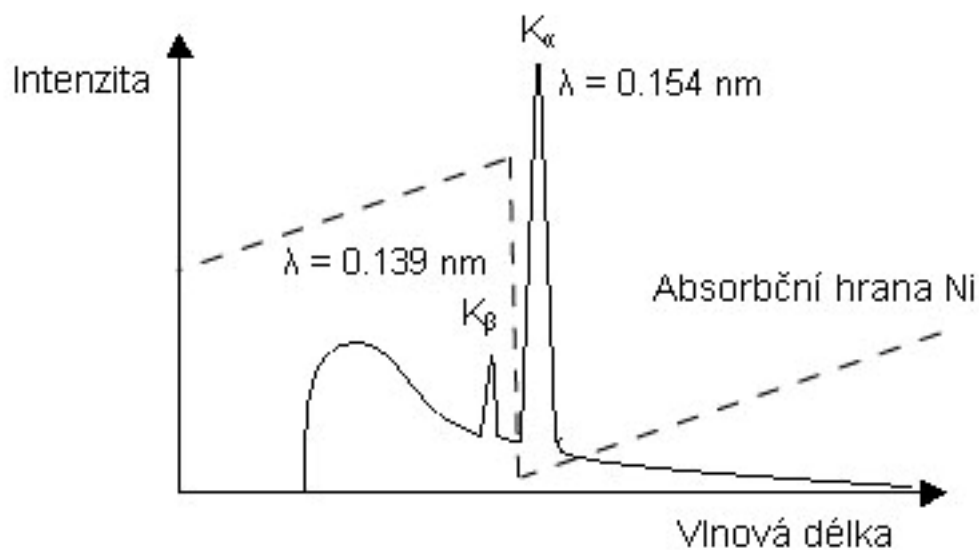
Při nižším potenciálovém rozdílu mezi elektrodami je vlnové spektrum rentgenového záření spojitě. Při určitém budícím napětí na rentgence však vzniká charakteristické čárové spektrum. K tomu dochází proto, že elektrony mají již takovou energii, že vyrážejí elektrony z energetických hladin v atomech anody a dochází k vyplnění vzniklých vakancí přeskoky elektronů z vyšších energetických hladin.

Při průchodu rentgenových paprsků látkou se tento svazek zeslabí, protože dochází k rozptylu a absorpci. Při rozptylu je část rovnoběžných paprsků rozptýlena do stran. Při absorpci dochází ke srážkám rentgenových fotonů s atomy absorbující látky. Pokud je vlnová délka záření dostatečně krátká (resp. má-li záření dostatečnou energii), dochází k vyrážení elektronů z atomů absorbující látky a jejich následným zaplňováním elektrony z vyšších energetických hladin, což se navenek projevuje jako vlastní záření absorbující látky (fluorescence). Díky tomu dochází ke skokovému zvýšení absorpce. Zeslabení intenzity záření pomocí rozptylu je zanedbatelné ve srovnání s efektem absorpce, proto můžeme výsledné zeslabení svazku přičíst pouze absorpci. Lze jej popsat vztahem

$$I = I_0 \exp(-\mu x)$$

Kde  $I$  je intenzita svazku zeslabeného absorpcí,  $I_0$  je intenzita dopadajícího

záření,  $x$  je tloušťka vrstvy absorbující látky a  $\mu$  je lineární absorpční koeficient, který však závisí na chemickém a fyzikální stavu látky. V praxi se zavádí hmotnostní absorpční koeficient, který závisí jen na látce samotné. Hmotnostní absorpční koeficient klesá spolu s klesající vlnovou délkou. Při vlnové délce, kterou nazýváme absorpční hrana, dojde k prudkému nárůstu hmotnostního absorpčního koeficientu, který pak opět pozvolna klesá.



**Obrázek 5 - Filtrace rentgenového záření niklovým filtrem**

Při rentgenostrukturní analýze je nutné, aby rentgenový svazek byl co nejdokonalěji monochromatický, k čemuž lze využít efekt absorpční hrany. Do cesty rentgenovému paprsku je vložen materiál, jehož absorpční hrana se nachází v takovém místě, aby se čárová část spektra rentgenového záření nacházela těsně před absorpčním minimem, a dojde tak k odfiltrování spojité části spektra.

Rentgenové záření má velice neblahý vliv na lidské zdraví, při kontaktu intenzivního záření s pokožkou dochází k obtížně hojitelným popáleninám a při dlouhodobém vystavení vyvolává patologické změny v organismu. K ochraně zdraví pracovníků rentgenových laboratoří se využívá dokonalé absorpce rentgenového

záření těžkými kovy. Nejčastěji se používají několikamilimetrové olověné pláty.

### 2.1.5 Strukturní analýza krystalů

Při strukturní analýze krystalů se využívá difrakce rentgenového záření při průchodu krystalem<sup>3</sup>. Vlna primárního rentgenového záření dopadající na krystal rozkmitá elektrony krystalu se stejnou frekvencí, jako má sama. Atomová jádra jsou mnohem hmotnější než elektrony, takže zůstávají v klidu. Kmitající elektrony se stávají zdrojem nového (sekundárního) záření se stejnou vlnovou délkou jako primární záření.

Sekundární záření se šíří do prostoru v kulovitých plochách a jednotlivé vlny spolu interferují. Při tomto ději je důležité, v jaké jsou vlny fázi, zda se vlny budou navzájem posilovat či rušit. Výsledek celého procesu je soubor difraktovaných vln, které se šíří směrem od krystalu a pouze v určitých směrech. Prostorové rozložení difraktovaných vln tvoří difrakční obrazec krystalu.

Každá difraktovaná vlna je charakterizována třemi parametry:

- Difrakční úhel – úhel, který svírá směr šíření difraktovaného záření s primárním zářením
- Amplituda vlny
- Fázový úhel

Otáčíme-li plynule krystalem, na který dopadá monochromatický rentgenový svazek, tak při určitém natočení zaznameneáme difraktovaný paprsek. Z hlediska výsledného efektu je difrakce analogická reflexi. Přestože je mechanismus reflexe a difrakce odlišný, chová se krystal vůči rentgenovému záření stejně, jako by docházelo k reflexi. Zvykem bývá hovořit o difrakci jako o reflexi na souboru rovin ( $hkl$ ). Této podobnosti využil Bragg k odvození Braggovy rovnice (nebo též Braggovy difrakční podmínky). Ta nám říká, jaký musí být úhel dopadajícího záření, aby docházelo k reflexi, a je pomocí ní možné spočítat mezirovinnou vzdálenost.

Difrakce probíhá na difraktometru<sup>4</sup>. Krystal je umístěn na goniometrické hlavičce. Ta se skládá ze dvou na sebe kolmých kolébek a dvou na sebe kolmých sáněk. Kolébky umožňují naklánění krystalu, zatímco sánky umožňují krystal posouvat. Celý tento mechanismus je velice precizní a je schopen krystal natáčet s přesností zlomků úhlových vteřin. Záření je detekováno scintilačním detektorem difraktovaného záření a vše je řízeno počítačem. Krystal je pokaždé natočen tak, aby záření procházelo jednou mřížkovou rovinou. Cílem je naměřit stovky až tisíce reflexí. Celý proces je velice časově náročný – trvá až několik dní.

Výsledkem následných výpočtů je třírozměrná mapa elektronových hustot, ze které již lze získat geometrické parametry atomů v buňce.

## **2.2 Krystalografické databáze**

Krystalografických dat, ať už naměřených či spočítaných existuje v současné době již velká spousta. Je tedy nezbytné tato data uchovávat v nějaké strukturované formě, ve které nebude problém vyhledat určité struktury podle různých kritérií. Tento problém řeší krystalografické databáze. Ty se většinou specializují na určitou oblast krystalografie (minerály, organické a organometalické molekuly, biologické makromolekuly atd.). Některé databáze jsou zdarma a jiné jsou placené.

### **2.2.1 Crystallography Open Database**

Tato databáze byla založena v únoru 2003<sup>5, 6</sup>. Její primární účel byl sjednotit databáze, které se specializovaly zvláště na anorganické krystaly, zvláště na kovy a slitiny a zvláště na organické a organometalické molekuly. Tato databáze postupně získávala data z různých specializovaných databází a v současné době lze i zasílat vlastní struktury ve formátu CIF, REF či PDF. Uživatelské rozhraní databázového stroje (použita je MySQL<sup>7, 8</sup>) je implementováno jazyce PHP<sup>9, 10</sup>. Databáze nyní obsahuje něco přes 140000 záznamů.

### **2.2.2 American Mineralogist Crystal Structure Database**

V této databázi lze vyhledat všechny struktury, které byly publikovány v časopisech American Mineralogist, The Canadian Mineralogist, European Journal of Mineralogy and Physics a Chemistry of Minerály<sup>11, 12</sup>. Dále zde lze nalézt některé vybrané struktury z dalších časopisů. Databáze je pod správou Mineralogical Society of America a Mineralogical Association of Canada a je financována National Science Foundation. Databáze je volně přístupná z webu Mineralogical Society of America a University of Arizona.

### 2.2.3 Databáze NIST

NIST (National Institute of Standards and Technology) provozuje celkem tři krystalografické databáze označené jako SRD (Standard Reference Database)<sup>13</sup>. Databáze SRD 3 (NIST Crystal Data) je limitována pouze na parametry jednotkových buněk a fázová složení a obsahuje asi 250.000 záznamů. Databáze SRD 83 (NIST Metals Structural Database) obsahuje kompletní krystalografické a atomové pozice celkem asi 60.000 metalických a intermetalických materiálů a databáze SRD 84 (FIZ-NIST Inorganic Crystal Structure Database) obsahuje kompletní krystalografické a atomové pozice 100.000 anorganických nemetalických materiálů. NIST neustále vyvíjí nové algoritmy, které ověřují platnost krystalografických dat, udržují všechny tři databáze a poskytují tyto databáze v různých formátech. Přístupy k těmto databázím jsou placené.

## 2.3 Datové formáty

Krystalografických dat je v dnešní době ohromné množství a je tedy žádoucí tato data v nějaké strukturované, jednoznačné a pokud možno jednoduché formě uchovávat a šířit. V oblasti výpočetní techniky pro to existuje hned několik přístupů.

### 2.3.1 STAR

Mnoho počítačových archivních postupů využívá tzv. „fixed format“ datové struktury, tedy soubory obsahující určité informace na předem určených pozicích<sup>14</sup>. Tato metoda je velmi efektivní z hlediska přístupu k těmto datům, avšak postrádá flexibilitu – není možné provádět změny ve struktuře takovýchto souborů bez konverze starších dat. Proto není vhodné používat takovéto formáty pro dlouhodobé uchování dat.

Další možností jsou tzv. „předdefinované svobodné formáty“ (pre-defined free formats). Tyto již nevyžadují, aby byla data na určitých pozicích a často obsahují datové klíče, které pomohou s rozpoznáváním dat. Příkladem takových formátů je

např. BCCAB archive file, Standard Crystallographic File Structure<sup>15</sup>, JCAMP-DX File<sup>16</sup> nebo Standard Molecular Data (SMD)<sup>17</sup>. Tyto formáty jsou však syntakticky poměrně složité a extrakce dat vyžaduje vysoké úsilí. Vysoká komplexita a nízká flexibilita těchto formátů významně limitují výměnu informací. Navíc v době vzniku tohoto formátu existovalo velké množství různým „místních“ a úzce specializovaných formátů dat, což v době velkého rozmachu internetové komunikace začalo být nežádoucí a bylo nutné tyto rozdíly nějak sjednotit. Bylo zapotřebí vyvinout datový formát, který by byl syntakticky jednoduchý a zároveň univerzální, nezávislý na platformě a lehce rozšiřitelný.

Formát STAR (Self-defining Text Archive and Retrieval) obsahuje čistě textová data, a tedy může být vytvořen a editován pomocí běžných textových editorů. STAR formát obsahuje následující prvky:

- Název dat – jméno přiřazené k hodnotě dat (např. `_absorption_coefficient`)
- Hodnota dat – konkrétní alfanumerická hodnota (např. 16.328)
- Datová smyčka – seznam opakujících se dat jako seznam naměřených intenzit
- Save frame – kolekce výše zmíněného v datovém bloku, na kterou se lze odkazovat pomocí tzv. „framecode“
- Datový blok – kolekce všeho zmíněného výše, sada informací o jedné sloučenině

STAR syntaxe:

- Textový řetězec ohraničený symboly `[ ]`, `[“ ”]`, `[;]` a `[ ]` - pokud neobsahuje mezeru (většinou je použit pro numerické hodnoty)
- Název dat – textový řetězec začínající podtržítkem
- Hodnota dat – textový řetězec nezačínající podtržítkem, kterému vždy předchází identifikátor – název dat
- Datová smyčka – vždy začíná řetězcem `,loop_‘`, následuje výpis názvů dat a po něm všechny hodnoty. Zde vždy platí, že počet hodnot dat musí být  $n$ -

násobek počtu názvů dat, kde  $n$  je nenulové přirozené číslo.

- Datový blok – kolekce dat, začíná řetězcem ‚data\_‘ spojeným s identifikátorem tohoto bloku (př. ‚data\_molecule1‘, kde ‚molecule1‘ je identifikátor)
- Všechny názvy dat musí být v jednom bloku unikátní.

Příklad jednoduchého STAR souboru:

```
data_crystal_5A5
_cell_volume          2310 (2)
_chemical_formula     'IC23 H36 O1'
_publication_author_address
;
Prof Barry O'Connell
Department of Chemistry
University of Kalamazoo
Michigan U.S.A.
;
```

Tento příklad obsahuje jediný datový blok – ‚crystal\_5A5‘ s třemi položkami (datovými názvy). Hodnoty používají různé oddělovače. Tyto hodnoty jsou „typu“ číslo, písmo a text, avšak syntaxe STAR formátu mezi těmi to typy nijak nerozlišuje.

Zde je příklad datové smyčky:

```
loop_
_exptl_crystal_face_h
_exptl_crystal_face_k
_exptl_crystal_face_l
_exptl_crystal_face_distance
_exptl_crystal_face_name
_exptl_crystal_face_description
0 0 -1 0.012 A 'well formed'
0 0 1 0.012 B -
1 0 0 0.023 C uneven
-1 0 0 0.027 D 'requires further grinding'
```

Tento příklad obsahuje smyčku o 6 položkách (názvy položek začínají podtržítkem). Po výpisu položek začínají samotná data (v tomto případě má smyčka 4 opakování). Použité formátování zde není důležité, avšak napomáhá k lepší čitelnosti. Především musí souhlasit počet definovaných položek a počet datových hodnot. Takovýto zápis je ekvivalentní:

```

loop_
_exptl_crystal_face_h _exptl_crystal_face_k _exptl_crystal_face_l
_exptl_crystal_face_distance _exptl_crystal_face_name
_exptl_crystal_face_description
0 -1 0.012 A 'well formed' 0 0 1 0.012 B - 1 0 0 0.023 C uneven -1 0 0
0.027 D 'requires further grinding'

```

Formát STAR také podporuje komentáře. Za komentář se považuje zbytek řádky za symbolem [#], pokud není součástí hodnoty dat.

Příklad komentáře:

```
_exptl_crystal_face_name # not std name
```

Zde se nejedná o komentář:

```

_publ_contact_author
;
Prof Barry O'Connell
Department of Chemistry
Building #57-M5
University of Kalamazoo
Michigan USA.
;

```

### 2.3.2 CIF

Formát CIF (Crystallography Information File) upřesňuje a omezuje formát STAR<sup>18</sup>. Veškeré restriktce formátu STAR jsou uvedeny výše. Nejsou tu žádná omezení co se týče např. počtu vnořených smyček či délky záznamů (např. délka názvu dat). Z důvodu zjednodušení a ulehčení zpracování CIF souborů bylo přikročeno k zavedení následujících omezení:

- Řádky mohou mít maximálně 80 znaků.
- Názvy dat nesmí překročit 32 znaků a nerozlišují se malá a velká písmena (, \_AbC‘ je to samé jako ,\_abc‘).
- STAR formát nerozlišuje mezi různými datovými typy. CIF také ne, ale pro zpracování je žádoucí, když je datový typ znám předem. Některé specifické názvy dat pro CIF očekávají číselné hodnoty.
- Hodnota dat je považována za číslo začíná-li číslicemi 0-9 nebo symboly [+], [-] nebo [.] a není uzavřena symboly [‘], [“] nebo [;].

- Číselná hodnota může být celé číslo, desetinné číslo nebo matematický zápis. Pokud je číslo zřetězeno s celým číslem v závorkách, je toto uzávorkované číslo bráno jako směrodatná odchylka posledních číslic. Např. 34.5, 3.45E1, 34.5(12), 3.45E1(12) jsou čtyři zápisy čísla 34,5 s nebo bez směrodatné odchylky 1,2.
- Hodnota dat je považována za text, pokud jde přes více řádků, tedy začíná a končí symbolem středníku
- Pokud hodnota není typu text ani číslo, pak je typu písmo.
- Nejsou povolené vnořené smyčky. Další úrovně opakujících se dat musí být uloženy v seznamu jako textové pole.
- Mnoho předdefinovaných CIF numerických polí vyžaduje u hodnoty dat také jednotky. Každé takové CIF pole má předdefinované implicitní jednotky. Pokud bychom chtěli uložit data v jiných než implicitních jednotkách, připojíme tuto informaci k názvu dat. Například implicitní jednotka pro rozměr buňky je ångström. Pokud bychom chtěli tento rozměr mít v pikometrech, nahradíme název dat `‚_cell_length_a‘` názvem `‚_cell_length_a_pm‘`. Povolené jsou jen jednotky definované ve standardu CIF. Všechny implicitní jednotky (s výjimkou ångströmů) jsou podřízeny jednotkám SI a měly by se používat co možná nejvíce.

## ***2.4 Použitý software***

### **2.4.1 Python**

Python je interpretovaný, objektově orientovaný, multiparadigmatický programovací jazyk<sup>19-22</sup>. Jméno tohoto jazyka pochází od Monty Pythonova létajícího cirkusu (britský seriál skečů satirické skupiny Monty Python). Jazyk python byl navržen v druhé polovině 80. let dvacátého století. Jeho implementaci začal roku 1989 nizozemský programátor Guido Van Rossum. Současná verze Pythonu je 3.0. Ta byla poprvé uvolněna již v roce 2008. Bohužel však není zpětně kompatibilní s Pythonem 2.x, a tak některé změny z verze 3.0 byly zavedeny ve zpětně

kompatibilních verzích 2.6 a 2.7.

Tím, že Python je multiparadigmatický, je míněno, že programátor není nucen do jednoho stylu programování, ale může jich používat více a libovolně je kombinovat. Python plně podporuje objektově orientované programování a strukturované programování a také má mnoho rysů funkcionálního a aspektově orientovaného programování.

Interpreter Pythonu je implementován hned v několika jazycích. Nejznámější a nejrozšířenější implementace je v jazyce C, označovaná jako CPython. Programy v jazyce C jsou obecně vysoce výkonné, a tak i tato implementace je ve většině případů nejrychlejší. Další implementace je v jazyce Java, známá jako Jython. V této implementaci se program Pythonu překládá do javovského byte-kódu a běží na Java Virtual Machine. Výhoda této implementace je spolupráce s javovskými knihovnami. Třetí důležitá implementace pythonu je na platformě .NET pod názvem IronPython, kterou vyvíjí společnost Microsoft. Za experimentální implementaci lze považovat PyPy, což je implementace Pythonu v Pythonu. Jejím hlavním účelem je odhalovat slabá místa primární implementace Pythonu.

Python je dynamicky typovaný. To znamená, že datový typ proměnné nese její hodnota. Jinými slovy Python nezná typ proměnné, dokud se s ní nepracuje. To může v některých případech poněkud mást, ale poskytuje to programátorovi velikou volnost.

Jádro Pythonu je velmi malé. Obsahuje jen několik desítek klíčových výrazů a funkcí. Velké množství dalších funkcí a tříd je možné importovat v základních či externích modulech. Modulární architektura Pythonu zaručuje přehlednost. To kontrastuje například s jazykem PHP, jehož jádro obsahuje velké množství základních funkcí, které tak musí mít velmi dlouhé názvy.

Python klade velký důraz na jednoduchost, čitelnost a čistotu kódu. Bloky

kódu například nejsou vymezeny obvyklými složenými závorkami, ale pouze dvojtečkou a následným odsazením. To je poměrně nezvyklé a často kritizované řešení, ale výsledný kód je velice úhledný a bez velkého množství ukončujících závorek, ve kterých se často chybuje.

Python můžeme najít v mnoha aplikacích jako například Google Maps, Gmail, YouTube, používá ho NASA, objevuje se ve spoustě skriptovatelných aplikacích jako VIM, GIMP, Blender či Amarok. Často se také používá jako skriptovací jazyk pro webové aplikace. Jako webový server se nejčastěji používá Apache, který s Pythonem komunikuje obvykle přes `mod_python` či `mod_wsgi`. V Pythonu je napsáno několik webových frameworků, jako Django, TurboGears nebo web2py.

## 2.4.2 Django

Django je open source webový framework, který se drží architektury MVC<sup>23</sup>. Pro architekturu MVC (model-view-controller) jsou typické následující komponenty:

- Model (model) – strukturovaná data, v případě Django se jedná o Python objekt
- View (pohled) – pohled předvádí data modelu uživateli ve vhodné interaktivní formě. V Django je typickým výsledkem pohledu HTML stránka.
- Controller (řadič) – Řadič reaguje na události vyvolané uživatelem a na základě nich mění pohled nebo model.

Konkrétně v případě Django se také někdy mluví o architektuře MTV (model, template, view), kde view specifikuje pouze jaká data se budou uživateli zobrazovat a template určuje jak tato data budou vypadat vizuálně. V tomto případě nám ovšem vypadla komponenta ‚controller‘, kterou zde představuje samotný framework Django.

Django nabízí:

- Odlehčený webový server, určený pouze pro testovací účely, neboť postrádá zabezpečení. Jeho hlavní výhodou je, že není třeba žádné nastavování a že se pokaždé automaticky restartuje, když dojde ke změně zdrojových souborů, což je žádoucí ve vývojovém prostředí.
- Serializační a validační nástroj pro formuláře, který překládá hodnoty mezi formou vhodnou pro databázi a pro HTML.
- Cacheovací Framework
- Sadu speciálních tříd tzv. middleware, které mohou upravit dotaz kdykoli ve stádiu jeho vyřizování.
- Vnitřní komunikační systém, který umožňuje komunikaci mezi jednotlivými komponentami systému pomocí předem stanovených signálů.
- Lokalizační systém umožňující překlad všech komponent do libovolného počtu jazyků.
- Serializační systém, který umožňuje vytvářet a číst XML<sup>24</sup> a JSON<sup>25</sup>,<sup>26</sup> reprezentace Django objektů.
- Systém rozšiřující možnosti šablonového enginu.
- Rozhraní k pythonovským unit testům.

Django je v podstatě pythonovská knihovna. Po jejím nainstalování stačí pro založení projektu zadat příkaz:

```
django-admin [název projektu]
```

což nám vytvoří novou složku se zadaným názvem projektu. Složka obsahuje následující soubory:

- settings.py – soubor obsahující základní nastavení projektu. Lze zde nastavit například používanou databázi, middleware, cesty k šablonám či nainstalované aplikace v projektu.
- manage.py – skript umožňující správu konkrétního projektu. Zde lze například synchronizovat databázovou strukturu s definovanými modely, spustit webový server nebo založit novou aplikaci.

- urls.py – tento skript obsahuje globální nastavení URL a přesměrování na obsluhující funkce (ty se obvykle nachází v souborech views.py v jednotlivých aplikacích).

Novou aplikaci v projektu lze založit příkazem:

```
python manage.py startapp [název aplikace]
```

To nám v adresáři projektu vytvoří novou složku se zadaným názvem aplikace, která obsahuje tyto soubory:

- views.py – tento soubor je určený pro funkce, na které je odkazováno ze skriptu urls.py, které poskytují odpovědi na http<sup>27</sup> dotazy.
- urls.py – obsahuje další dělení URL adres a volání view funkcí.
- models.py – obsahuje definice jednotlivých datových modelů aplikace.
- tests.py – obsahuje unit testy.

Modely Django se skládají z několika polí, která obsahují informace o konkrétním objektu. V Django existuje mnoho typů polí, zde budou popsány je ty, které byly použity v praktické části:

- CharField – pole pro krátký text s definovanou maximální délkou
- DateTimeField – pole obsahující datum a čas
- TextField – pole obsahující delší text
- SlugField – v tomto poli mohou být pouze ASCII alfanumerické znaky, pomlčky a podtržítka, je ho tedy vhodné používat jako identifikátor nebo součást URL
- ForeignKey – pole odkazující na jiný objekt

Šablonový systém Django umožňuje jednoduše a efektivně vytvářet html stránky. Šablona je html dokument, který obsahuje některé speciální sekvence, které jsou Djangem interpretovány. Jedná se o konstrukce `{% %}` pro příkaz a `{{ }}` pro vložení hodnoty proměnné. Příklad:

```
<div>
{% if a > 0 %}
    {{ a }}
{% endif %}
</div>
```

V tomto příkladu je do elementu *div* vložena hodnota proměnné *a*, pokud je tato větší než 0. V Django jsou tyto šablony většinou uloženy zvlášť v souborech a při odpovědi na http dotaz jsou z nich vyrenderovány html stránky a odeslány jako http odpověď.

V šablonovém systému také existuje rozšiřování šablon. V základní šabloně se určí pojmenované bloky kódu, které mohou obsahovat implicitní hodnotu. Příklad:

```
<html>
<hl>Nadpis</hl>
{% block obsah %}
    <span>Implicitní hodnota</span>
{% endblock %}
</html>
```

Rozšiřující šablona už poté jen obsahuje informaci o tom, kterou šablonu rozšiřuje a bloky, které svým obsahem nahradí bloky stejného jména v základní šabloně. Příklad:

```
{% extends „sablona.html“ %}
{% block obsah %}
    <span>Tento obsah pochází z rozšiřující šablony</span>
{% endblock %}
```

Renderuje se pak už jen rozšiřující šablona. Výsledek by potom vypadal takto:

```
<html>
<hl>Nadpis</hl>
    <span>Tento obsah pochází z rozšiřující šablony</span>
</html>
```

### 2.4.3 PyCIFRW

PyCIFRW je knihovna napsaná v Pythonu, která poskytuje podporu pro soubory ve formátu CIF<sup>28, 29</sup>. Byla vyvinuta v Australian National Beamline Facility (ANBF), které spadá pod Australian Synchrotron Research Program (ASRP). Poslední zpráva o aktivním vývoji pochází z roku 2009.

PyCIFRW používá pro parsování Yapps (Yet Another Python Parsing system), což je generátor parseru pro LL(1) gramatiky. Pozoruhodná vlastnost vygenerovaného parseru je ta, že je lidsky čitelný. Především je zde ale kladen důraz na jednoduchost a snadné použití.

Knihovnu PyCIFRW lze použít jednak k parsování hotových CIF souborů, ale také k vytváření nových CIF souborů. Výsledek parsování je objektový model CIF souboru, který se ale navenek chová jako pythonovský slovník. K datům lze tedy přistupovat stejně, jako by byla uložena ve slovníku. Při vytváření nového CIF souboru se již klasicky instancují třídy balíku PyCIFRW (CifBlock, CifLoopBlock aj.).

Výstupu v CIF formátu dosáhneme pomocí konverze objektového modelu na textový řetězec pomocí funkce *str*.

#### **2.4.4 Jmol**

Jmol je open-source vizualizační nástroj pro zobrazování 3D molekul chemických sloučenin<sup>30,31</sup>. Primárně je určen pro výuku a výzkum v oblasti chemie a biochemie. Jmol je zcela zdarma, je napsán v jazyce Java, takže jej lze provozovat na platformách Windows, Linux, Unix i Mac OS. Nejzajímavější částí programu je applet, který lze lehce integrovat do HTML stránky ve webovém prohlížeči.

Jmol podporuje mnoho způsobů zobrazení. Například atomy lze zobrazit jako tyčinkový, kalotový nebo drátěný model, jako kuličky a tyčinky nebo jako stužkový model (často používaný pro zobrazování proteinů). Podpora chemických formátů je také velice široká. Patří sem třeba Protein Data Bank (pdb), Crystallographic Information File (cif), MDL Molfile (mol) nebo Chemical Markup Language (cml). Jmol tak nabízí jednodušší alternativu k pluginu Chime, který v současné době již není v aktivním vývoji<sup>32</sup>.

Pro snadnou inicializaci a ovládání appletu je připravena javaScriptová knihovna, které kromě inicializačních funkcí obsahuje také funkce generující HTML

ovládací prvky pro Jmol applet.

### 2.4.5 JavaScript

JavaScript (též známý jako ECMAScript) je interpretovaný objektově orientovaný programovací jazyk vyvinutý americkým programátorem Brendanem Eichem<sup>33</sup>. V současné době je velice oblíbený a často používaný při programování webových stránek. Kousky kódy JavaScriptu se vkládají přímo do HTML kódu stránky. Nejčastěji se jím ovládají různé prvky GUI (Graphical user interface – grafické uživatelské rozhraní) jako tlačítka atp.

Název JavaScript se zcela jasně odkazuje na programovací jazyk Java, ačkoli s ním nemá mnoho společného. Slovo Java je jeho součástí čistě z marketingových důvodů. S Javou ho spojuje snad jen podobná syntaxe, která patří do rodiny jazyků C/C++/Java.

JavaScript byl standardizován roku 1997 asociací ECMA (European Computer Manufacturers Association) a v roce 1998 ISO (International Organization for Standardization).

Kód JavaScriptu se, na rozdíl od jiných programovacích jazyků jako PHP či Python, obvykle spouští až po stažení HTML stránky – tedy na straně klienta. Z toho plynou některá bezpečnostní omezení, jako že JavaScript neumí pracovat se soubory nebo přistupovat do databáze.

JavaScript je dynamicky typovaný, takže typ proměnné nese její hodnota. Je tedy možné, aby jedna proměnná nabývala hodnot různých datových typů. Objekty jsou řešeny jako asociativní pole. Atributy objektů tvoří klíče asociativního pole, takže k hodnotám atributů objektů v JavaScriptu lze přistupovat jak přes tečkovou konvenci, tak přes zadávání klíčů pole do hranatých závorek. Funkce jsou v JavaScriptu objekty a je tedy možné s nimi nakládat jako s kterýmkoliv jiným objektem. Lze je například použít jako hodnoty asociativního pole. Je také možné

vytvářet anonymní funkce – tedy nepojmenované funkce, což se nejčastěji používá při posílání funkcí jako argumentů jiným funkcím.

## 2.4.6 JQuery

JQuery je javascriptová knihovna, která si klade za cíl usnadnit programování na straně klienta<sup>34</sup>. JQuery bylo uvolněno v lednu roku 2006 a dnes je to nejoblíbenější javascriptová knihovna. Hlavní přínos JQuery je snadný výběr elementů DOM struktury, a jejich manipulace, manipulace s kaskádovými styly, správa událostí a jednoduchá tvorba animací. Velice důležitou součástí JQuery je také AJAX, který umožňuje asynchronní tvorbu a zpracování http dotazů.

Knihovnu JQuery lze do HTML kódu vložit velice snadno stažením lokální kopie celé knihovny a vložením pomocí elementu *script*, nebo alternativně element *script* může odkazovat na externí umístění knihovny např. na serveru [googleapis.com](http://googleapis.com).

Ke knihovně přistupujeme pomocí jména *JQuery*, které je implicitně namapované také na znak *\$*. Voláním s patřičným argumentem - selektorem (který se shoduje se selektory v CSS3) dostaneme jeden nebo více DOM objektů, se kterými pak můžeme manipulovat. Přes tečkovou konvenci pak získáme přístup ke „statickým metodám“, jako je například AJAX.

Příklad kódu, kde elementu *div* s atributem *id* s hodnotu *divId*, přiřadíme kaskádový styl pro červenou barvu pozadí a bílé písmo:

```
$(„div#divId“).css({background-color : „red“,  
                    color : „white“});
```

Příklad kódu, kde do vyskakovacího okna vypíšeme obsah stránky */text.html*:

```
$.ajax({url : „/text.html“,  
        success : function(data) {  
            alert(data);  
        }  
    });
```

## 3. Praktická část

### 3.1 Základní popis projektu PrSK (Prohlížeč struktur krystalů)

Projekt je vytvořen ve webovém frameworku Django. Skládá se z následujících aplikací:

- news – jednoduchý novinkový systém. Model novinky se skládá z pole pro datum `DateTimeField`, které se automaticky vyplňuje při založení novinky, pole pro autora, které je typu `ForeignKey` a obsahuje objekt `User`, který je implicitní pro Django. Dále obsahuje pole `header` typu `CharField` s maximální délkou 100 znaků pro nadpis novinky a konečně pole `content` typu `TextField` pro obsah novinky.
- search – prohledávání externích databází. Tato aplikace neobsahuje žádný model, je zde spíše pro větší uspořádanost a přehlednost.
- structures – tato aplikace umožňuje uložit do databáze krystalové struktury v CIF formátu a přidat k nim doprovodný text. Model aplikace `Structures` obsahuje pole `name` typu `CharField` s maximální délkou 30 znaků pro název struktury, pole `description` typu `TextField` pro popis struktury a pole `model` typu `TextField`, které obsahuje CIF soubor se strukturou.
- cms – pomocí této aplikace lze spravovat statický obsah stránek. Model této aplikace obsahuje pouze identifikátor `id` typu `SlugField` a obsah `content` typu `TextField`. Důvod existence této aplikace je, aby uživatel mohl spravovat obsah statických HTML stránek, aniž by musel editovat šablony.

Ostatní funkcionality (jako 3D vizualizace krystalů) nemají vlastní aplikaci a jsou uloženy v kořenu projektu.

## 3.2 Webový výstup

Základní šablona se jmenuje *index.html*. Tento název byl zvolen spíše ze zvyku, neplatí zde pravidlo, že pokud není specifikována konkrétní stránka, implicitně se hledá *index.html*. Tato šablona obsahuje základní kostru HTML dokumentu společnou pro všechny stránky portálu.

Verze HTML byla zvolena nejnovější dostupná, tedy HTML5<sup>35, 36</sup>. Deklarace typu dokumentu je pouze:

```
<!DOCTYPE HTML>
```

Kódování bylo zvoleno univerzální utf-8<sup>37</sup>, deklarováno pomocí:

```
<meta charset="utf-8" />
```

v hlavičce HTML dokumentu.

Rozložení webové stránky je řešeno pomocí elementů *<div>* a následným přiřazením kaskádových stylů v externím souboru CSS<sup>38</sup>, který je vložen do HTML dokumentu pomocí elementu *<link>*.

HTML stránky mají nastavenou pevnou šířku. V horní části stránek se přes celou šířku rozkládá logo a nadpis portálu. Pod tímto pruhem je v levé části v užším svislém pruhu umístěna nabídka navigace a vpravo od ní je samotný obsah.

Nabídka navigace obsahuje tyto možnosti:

- Domů – úvodní stránka s uvítacím textem a novinkami
- Krystalové struktury – tato stránka obsahuje vybrané krystalové struktury s jejich 3D vyobrazením pomocí appletu Jmol a doprovodnými texty
- Prohledávat databáze – zde můžeme prohledávat externí databáze a získávat další modely krystalů
- Vizualizace – na této stránce si můžeme zobrazit libovolnou krystalovou strukturu buď nahráním CIF souboru, nebo zadáním parametrů vlastní

struktury

Základní HTML šablona obsahuje celkem 4 bloky. První se jmenuje *title* nachází v HTML hlavičce v elementu `<title>`, a tedy lze tak měnit titulek stránky pro každou konkrétní stránku. Druhý se jmenuje *scripts* a nachází se na konci HTML hlavičky. Blok *scripts* je určen pro vkládání skriptů (primárně javaScriptových), které se samozřejmě liší pro každou stránku. Další blok se jmenuje *pageHeader* a je umístěn obsahové části v elementu `<h1>` a umožňuje modifikovat nadpis každé stránky. Poslední blok se jmenuje *content* a ten obsahuje vlastní obsah stránky. Vedle základní HTML kostry jsou v projektu další šablony (pro každou stránku jedna), které rozšiřují základní šablonu a vlastně definují jednotlivé bloky základní HTML kostry.

### **3.3 Popis jednotlivých funkcionalit portálu**

#### **3.3.1 Novinkový systém**

Novinkový systém se nachází na úvodní stránce. Nad samotnými novinkami je uvítací text, který je brán z objektu *Cms* s identifikátorem *home*. Novinky jsou stránkovány a jednotlivé stránky novinek jsou do úvodní obrazovky vkládány dynamicky pomocí JavaScriptu.

Pro formátování novinek je připravena zvláštní novinková šablona. Při načtení úvodní stránky či při změně stránky novinek je pomocí AJAXu zavolán skript, který novinky vytáhne z databáze, pomocí připravené novinkové šablony je zformátuje a vytvoří seznam stránek a takto upravené novinky vrátí jako výsledek. AJAXová funkce, která si tento výsledek vyžádala, je poté vloží na patřičnou pozici do stránek.

Kvůli lepšímu optickému efektu je při kliknutí na stránku novinek zavolána JQuery funkce *fadeOut*, která postupně mění průhlednost celého bloku novinek až do ztracena. Stejně tak, je nová stránka novinek do úvodní stránky vložena skrytá a

následně je na ni zavolána funkce *fadeIn*, která naopak průhlednost postupně snižuje až k nule. Výsledný efekt je plynulý přechod jedné stránky novinek v druhou.

**Menu** **Home**

- » Domů
- » Krystalové struktury
- » Prohledávat databáze
- » Vizualizace

Vítejte na krystalochemickém webu

## Novinky

### Novinka 4

---

**Text novinky 4**  
26. duben 2011 19:33:04 Tomáš Müller

### Třetí novinka

---

A už tu máme 2 stránky  
18. duben 2011 17:08:04 Tomáš Müller

### Druhá novinka

---

Ahoj světe  
18. duben 2011 17:07:42 Tomáš Müller

1 2

Obrázek 6 - úvodní stránka portálu

### 3.3.2 Krystalové struktury

Stránka krystalové struktury obsahuje vlastně výpis všech uložených struktur v databázi a zobrazuje je v HTML formě. Funkce v souboru *views*, která obsluhuje příslušný http požadavek, v podstatě jen získá všechny modely *structures* a zformátuje je pomocí příslušné šablony.

Problematická část je zde pouze přenesení informací z CIF souboru uloženého v databázi ve formě textového řetězce do HTML kódu výsledné stránky. Pro veškeré operace s CIF formátem je zde použit modul *PyCIFRW*, který odstraňuje potíže, které vznikají při práci s CIF formátem ve formě textového řetězce, jako

používání různých znakových sad nebo různých znaků pro konec řádky<sup>39</sup>.

Informace o krystalu ve formátu CIF jsou získány z databáze jako textový řetězec a je zapotřebí z tohoto řetězce získat objekt pomocí knihovny PyCIFRW. Zde narážíme na problém, že knihovna PyCIFRW podporuje čtení vstupu pouze ze souborů nebo z internetového zdroje, ale nikoliv přímo z textového řetězce. Pro vyřešení tohoto problému je třeba se podívat přímo na implementaci načítání zdroje knihovnou PyCIFRW. Načítací procedura vypadá takto:

```
if isinstance(filename, basestring):
    filestream = urlopen(filename)
else:
    filestream = filename #already opened for us
...
text = filestream.read()
```

Nejprve se tedy zkusí, jestli zadaný argument je textový řetězec. Pokud ano, tak se předpokládá, že se jedná o cestu k souboru (nezáleží, jestli jde o soubor na místním disku nebo o soubor na internetu). Pokud se nejedná o textový řetězec, tak knihovna předpokládá, že jde o otevřený soubor a neprovádí žádnou zvláštní akci. Na výsledek prvního či druhého případu je pak zavolána metoda *read()*. Je tedy možné knihovně podstrčit nějaký objekt, na který když se zavolá metoda *read()*, tak vrátí CIF soubor ve formě textového řetězce. Byla teda vytvořena třída *FakeFile*, která si do svého atributu uloží obsah CIF souboru a která má metodu *read()*, která obsah tohoto atributu vrátí. Instance této třídy je pak předána načítací proceduře knihovny PyCIFRW.

Pokud používáme javascriptovou ovládací knihovnu pro Jmol, tak je nezbytné CIF soubor přetransformovat tak, aby ho bylo možné uložit jako javascriptový textový řetězec. V JavaScriptu totiž není možné, aby textový řetězec byl víceřádkový. Bílý znak řádkového zlomu je tedy třeba nahradit například symbolem `|'`, který je načítací knihovnou Jmolu akceptován jako nový řádek.

Všechny struktury jsou pak vyjmuty z databáze a zformátovány pomocí

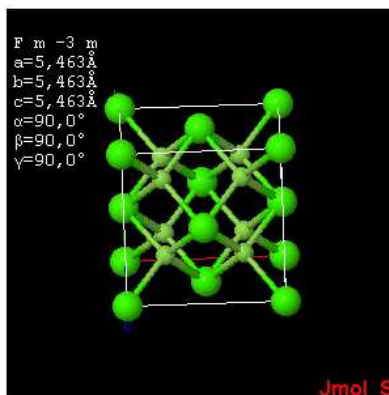
příslušné šablony. Pro každou strukturu je v HTML výstupu zobrazen její název a popis a pomocí appletu Jmol a informací o krystalové struktuře je také zobrazen její 3D model.

## Fluorid vápenatý

Bohatě formátované

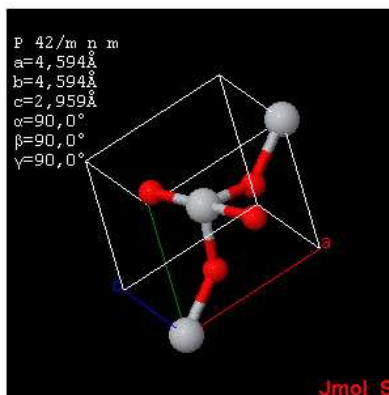
*informace o*

Chloridu vápenatém



## Oxid titaničitý

Těž známý jako titanová běloba



Obrázek 7 – výpis uložených krystalových struktur

### 3.3.3 Prohledávat databáze

Pro účel prohledávání externích databází bylo zapotřebí najít takové databáze, které se zaměřují na anorganické krystaly, jsou zdarma, mají použitelný vyhledávací systém a obsahují data v CIF formátu nebo alespoň obsahují data potřebná ke zkonstruování minimálního CIF záznamu potřebného pro vizualizaci. V současné době byly nalezeny pouze dvě databáze splňující tyto požadavky a to

Crystallography Open Database a American Mineralogy Structure Database.

Obě databáze mají samozřejmě zcela jiné vyhledávací rozhraní a používají rozdílnou syntaxi pro vyhledávací pole a také se liší HTML výstup vyhledávání, takže je nutné přistupovat k oběma databázím zcela individuálně. Vyhledávací rozhraní databáze COD nabízí hledání podle maximálně 2 klíčových slov, dále je možno zadat 1-8 prvků, které chceme hledat, 1-4 prvky, které ve výsledku nechceme, přesný počet prvků a také lze vyhledávat podle různých parametrů buňky. Databáze AMCDS nabízí stejné možnosti, jen bez výše uvedených omezení (jako maximální počet prvků) a rozšiřuje je o hledání pomocí difrakčních parametrů, symetrie, autora článku a další.

S přihlédnutím k tomu, že vytvářený portál je primárně určen pro účely výuky a ne pro vědecké účely, tak vyhledávání zde je omezeno pouze na název minerálu a na vyhledávání dle chemických prvků. Je také možné si zaškrtnout, ve kterých databázích chceme vyhledávat.

Z technického hlediska mají tyto dva portály některé rysy společné. Obě vyhledávací rozhraní jsou realizována pomocí klasických HTML formulářů. Data jsou zasílána metodou POST (data z formulářů jsou přenášena v http hlavičce místo v URL adrese). Pro uskutečnění vyhledávání tedy stačí nasimulovat http požadavek s POST proměnnými podobný tomu, jaký je vytvořen odesláním HTML formuláře a poté zpracovat HTML výsledek.

První krok je vědět, jaké proměnné jsou formulářem posílány. To lze lehce zjistit inspekcí HTML kódu formuláře. Zde já část HTML kódu vyhledávacího rozhraní AMCDS:

```
<tr>
  <td width=50%>
    <input type="text" Name="Mineral" value="" size=50 >
  </td>
  <td>
    <a href="javascript:submitForm('index_min.php');">
```

```

        class="title">Mineral</a>
    </td>
</tr>
...
<tr>
    <td><input type="text" Name="Periodic" value="" size=50 ></td>
    <td><A HREF="javascript:submitForm('periodictable.php');"
class="title">Chemistry Search</a></td>
</tr>

```

Zde nás zajímají textová pole, tedy HTML elementy *input* s atributem *type* a hodnotou *text*. Jméno proměnné, ve které se odesílá obsah tohoto pole je obsaženo v atributu *name*. Pak je také třeba zjistit, jaký skript bude náš dotaz zpracovávat. To lze zjistit z atributu *action* elementu *form* nebo jednodušeji z URL adresy výsledku vyhledávání. U databáze COD bychom postupovali analogicky.

Vyplněná pole formuláře je potřeba zpracovat pro každou databázi jinak. Pole *Název* lze do databáze AMCSD odeslat tak, jak ho vyplní uživatel. Databáze COD však vyžaduje jednotlivá slova a to maximálně 2. Zadaný řetězec je tedy rozdělen podle mezer a interpunkčních znamének a první dvě slova jsou použita pro vyhledávání a ostatní jsou ignorována.

Do pole *Prvky* napíše uživatel prvky, které chce a nechce vyhledávat. Syntaxe tohoto pole je velice jednoduchá. Nejdříve se zapisují prvky, které chceme hledat. Tento výpis je ukončen slovem ‚NOT‘ (akceptováno je i ‚not‘ či ‚Not‘), za kterým následuje výpis prvků, které vyhledávat nechceme. Prvky a slovo ‚NOT‘ jsou vyhledávány regulárním výrazem

```
(?:NOT) | (?:Not) | (?:not) | (?:[A-Z][a-z]?)
```

Konstrukce „(?:“ není součástí klasických regulárních výrazů. Znamená, že se jedná o počáteční závorku, která pouze vymezuje část regulárního výrazu a nemá jinou funkci. Primárně se snažíme najít oddělovač ‚NOT‘ ve všech akceptovaných modifikacích a hned následně symbol prvku – jedno velké písmeno nepovinně následované malým písmenem. Z dotazu „As Si NOT S Fe“ tak dostaneme seznam [„As“, „Si“, „NOT“, „S“, „Fe“], který pak lze jednoduše rozdělit na seznam chtěných a nechtěných prvků. Vzhledem k tomu, že se symboly prvků hledají regulárním

výrazem a ne dělením textu podle nějakého znaku, tak nezáleží na tom, jaký oddělovač mezi prvky použijeme.

V rozhraní COD se chtěné a nechtěné prvky vyplňují do zvláštních polí, proto jsou omezené počtem. Každý chtěný i nechtěný prvek tam má svou vlastní proměnnou. Rozhraní AMCSD má svou syntaxi, kde se množiny prvků uzavírají do závorek a před množinou nechtěných prvků je znaménko *,-'*. Zápis předchozího dotazu by tedy vypadal *(As,Si) - (S,Fe)*.

Nyní již máme vše připraveno k provedení dotazu. Dotaz budeme provádět stranou od portálu uvnitř skriptu zpracovávajícího náš dotaz. Ten provedeme pomocí pythonovských knihoven *urllib* a *urllib2*. Nejprve vytvoříme pythonovský slovník, jehož klíče budou názvy http proměnných a hodnoty slovníku budou samozřejmě hodnoty http proměnných. Funkcí *urlencode* z modulu *urllib* transformujeme pythonovský slovník do klíčů a hodnot používaných v URL. K vytvoření http požadavku pak z modulu *urllib2* použijeme třídu *Request*, která jako první argument vezme URL a jako druhý argument transformovaný slovník a tento dotaz odešleme. Výsledkem je HTML stránka, jakou bychom dostali při použití vyhledávacího rozhraní databáze.

HTML výsledek vyhledávání v databázi COD je poměrně přehledný a dobře zpracovatelný. Každá nalezená struktura je umístěna zvlášť v tabulce, která obsahuje odkaz na CIF soubor, chemický vzorec, komentář, který je v podstatě citací článku, ze kterého byla struktura získána, prostorovou grupu, objem buňky a parametry buňky. Tyto informace jsou extrahovány pomocí regulárních výrazů. Specializované nástroje pro zpracování XML nebyly použity kvůli hrozící nevaliditě dokumentu a kvůli větší rychlosti regulárních výrazů.

HTML výsledek z databáze AMCSD je oproti tomu velice špatně zpracovatelný. Všechny záznamy jsou v jediné tabulce a nerozlišené. Třeba jen samotné rozlišení jednotlivých záznamů by bylo velice nespolehlivé. Proto v tomto

případě nejsou informace o výsledcích brány z HTML stránky. Z té jsou vyjmuty pouze URL CIF souborů, které jsou staženy a zparsovány knihovnou PyCIFRW a veškeré informace jsou brány z nich. Tento postup je mnohem pomalejší, ale spolehlivější.

Všechny extrahované informace jsou zkompletovány do HTML stránky, jsou přidány odkazy na stažení CIF souboru a zobrazení struktury přímo na stránkách portálu a výsledek je vrácen ve formě JSON pole spolu s počtem nalezených záznamů.

Prohledávání externích databází

Crystallography Open Database  
 American Mineralogist Crystal Structure Database

Název:

Prvky:  ?

## Výsledky hledání

▶ **Crystallography Open Database (15)**

▼ **American Mineralogist Crystal Structure Database (9)**

**Název:** Pyrite  
**Vzorec:** Fe S<sub>2</sub>  
**Prostorová grupa:** P a 3  
**Objem buňky:** 155.721  
**Parametry buňky:** 5.38 5.38 5.38 90 90 90



**Název:** Pyrite  
**Vzorec:** Fe S<sub>2</sub>  
**Prostorová grupa:** P a 3  
**Objem buňky:** 158.921  
**Parametry buňky:** 5.4166 5.4166 5.4166 90 90 90



Obrázek 8 – Výsledky prohledávání externích databází

Vyhledávání v externích databázích trvá poměrně dlouho (řádově několik sekund) a je tedy vhodné, aby probíhalo paralelně ve všech prohledávaných databázích a na pozadí. Při stisknutí tlačítka „Hledej“ se tedy neodešle celý formulář a nečeká se, až skript vrátí výsledek, ale celé hledání probíhá v režii JQuery.

Informace z formuláře jsou na pozadí zaslány skriptům, které je zpracují a vrátí výsledek (popsáno výše). Tento výsledek je pak vložen do HTML stránky. Výsledky vyhledávání u obou databází je možno skrýt pro větší přehlednost. V závorce vedle nadpisu konkrétní databáze ve výsledcích je počet nalezených záznamů.

Během hledání v externích databázích se místo výsledků u každé databáze zobrazuje animovaný obrázek (lupa přejíždějící po krystalu). Toto může vypadat pouze jako záležitost designu, avšak tento obrázek zde plní velice důležitou psychologickou funkci. Od kliknutí na tlačítko ‚hledej‘ do zobrazení výsledků hledání může uplynout řádově několik sekund a uživateli se musí dát jasně najevo, že skutečně probíhá nějaká akce. Pokud by se několik sekund vizuálně nic nedělo, uživatel by mohl nabýt dojmu, že vyhledávání nefunguje. Po obdržení výsledků hledání a po jejich vložení do stránky je samozřejmě pohyblivý obrázek skryt.

## Výsledky hledání

▶ **Crystallography Open Database (300)**

▼ **American Mineralogist Crystal Structure Database (0)**



Obrázek 9 – probíhající vyhledávání

### 3.3.4 Vizualizace

Na této stránce je možno vizualizovat si vlastní struktury. Vizualizace funguje uživatelsky ve dvou, technicky ve třech módech. Strukturu je možno zobrazit nahráním CIF souboru. Další možností zobrazení struktury je definovat její parametry pomocí formuláře. Poslední, uživatelsky prakticky nepřístupnou

metodou zobrazení struktury je předání URL adresy CIF souboru v GET proměnné *externalCif*. Tato metoda nedisponuje žádným rozhraním a je určena pouze pro zobrazování CIF souborů z externích databází.

Základní zobrazení stránky Vizualizace je pouze s formulářem pro upload CIF souboru a bez Jmol appletu. Nad tímto formulářem je umístěn přepínač mezi formulářem pro upload a formulářem pro vytvoření struktury. Pokud během zpracování odeslaných dat dojde k chybě, je tato chyba vypsána na místo pro applet Jmol.

#### 3.3.4.1 Upload CIF souboru

Všechny tři módy vizualizace jsou zpracovány jedním skriptem. Pokud se v odeslaných datech nachází soubor, jedná se o vizualizaci pomocí uploadu souboru. Nejprve je zkontrolována velikost uploadovaného souboru. Ta nesmí přesáhnout 2 MB. Pokud soubor tento požadavek nesplní, je vrácena chybová hláška. Poté je obsah souboru parsován modulem PyCIFRW. Pokud se parsování nezdaří, je vrácena chybová hláška od modulu PyCIFRW. Pokud k žádné chybě nedojde, tak je objektový model CIF souboru převeden zpět na textový řetězec, rozdělen podle řádkových zlomů a spojen pomocí oddělovače ‚|‘. V této formě je vložen do stránky jakožto javascriptová proměnná. Tímto se aktivuje inicializace appletu a naše struktura se zobrazí.

**Nahrát strukturu / Vytvořit strukturu**

Nahrát soubor \_\_\_\_\_

Soubor nevybrán

Obrázek 10 – formulář pro upload struktury

#### 3.3.4.2 Tvorba vlastní struktury

Při použití přepínače nad formulářem pro upload se nám zobrazí formulář

pro zadání parametrů struktury. Jedná se vlastně o rozhraní, které nám umožňuje vytvořit si vlastní CIF soubor a rovnou ho zobrazit. Pro přechod z jednoho formuláře na druhý byl použit stejný optický efekt jako při přechodu z jedné stránky novinek na druhou, tedy postupná změna průhlednosti formulářů, což budí lepší dojem z plynulosti práce.

Formulář pro vytvoření vlastní struktury obsahuje následující pole:

- Prostorová grupa – ta je pro větší přehlednost rozdělena na soustavu, centraci a symetrii. Formulář je ošetřen tak, aby nebylo možné vybrat nesmyslnou kombinaci.
- Parametr buňky  $a$  – šířka buňky
- Parametr buňky  $b$  – výška buňky
- Parametr buňky  $c$  – hloubka buňky
- Úhel  $\alpha$  – úhel mezi stranou  $b$  a  $c$
- Úhel  $\beta$  – úhel mezi stranou  $a$  a  $c$
- Úhel  $\gamma$  – úhel mezi stranou  $a$  a  $b$
- Parametry atomů

Parametry atomů se dále dělí na:

- Symbol atomu
- Oxidační číslo
- Frakční souřadnici  $x$
- Frakční souřadnici  $y$
- Frakční souřadnici  $z$

Atomů v krystalu samozřejmě může být více než jeden, ideálně neomezené množství, a tento požadavek je nutné promítnout do formuláře. Formulář je zformátovaný pomocí tří tabulek. První tabulka formátuje prostorovou grupu, druhá parametry buňky a třetí atomy. Všechny parametry jednoho atomu jsou jeden řádek

tabulky a tento řádek představuje jednotku, kterou budeme chtít replikovat. Při načtení stránky je spuštěn JQuery skript, který si uloží kopii prázdné replikační jednotky. Pod tabulkou je umístěn obrázek symbolu „+“, který v této situaci plní funkci tlačítka. Po kliknutí na něj je uložená replikační jednotka opět zkopírována a vložena za poslední řádek v tabulce s parametry atomů. Tento postup můžeme opakovat dle libosti, takže je splněna podmínka neomezeného počtu atomů ve struktuře.

Pokud můžeme řádky tabulky přidávat, je rozumné mít také možnost je odebírat. Spolu s vložení prvního nového – tedy druhého řádku, se vedle nově vloženého řádku vloží také obrázek znaménka „-“, též plnícího funkci tlačítka. Po kliknutí na tento obrázek se odstraní příslušná řádka.

Při přidání nového řádku se vždy odstraní symbol „-“ z posledního řádku a objeví se u nově vkládaného řádku. Stejně tak se po odstranění řádku objeví na aktuálně posledním řádku s výjimkou případu, kde je ve formuláři pouze jeden řádek pro atom, pak není symbol „-“ nikde.

Prostorových grup je celkem 230, což není tolik, aby zde nemohly být ukázány všechny, ale je to příliš na prostý výčet. Proto je selekce prostorové grupy rozdělena na tři části – systém, centrace a symetrie. Každá z těchto skupin je závislá na té předešlé. Krystalografických soustav je konstantně 7. Každá soustava má své centrace a každá centrace má své symetrie. Celý tento systém lze poměrně jednoduše zachytit v jediné javascriptové proměnné. Tato proměnná je asociativní pole, kde klíčem je krystalografická soustava a hodnota je další asociativní pole. V tomto vnořeném asociativním poli jsou klíče symboly centrací a hodnotou je pole polí o dvou prvcích. První prvek je symetrie ve formátu jaký vyžaduje CIF soubor a druhý prvek je symetrie v HTML formátu. Zde je příklad části této proměnné – pouze pro monoklinickou soustavu, která má 2 centrace *P* a *C*:

```
var spacegroups = {monoclinic : {"P" : [{"2", "2"}, {"21", "2#8321;"},  
                                     {"m", "m"}, {"c", "c"}],
```

```

["2/m", "2/m"], ["21/m",
"2#8322;/m"], ["2/c", "2/c"], ["21/c", "2#8322;/c"]],
"C" : [{"2", "2"}, {"21",
"2#8321;"}],
["m", "m"], [{"c", "c"}],
["2/m", "2/m"], [{"21/m",
"2#8322;/m"}, {"2/c", "2/c"}, {"21/c", "2#8322;/c"}]},
...
}

```

Výpis prostorových grup byl převzat z portálu Wikipedia<sup>40</sup>. Zde se nachází velká HTML tabulka, která obsahuje kompletní krystalografický systém. Tato data bylo třeba transformovat z podoby HTML do javascriptové datové struktury. Plně automatické zpracování celé tabulky najednou by bylo velice obtížné a časově náročné. Proto byl napsán pythonovský skript pouze pro poloautomatické zpracování vždy jen jedné krystalografické třídy najednou. Výsledkem tohoto zpracování je pythonovský slovník se stejnou strukturou jako vnořené javascriptové asociativní pole. A protože se textová reprezentace pythonovského slovníku zcela shoduje se syntaxí javascriptového asociativního pole, tak nebylo potřeba výsledný slovník ani serializovat a do javascriptové proměnné mohla být vložena přímo textová reprezentace pythonovského slovníku. V sedmi krocích tak vznikla kompletní strukturovaná krystalografická soustava v javascriptovém formátu.

Nyní je potřeba tuto strukturu implementovat do grafického rozhraní formuláře. Celá implementace je psána v JavaScriptu, resp. JQuery. Krystalografický systém je konstantní množina ve formě roletového menu, a tedy se po celou dobu nemění. Výběrem krystalografického systému se okamžitě volá funkce, která mění výběr centrací. Tato funkce si nejdříve zjistí, jaký krystalografický systém byl zvolen a název tohoto systému použije jako klíč do javascriptové proměnné *spacegroups*. Iterací přes klíče získaného asociativního pole funkce získá všechny možné centrace pro tuto konkrétní soustavu. Centrace jsou pak zpracovány do HTML formy možností pro roletové menu a vloženy na příslušné místo do stránky.

Ohledně vložení centrací je volána funkce, která vypíše všechny možné symetrie pro danou kombinaci systému a centrace. Obě dvě tyto hodnoty použije

jako klíče do asociativního pole *spacegroups*, čímž získá pole dvojic symetrií – jedna v CIF formátu a druhá v HTML formátu. Pomocí těchto dvojic vytvoří množinu možností pro roletové menu symetrie, HTML formát se zobrazuje a CIF formát se odesílá pro tvorbu CIF souboru. Změnou centrace se pak už jen volá funkce vyplňující symetrie.

Některé parametry buňky jsou závislé na zvolené krystalografické soustavě. Pokud například zvolíme kubickou soustavu, tak všechny úhly jsou automaticky 90° a všechny tři rozměry buňky jsou shodné. Implementací těchto závislostí se značně ulehčí práce s formulářem.

Typy závislostí jsou zde dvě: fixace a svázanost. Graficky se projevují stejně – zašednutím některého políčka a nemožností jeho editace. Při fixaci je zašedlé políčko nastaveno na určitou hodnotu a nelze s ním nic udělat. Při svázání je hodnota zašedlého políčka svázána s hodnotou prvního nezašedlého políčka nad ním. Při změně hodnoty nezašedlého políčka se mění hodnota políčka s ním svázaného. Celkem je vytvořeno sedm funkcí (pro každou krystalografickou soustavu jedna), které mění fixace a svázanosti a jedna funkce, která vše vrací do normálu. Je pro ně vytvořeno asociativní pole, které má jako klíče krystalografické soustavy a jako hodnoty tyto funkce. Při každé změně krystalografické soustavy je nejprve volána normalizační funkce a poté jedna z funkcí měnící fixace a svázanosti.

Po odeslání formuláře dochází k jeho zpracování. Důležitá je validace hodnot, protože uživateli nic nebrání tomu, aby například jako rozměr buňky zadal nečíselnou hodnotu. Validace je rozdělena na tři části – validace rozměrů buňky, úhlů buňky a validace elementů. Pokud dojde k chybě, je na místo Jmol appletu vypsána příslušná chybová hláška např. „Neplatné rozměry buňky“ či „Neplatné úhly v buňce“.

Prostorová grupa je zadávána z roletových menu, které se dynamicky mění podle aktuálně nastavených hodnot, takže není možné zadat nesmyslnou hodnotu, a

tak není třeba prostorou grupu validovat.

Pro validaci číselných hodnot je vytvořena zvláštní funkce. Pokud uživatel hodnotu nezadá, je tato hodnota nastavena na 0,0. Znak „,“, používaný v češtině jako desetinná čárka, je nahrazen znakem „.“, desetinnou tečkou. Dále je hodnota porovnána s regulárním výrazem:

```
^\s*\d+\s*/\s*\d+\s*$
```

Tento výraz zkouší, zda se hodnota rovná výrazu [číslo] / [číslo], přičemž akceptuje případné mezery. Pokud test vyjde pozitivně, pak jsou tato dvě čísla vydělena. Toto je velice žádoucí především u zadávání frakčních souřadnic atomů, které se často nachází v např. v 1/3 nebo 1/4 některé z os, odpadá tedy nutnost psaní čísel jako 0.33333. Funkce pak vrací výsledné desetinné číslo. Stejná funkce pro validaci čísel je použita jak pro frakční souřadnice atomů, tak pro rozměry buňky i pro úhly, takže v zásadě lze používat zlomky i pro tyto hodnoty, ale není to primární účel.

Symbole prvků validovány nejsou, protože Jmol je schopen zobrazit i nesmyslné prvky.

Pokud během validace nedojde k žádné chybě, je ze zadaných informací pomocí modulu PyCIFRW vytvořen CIF objektový model, který je následně převeden na textovou reprezentaci, upraven do formy vhodné pro javascriptovou proměnnou a vložen do výsledné stránky.

Nahrát strukturu / Vytvořit strukturu

Parametry struktury

Soustava	Centrace	Symetrie		
Hexagonální	P	6 <sub>3</sub> /mmc		
Šířka buňky	3.938 Å			
Výška buňky	3.938 Å			
Hloubka buňky	17.25 Å			
Úhel alfa	90 °			
Úhel beta	90 °			
Úhel gamma	120 °			
Prvek	Ox. číslo	Souřadnice		
Cu	0	1/3	2/3	3/4
Cu	0	1/3	2/3	0.107
Se	0	1/3	2/3	1/4
Se	0	0	0	0.066



Obrázek 11 - formulář pro vytvoření vlastní struktury

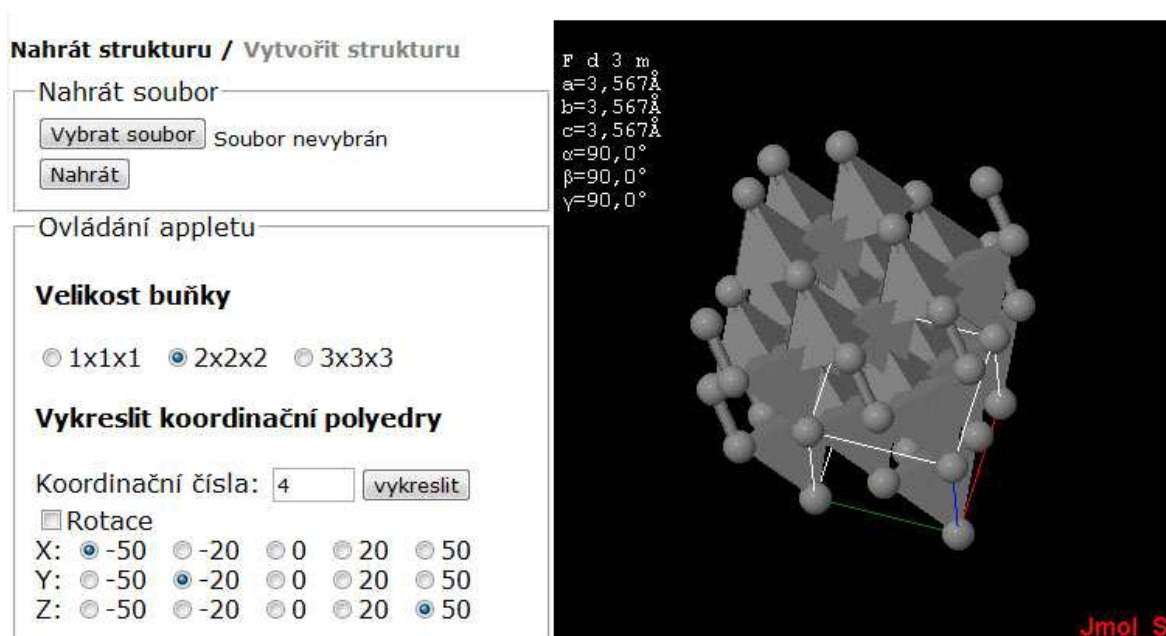
### 3.3.4.3 Zobrazení externího CIF souboru

Pokud je specifikována GET proměnná *externalCif*, je hodnota této proměnné považována za URL adresu CIF souboru. Soubor na této adrese je otevřen a prochází stejnou procedurou, jako soubor nahraný přes formulář, tedy kontrola velikosti (z http hlavičky), parsování, export do textového řetězce, úprava pro JavaScript a vložení do stránky. Tato metoda je určena primárně pro zobrazení externích CIF souborů nalezených pomocí funkce prohledávání databází.

### 3.3.4.4 Ovládání Jmol appletu

Při iniciaci Jmol appletu je také zobrazen ovládací panel, který se sestává z formulářových prvků, které ovládají zobrazený applet. V současné době je pomocí

ovládacímho panelu možné měnit násobky buňky (na výběr jsou velikosti 1x1x1, 2x2x2 a 3x3x3), vykreslení koordinačních polyedrů a nastavení rotace. Rotaci lze vypnout a zapnout a rychlost rotace podle jednotlivých směrů (x, y a z) lze nastavit na hodnoty -50, -20, 0, 20 a 50. Ovládací prvky měnící násobky buňky a rotaci byly vytvořeny automaticky pomocí javascriptové knihovny Jmol. Ovládací prvek vykreslující koordinační polyedry musel být vytvořen ručně. Skládá se z textového pole, kam se zapíše koordinační číslo a tlačítka „vykreslit“. Po stisknutí tlačítka je zavolán konzolový příkaz Jmolu „polyhedra [hodnota z pole]“, který vykreslí koordinační polyedry pro dané koordinační číslo.



Obrázek 12 – inicializovaný applet s ovládacím panelem

### 3.4 Administrace portálu

Pro administraci portálu byl použit administrační systém frameworku Django. Ten nám umožňuje spravovat definované modely, které tak můžeme vytvářet, upravovat a mazat. Administrační prostředí lze v poměrně velkém rozsahu upravovat, základní nastavení ve většině případů nebývá dostačující.

Administrační prostředí je nejprve nutno zapnout v seznamu nainstalovaných aplikací v souboru `settings.py`. Dále je nutné se do něj dostat přes webové rozhraní, a tedy musí být specifikována URL adresa v souboru `urls.py` v kořenu projektu. Zadáním příslušné URL adresy do webového prohlížeče se pak dostaneme do webového rozhraní Django administrace. Uživatel, pod kterým se lze poprvé přihlásit, se definuje při vytváření databáze její první synchronizací s modely nainstalovaných aplikací.

V úplně základním nastavení lze v Django admin spravovat pouze uživatele a skupiny. Ty se nachází v aplikaci *auth*, která je součástí jádra Django. Ostatní aplikace musí být do adminu registrovány v souboru *admin.py* ve složce aplikace. Soubor *admin.py* slouží k úpravě administračního prostředí konkrétní aplikace. Upravit zde lze například načítané skripty v HTML hlavičce, zobrazení výpisu objektů, pole, která je možno editovat při vytváření a změně objektů, ukládací procedura a jiné.

Hromadnou změnou společnou pro všechny modely, které obsahují pole typu `TextField`, bylo zavedení WYSIWIG editoru namísto prostého textového pole. Tato textová pole se obvykle vkládají do stránky jako HTML text. Užití WYSIWYG<sup>41</sup> editoru je jediná možnost, jak uživateli, který neovládá HTML, umožnit text formátovat. Práce s takovým editorem se pak podobá práci s textovým procesorem jako například MS Word. Použitý editor se jmenuje TinyMCE<sup>42</sup> a je zcela zdarma. Editor TinyMCE je napsán v JavaScriptu a do stránky se tedy zavádí stejně jako každá jiná javascriptová knihovna pomocí tagu *script*. Knihovna se uvádí do provozu inicializační funkcí, které se pošlu parametry pro WYSIWYG editor. Editor je velice flexibilní, všechna tlačítka jsou nastavitelná, podporuje použití zásuvných modulů a umožňuje i přímou editaci HTML kódu.

### 3.4.1 Novinkový systém

V novinkovém systému chceme upravovat pouze nadpis a obsah novinky.

Datum se vyplní automaticky při vytvoření novinky a autor novinky bude aktuálně přihlášený uživatel. Pole určená k editaci jsou tedy omezena na pole *header* a *content*. Při výpisu všech objektů se zase zobrazují pole *header*, *author* a *date*. Pole *content* může být poměrně velké a do výpisu se nehodí.

Pro vložení autora je nutno editovat funkci, která model ukládá. Pokud ukládaný objekt nemá autora (tedy jedná se o nově vytvářenou novinku), tak se z http požadavku vezme aktuálně přihlášený uživatel a ten je nastaven coby autor novinky. Poté se objekt uloží.

### **3.4.2 Krystalové struktury**

Pro administraci této aplikace bylo nutno změnit implicitní typ pole. CIF soubory se do databáze ukládají jako textový řetězec, ale je žádoucí, aby bylo možné tyto soubory do databáze nahrávat pomocí formuláře pro upload souborů. Proto je potřeba vytvořit si vlastní formulář, kde místo textového pole bude možno nahrát soubor. Je také potřeba změnit ukládací metodu. V této metodě se nejprve zkontroluje, zda byl nahrán soubor. Pokud ano, tak je tento soubor otevřen, je přečten jeho obsah a teprve ten je uložen do databáze namísto souboru samotného.

Pole pro výpis objektů jsou omezena jen na *id* a *name*.

## structures: změnit

Historie

Název:

Popis:

**B** *I* U ABC | HTML

Těž známý jako titanová běloba

Model:  Soubor nevybrán

Obrázek 13 – Django admin: editace modelu ‚structures‘

### 3.5 Nápovědy

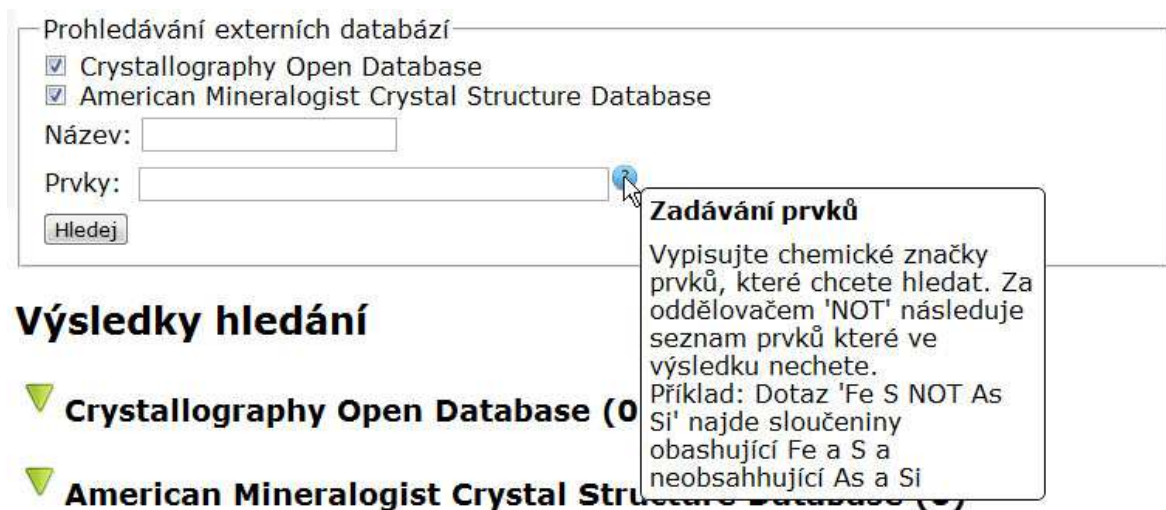
Uživatelské nápovědy jsou v portálu realizovány pomocí javascriptové knihovny *boxover* (dnes již neexistující projekt). Tato knihovna umožňuje z atributů *title* u různých HTML elementů vytvořit plovoucí okna, která jdou stylovat pomocí kaskádových stylů. Syntaxe knihovny je následující:

```
<div title="cssbody=[bClass] cssheader=[hClass] header=[Text v hlavičce nápovědy] body=[Text v těle nápovědy] fade=[on] fadespeed=[0.05]">Po najetí myši na tento text ze zobrazí nápověda</div>
```

Hodnoty *cssbody* a *cssheader* určují, jako hodnotu atributu *class* bude mít hlavička a tělo vzniknuvšího okna. Toho pak lze využít při aplikaci kaskádových stylů. Hodnota *fade* specifikuje, zda se okno postupně objeví a hodnota *fadespeed* určuje, jak rychle se toto okno bude objevovat.

V portálu jsou tyto nápovědy použity ve formě obrázku otazníku na příslušném místě a po najetí myši na tento otazník se zobrazí kýžená nápověda. Tyto nápovědy jsou použity například ve vyhledávání v externích databázích u políčka

pro zadávání symbolů chemických prvků, aby uživatel byl informován o možnosti specifikace chtěných a nechtěných prvků nebo ve vizualizačním nástroji u zadávání parametrů struktury, protože by nemuselo být jasné, že frakční souřadnice lze zadávat pomocí zlomku.



Prohledávání externích databází

- Crystallography Open Database
- American Mineralogist Crystal Structure Database

Název:

Prvky:

Hledej

**Zadávání prvků**

Vypisujte chemické značky prvků, které chcete hledat. Za oddělovačem 'NOT' následuje seznam prvků které ve výsledku nechete.

Příklad: Dotaz 'Fe S NOT As Si' najde sloučeniny obsahující Fe a S a neobsahující As a Si

## Výsledky hledání

- ▼ Crystallography Open Database (0)
- ▼ American Mineralogist Crystal Structure Database (0)

Obrázek 14 - zobrazená nápověda

## 4. Závěr

Ve frameworku Django byla vytvořena webová aplikace pro podporu výuky krystalografie. Velký důraz byl kladem na jednoduchost používání a přehledost. Každý uživatel zde získá informace o základních krystalových strukturách a je možné zobrazit si vlastní strukturu v CIF formátu. Aplikace navíc obsahuje nástroj, který umožňuje zobrazit krystalovou strukturu pouze zadáním jejích parametrů.

Na internetu existuje několik veřejně přístupných databází obsahujících krystalografická data. Výsledná aplikace obsahuje rozhraní pro vyhledávání dat ve dvou z těchto databází. Do budoucna je možné připojovat i další databáze.

Produkční verze portálu bude veřejně dostupná na stránkách VŠCHT a je počítáno i s dalším jeho vývojem.

## 5. Použitá literatura

- [1] KRATOCHVÍL, Bohumil; JENŠOVSKÝ, Lubor. *Úvod do krystalochemie*. Praha : SNTL, 1987. 240 s.
- [2] *University of Cambridge* [online]. c2006, 2008 [cit. 2011-05-01]. Lattice Planes and Miller Indices. Dostupné z WWW: <[http://www.doitpoms.ac.uk/tlplib/miller\\_indices/index.php](http://www.doitpoms.ac.uk/tlplib/miller_indices/index.php)>.
- [3] VALVODA, Václav; POLCAROVÁ, Milena; LUKÁČ, Pavel. *Základy strukturní analýzy*. 1. vydání. Praha : JP, 1992. 492 s. ISBN 80-7066-648-X.
- [4] Difrakční metody : Strukturní analýza přírodních látek. In *Speciální analýza přírodních látek* [online]. Praha : VŠCHT Praha [cit. 2011-05-01]. Dostupné z WWW: <<http://www.vscht.cz/lam/new/SpAnalPL-12.pdf>>.
- [5] GRAŽULIS, Saulius, et al. Crystallography Open Database – an open-access collection of crystal structures. *J. Appl. Cryst.* 2009, 42, s. 726-729. Dostupný také z WWW: <<http://journals.iucr.org/j/issues/2009/04/00/kk5039/kk5039.pdf>>. doi:10.1107/S0021889809016690.
- [6] *Crystallography Open Database* [online]. [cit. 2011-05-02]. Dostupné z WWW: <<http://www.crystallography.net/>>.
- [7] *MySQL* [online]. c1997 [cit. 2011-05-03]. What is MySQL?. Dostupné z WWW: <<http://dev.mysql.com/doc/refman/5.6/en/what-is-mysql.html>>.
- [8] DUBOIS, Paul. *MySQL : Developer's Library*. 4th ed. USA : Pearson Education, 2009. 1224 s. ISBN 978-0672329388.
- [9] *PHP* [online]. c2001 [cit. 2011-05-03]. What can PHP do?. Dostupné z WWW: <<http://www.php.net/manual/en/intro-whatcando.php>>.

- [10] LERDORF, Rasmus; TATROE, Kevin; MACINTYRE, Peter. *Programming PHP*. Second Edition. Sebastopol, California : O'Reilly, 2006. 521 s.
- [11] DOWNS, Robert T.; HALL-WALLACE, Michelle. The American Mineralogist crystal structure database. *American Mineralogist*. 2003, 88, s. 247–250. Dostupný také z WWW: <[http://rruff.geo.arizona.edu/xtal/group/pdf/am88\\_247.pdf](http://rruff.geo.arizona.edu/xtal/group/pdf/am88_247.pdf)>.
- [12] *American Mineralogist Crystal Structure Database* [online]. 2003, 2008 [cit. 2011-05-02]. Dostupné z WWW: <<http://rruff.geo.arizona.edu/AMS/amcsd.php>>.
- [13] *NIST* [online]. 2008, 2010 [cit. 2011-05-02]. Crystallographic Databases. Dostupné z WWW: <[http://www.nist.gov/mml/ceramics/structure\\_determination/crystallographic-databases.cfm](http://www.nist.gov/mml/ceramics/structure_determination/crystallographic-databases.cfm)>.
- [14] HALL, Sydney R. The STAR File: A New Format for Electronic Data Transfer and Archiving. *J. Chem. Inf. Comput. Sci.*. 1991, 31, 2, s. 326–333. doi:10.1021/ci00002a020
- [15] BROWN, I. D. The Standard Crystallographic File Structure. *Acta Crystallogr.*. 1988, A44, s. 232.
- [16] MCDONALD, R. S.; WILKS, P. A. Jr. JCAMP-DX: A Standard Form for Exchange of Infrared Spectra in Computer Readable Form. *Appl. Spectrosc.*. 1988, 43, s. 151-162.
- [17] BARNARD, J. N. Draft Specification for Revised Version of the Standard Molecular Data (SMD) Format. *J. Chem. Inf. Comput. Sci.*. 1990, 30, s. 81-96.
- [18] HALL, Sydney R.; ALLEN, Frank H.; BROWN, David. The Crystallographic Information File (CIF): a new standard archive file for crystallography. *Acta Crystallographica*. 1991, A47, s. 655–685.

doi:10.1107/S010876739101067X.

- [19] *Python* [online]. c1990 [cit. 2011-05-01]. About Python. Dostupné z WWW: <<http://python.org/about/>>.
- [20] VAN ROSSUM, Guido. Python for Unix/C Programmers. 1993, s. 1-8. Dostupný také z WWW: <<ftp://ftp.lip6.fr/pub/python/doc/nluug-paper.ps>>.
- [21] *The History of Python : A series of articles on the history of the Python programming language and its community*. [online]. c2004 [cit. 2011-05-01]. A Brief Timeline of Python. Dostupné z WWW: <<http://python-history.blogspot.com/2009/01/brief-timeline-of-python.html>>.
- [22] PILGRIM, Mark. *Dive into Python* [online]. 2004 [cit. 2011-05-03]. Dostupné z WWW: <<http://diveintopython.org/>>.
- [23] *Django* [online]. c2005 [cit. 2011-05-01]. Dostupné z WWW: <<http://www.djangoproject.com/>>.
- [24] *W3C* [online]. c1996 [cit. 2011-05-06]. Extensible Markup Language. Dostupné z WWW: <<http://www.w3.org/XML/>>.
- [25] *JSON* [online]. [cit. 2011-05-04]. Introducing JSON. Dostupné z WWW: <<http://www.json.org/>>.
- [26] *Adaptive Path* [online]. 2005 [cit. 2011-05-03]. Ajax: A New Approach to Web Applications. Dostupné z WWW: <<http://www.adaptivepath.com/ideas/e000385>>.
- [27] *Hypertext Transfer Protocol*. The Internet Society, 1999. 114 s. Dostupné z WWW: <<http://tools.ietf.org/pdf/rfc2616.pdf>>.
- [28] HESTER, J. R. A validating CIF parser: PyCIFRW. *J. Appl. Cryst.* 2006, 39, s. 621-625. Dostupný také z WWW: <<http://journals.iucr.org/j/issues/2006/04/00/wf5020/wf5020.pdf>>. doi:10.1107/S0021889806015627.

- [29] *PyCIFRW* [online]. [cit. 2011-05-01]. Dostupné z WWW: <<http://pycifrw.berlios.de/>>.
- [30] *Jmol* [online]. [cit. 2011-05-02]. Dostupné z WWW: <<http://jmol.sourceforge.net/>>.
- [31] *Jmol Wiki* [online]. 2011 [cit. 2011-05-02]. Dostupné z WWW: <<http://wiki.jmol.org/index.php>>.
- [32] *UMASS Amherst* [online]. c2011 [cit. 2011-05-02]. What is MDL Chime?. Dostupné z WWW: <[http://www.umass.edu/microbio/chime/whatis\\_c.htm](http://www.umass.edu/microbio/chime/whatis_c.htm)>.
- [33] ECMA-262. *ECMAScript Language Specification*. Geneva : Ecma International, 2009. 241 s. Dostupné z WWW: <<http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf>>.
- [34] *JQuery : write less, do more* [online]. c2010 [cit. 2011-05-02]. Dostupné z WWW: <<http://jquery.com/>>.
- [35] *W3C* [online]. c2010 [cit. 2011-05-03]. HTML & CSS. Dostupné z WWW: <<http://www.w3.org/standards/webdesign/htmlcss>>.
- [36] *W3C* [online]. c2011 [cit. 2011-05-03]. HTML5 differences from HTML4. Dostupné z WWW: <<http://www.w3.org/TR/2011/WD-html5-diff-20110405/>>.
- [37] *The Unicode Standard*. Dostupné z WWW: <<http://www.unicode.org/standard/standard.html>>.
- [38] *W3C* [online]. c2011 [cit. 2011-05-03]. Cascading Style Sheets Level 2. Dostupné z WWW: <<http://www.w3.org/TR/CSS21/>>.
- [39] *RFC Editor* [online]. 2004 [cit. 2011-05-03]. The End-of-Line Story. Dostupné z WWW: <<http://www.rfc-editor.org/EOLstory.txt>>.

- [40] *Wikipedia* [online]. [cit. 2011-05-04]. Space group. Dostupné z WWW:  
<[http://en.wikipedia.org/wiki/Space\\_group](http://en.wikipedia.org/wiki/Space_group)>.
- [41] *Webopedia* [online]. c2011 [cit. 2011-05-04]. WYSIWYG. Dostupné z WWW:  
<<http://www.webopedia.com/TERM/W/WYSIWYG.html>>.
- [42] *TinyMCE* [online]. [cit. 2011-05-04]. Dostupné z WWW:  
<<http://tinymce.moxiecode.com/>>.

## 6. Přílohy

Na přiloženém DVD naleznete kompletní zdrojový kód projektu včetně databáze a skript pro poloautomatickou extrakci informací o prostorových grupách z portálu Wikipedia.