UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Simon Okrogar

APLIKACIJA ZA PRIKAZ PERIODIČNEGA SISTEMA KEMIJSKIH ELEMENTOV

DIPLOMSKO DELO NA
VISOKOŠOLSKEMSTROKOVNEM ŠTUDIJU
PRVE STOPNJE

Mentor: viš. pred. dr. Igor Rožanc

Ljubljana, 2011

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00092/2011
Datum: 04.04.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **SIMON OKROGAR**

Naslov: **APLIKACIJA ZA PRIKAZ PERIODIČNEGA SISTEMA KEMIJSKIH ELEMENTOV**

AN APPLICATION FOR THE PRESENTATION OF THE PERIODIC SYSTEM OF CHEMICAL ELEMENTS

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Predstavite razvoj aplikacije za prikaz periodičnega sistema kemijskih elementov od zamisli do izvedbe. Razvoj izvedite v skladu z zahtevami metodologije Rational Unified Process (RUP), pri izvedbi pa uporabite javansko tehnologijo. V okviru naloge izpostavite še zlasti posebnosti in zanimive tehnične rešitve vaše aplikacije, v zaključnem delu pa izgled in uporabo vaše aplikacije.

Mentor:

viš. pred. dr. Igor Rožanc

Dekan:

prof. dr. Nikolaj Zimic

Fakulteta za računalništvo in informatiko
2
UNIVERZA V LJUBLJANI

Zahvala

Za uspešno dokončano diplomsko nalogo se moram zahvaliti mojemu mentorju dr. Igor Rožancu, ki mi je dajal napotke in me usmerjal pri njeni izdelavi.

Zahvaljujem se vsem domačim, ki so me do konca študija ves čas podpirali.

Kazalo

Kazalo tabel

Kazalo slik

Uporabljene kratice in simboli

CAS	ang. ChemicalAbstractsService - Kemični povzetki storitev
GNU	ang. GNU's Not Unix - GNU ni Unix
GPL	ang. General PublicLicense - Splošna javna licenca
GTK+	ang. GIMP Toolkit - GIMP orodje
GUI	ang. GraphicalUserInterface- Grafični uporabniški vmesnik
HTML	ang. HyperTextMarkupLanguage - Hiper-tekstovni označevalni jezik
IUPAC	ang.International Union of Pure andAppliedChemistry- Mednarodna zveza za čisto in uporabno kemijo
JDT	ang. Java DevelopmentTools - Razvojna orodja za Javo
OpenCL	ang. Open Computing Language - Odprt računalniški jezik
OSGi	ang. Open ServicesGatewayinitiativeframework- Okvir pobude storitev odprtih vrat
RUP	ang.RationalUnifiedProcess- Racionalno združevanje procesov
UML	ang. UnifiedModelingLanguage - Združljiv modelirni jezik
XML	ang. ExtensibleMarkupLanguage - Razširljiv označevalni jezik

Povzetek

V diplomskem delu bomo predstavili razvoj programske opreme, ki je namenjena vsem, ki se zanimajo za kemijo. Programska oprema temelji na kemijski periodični tabeli kemijskih elementov, uporabniku pa nudi enostaven prikaz opisa posameznih elementov ali spojin.

Diplomskega dela smo se lotili v programskem jeziku Java.Ta nudizelo dobro podporo zaizdelavo grafičnega uporabniškega vmesnika. Za razvoj programske opreme smo uporabili programsko okolje Eclipse.

V programsko opremo smo integrirali iskanje spojin glede na njihovo kemijsko sestavo ter omogočili uporabo programske opreme različnim uporabnikom, ki si delijo računalnik. Navadni uporabniki lahko tako dodajajo, spreminjajo ali brišejo kemijske spojine in opise le-teh iz svoje podatkovne baze, administratorji pa lahko poleg svoje urejajo še splošno podatkovno bazo.

Ključne besede: Periodični sistem kemijskih elementov, opis kemijskih elementov in spojin, iskanje kemijskih spojin, Java, aplikacija, XML, HTML, RUP.

Abstract

For our thesis we are going to present the development of our software that is intended for everyone who is interested in chemistry. The software is based on the periodic table of elements and offers a simple description of chemical elements and compounds to the user

We have written the software in the Java programming language. Java offers a very good support for making a graphical user interface. We developed our software in Eclipse which is a software development environment.

We have implemented a search for chemical compounds based on their chemical structure and enabled the use of our software to multiple users that share a personal computer. Normal users can add, edit or remove chemical compounds and their descriptions from their database. Administrators can do the same as normal users but they can also edit the general database.

Key word: Periodic table of elements, description of chemical elements and compounds, search for chemical compounds, Java, application, XML, HTML, RUP.

1 Uvod

V diplomskem delu želimo zagotovitienostavnejši dostop in uporabniku prijaznejšiprikaz opisov kemijskih elementov, ki so vsebovani v periodičnem sistemu, ter posameznih kemijskih spojin, ki so shranjene v podatkovni bazi.

Izhodiščni cilj diplomskega dela je narediti program za opisperiodičnega sistema kemijskih elementov z grafičnim uporabniškim vmesnikom, ki uporabniku omogoča iskanje spojin po kemijskih elementih, ki jih uporabnik izbere za iskanje. Ob izbiri ene od spojin se uporabniku prikaže opis le-te. Podobne rešitve nismo nikjer zasledili, čepravlahko uporabniku služi kot dobro orodje pri spoznavanju sestaveznanih kemijskih spojin ter pri odkrivanju novih kemijskih spojin.

Problema iskanja kemijskih spojin v podatkovni bazi smo se lotili tako, da smo kemijske elemente, ki jih je uporabnik označil, shranili v poseben seznam. Nato smo se sprehodili po vsebini podatkovne bazein kjer smo izluščili kemijsko formulo spojine, smo jo razbili na elemente in dobljene elemente shranili v začasni seznam. Tega smo primerjali s prvotnim seznamom in preverili, če so elementi iz prvega seznama vsebovani v začasnem seznamu. V kolikor so bili vsi elementi iz prvega seznama vsebovani v začasnem seznamu, smo kemijsko spojino dodali v končni seznam. Končni seznam je uporabniku prikazan in pripravljen na nadaljno uporabo.

2 Predstavitev periodičnega sistema kemijskih elementov

Prednose lotimo obravnavediplomskega dela je smiselno, če najprej predstavimo zgodovino, sestavo in uporabo periodičnega sistema kemijskih elementov za boljše razumevanje problema.

2.1 Splošno o periodičnem sistemu

Periodični sistem kemijskih elementov (lahko tudi samo periodični sistem elementov ali periodični sistem) je tabelarični prikaz kemijskih elementov. Zasluge za nastanek periodičnega sistema gredo ruskemu kemiku DmitrijuMendeleevjevu, ki je leta 1869 želel, da tabela prikazuje ponavljajoče »periodične« trende v lastnostih kemijskih elementov, čeprav obstajajo predhodniki njegovemu periodičnemu sistemu [2].

Skozi čas, ko so bili odkriti novi kemijski elementi in novi teoretični modeli za obrazložitev kemijskega vedenja, je bil razpored tabele izboljšan in razširjen. Periodični sistem je sedaj prisoten v vseh šolah,kjer se poučujekemijo.Poleg tega pa zagotavlja uporabno ogrodje za sistematiziranje in razvrščanje vseh različnih oblik kemijskega vedenja. Periodični sistem je bil pogosto uporabljen pri kemiji, fiziki, biologiji in inženiringu, še posebej pa v kemijskem inženiringu.

2.2 Zgodovina periodičnega sistema

Vse se je začelo leta 1789, ko je AntoineLavoisier objavil seznam 33 kemijskih elemenotv [1].Kljub temu, da je Lavoisierkemijske elemente razdelil v pline, kovine, nekovine in zemljineso kemiki v naslednjem stoletju iskali bolj natančno razvrstitveno shemo.

Johann Wolfgang Döbereiner je leta 1829 opazil, da bi lahko bilo veliko elementov razvrščenih v triade(skupine po tri) glede na njihove kemijske lastnosti. Na primer litij, natrij in kalij so bili razvrščeni v isto skupino, ker so mehke in reaktivne kovine. Poleg tega je Döbereiner opazil, da imajo drugi člani posamezne triade skoraj enako vrednost kot povprečje prvega in tretjega člana triade, če jih razporedi po njihovi atomski teži. To je postalo znano kot zakon triad [10].

Jean Baptiste Dumas je objavil svoje delo leta 1857 v katerem je opisal zvezo med različnimi skupinami kovin. Čeprav so mnogi kemiki lahko prepoznali zvezo med majhnimi skupinami elementov, še vedno niso zgradili sheme, ki bi obsegala vse.

Leta 1858 je nemški kemik AugustKekulé opazil, da ima ogljik težnjo po združevanju z drugimi kemijskimi elementi v razmerju ena proti štiri. Na primer metan ima en atom ogljika in štiri atome vodika. Ta koncept je sčasoma postal znan kot valenca [15].

Nemec JuliusLotharMeyer, Kekulejev kemijski kolega, je leta 1864 objavil tabelo že poznanih 49 elementov razvrščeno po njihovi valenci. Tabela je razkrila, da si elementi s podobnimi lastnostmi pogosto delijo valenco.

Angleški kemik John Newlands je napisal vrsto člankov v letih 1864 in 1865 kateri so opisali njegovo klasifikacijo kemijskih elementov. Omenil je, da če so bili elementi zapisani v naraščujočem vrstnem redu po atomski masi so se ponavljale podobne fizične in kemijske lastnosti elementov v intervalu po osem, kar je sam označil kot oktavo kakor v glasbi [13]. Temu zakonu o oktavah so se posmehovali njegovi sodobniki in združenje kemikov ni želelo objaviti njegovega dela [11]. Kljub temu je Newlands naredil osnutek atomske tabele katero je uporabil za napoved obstoja manjkajočih elementov, eden izmed njih je bil germanij.Združenje kemikov je priznalo pomembnost njegovega odkritja šele kakšnih pet let po tem, ko so pripisali zasluge Mendeleevjevu.

Ruski kemijski profesor DmitriIvanovichMendeleev in JuliusLotharMeyer sta neodvisno objavila svoja periodična sistema v letih 1869 in 1870 [1]. Oba sta svoj sistem izdelala na podoben način tako, da sta elemente razporedila v vrstice in stolpce po vrstnem redu atomskih uteži ter se premaknila v novo vrstico ali stolpec, ko so se karakteristike elementov začele ponavljati. K uspešnosti Mendeleevjeve tabele sta pripomogli dve bistveni odločitvi, ki jih je sprejel. Prva je bila ta, da je pustil prazen prostor v tabeli, ko se je zdelo, da ustrezen element še ni bil odkrit.Mendeleev sicer ni bil prvi kemik, ki je to storil, je pa bil prvi, ki je bil poznan po tem, da je uporabil te trende v svojem periodičnem sistemu za napoved lastnosti mankajočim elementom kot sta galij in germanij. Druga odločitev pa je bila, da je občasno ignoriral vrstni red glede na atomske uteži in je zamenjal sosednje elemente, kot sta na primer kobalt in nikelj, ter jih s tem bolje uvrstil v kemijske družine. Z razvojem teorij o zgradbi atoma je postalo očitno, da je Mendeleev elemente razvrstil po naraščujočem vrstnem redu atomskih števil [12].

Z razvojem teorij v moderni kvantni mehaniki o konfiguraciji elektronov znotraj atomov je postalo očitno, da je vsaka vrstica oziroma perioda v tabeli ustrezala zapolnitvi kvantne lupine elektronov. V prvotni Mendeleevjevi tabeli so bile vse periode enako velike. Moderne tabele imajo postopoma večje periode proti koncu tabele, ker imajo večji atomi več elektronskih podlupin.

Po objavi Mendeleevjevega periodičnega sistema so se v naslednjih letih prazni prostori, ki jih je Mendeleev pustil, zapolnili, ko so kemiki odkrili nove kemijske elemente [1]. Zadnji element, ki se pojavlja v naravi, je bil francij. Nanj se je Mendeleev skliceval kot eka-caesium in je bil odkrit leta 1939. Periodični sistem se je povečal z dodatkom sintetičnih in transuraničnih elementov. Prvi transuranični element, ki je bil odkrit, je bil neptunij do katerega so prišli, ko so bombardirali uran z nevtroni v Cyclotronu leta 1939 [2].

2.3 Zgradba periodičnega sistema

2.3.1 Skupine

Skupina ali družina je navpičen stolpec v periodičnem sistemu. Skupine so najpomembnejša metoda pri klasifikaciji kemijskih elementov.Vnekaterih skupinah imajo elementi zelo podobne lastnosti in jasno kažejo trende v lastnostih navzdol po skupinah. V okviru novega mednarodnega poimenovalnega sistema(IUPAC - International Union of Pure andAppliedChemistry, prevedeno v mednarodna zveza za čisto in uporabno kemijo) se skupine označujejo numerično od 1 do 18, od skrajnega levega stolpca(alkalne kovine) vse do skrajnega desnega stolpca(žlahtni plini). Starejši sistemi za poimenovanje so se malce razlikovali med Evropo in Ameriko [2].

IUPAC	CAS(Amerika)	Ime skupine	IUPAC	CAS(Amerika)	Ime skupine
skupina 1	IA	alkalne kovine	skupina 10	VIIIB	nikelj
skupina 2	IIA	zemljoalkalne kovine	skupina 11	IB	baker
skupina 3	IIIB	skandij	skupina 12	IIB	cink
skupina 4	IVB	titanij	skupina 13	IIIA	boron
skupina 5	VB	vanadij	skupina 14	IVA	ogljik
skupina 6	VIB	krom	skupina 15	VA	dušik
skupina 7	VIIB	magnezij	skupina 16	VIA	kisik
skupina 8	VIIIB	železo	skupina 17	VIIA	fluor
skupina 9	VIIIB	kobalt	skupian 18	VIIIA	helij

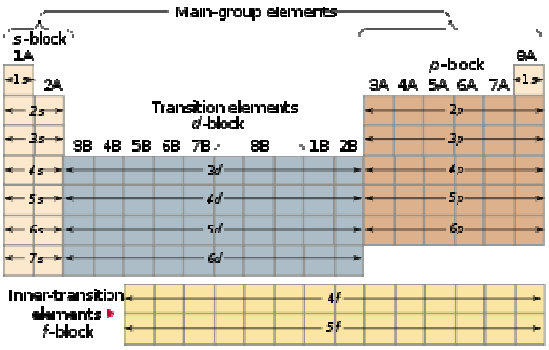
Tabela 1: Razlika označitveskupin po sistemih IUPAC in CAS

2.3.2 Periode

Perioda je vodoravna vrstica v periodičnem sistemu. Čeprav je klasifikacija elementov po skupinah najbolj pogosta, obstajajo v periodični tabeli regije, kjer so vodoravni trendi in podobnosti v lastnostih bolj pomembne kakor trendi navpičnih skupin.To je lahko res za prehodne kovine, še posebej pa za lantanide in aktinide, kjer ti tvorita dve veliki vodoravni seriji kemijskih elementov.

2.3.3 Bloki

Zaradi pomembnosti najbolj oddaljene lupine se včasih na različne regije periodičnega sistema sklicujemo kot bloki periodičnega sistema, ki so primerno poimenovani glede na podlupino, v kateri je prisoten »zadnji« elektron. Blok S obsega prvi dve skupini, alkalne kovine in zemljoalkalne kovine ter vodik in helij. Blok P obsega zadnjih šest skupin, to so skupine od 13 do 18 po IUPAC sistemu (od 3A do 8A po Ameriškem sistemu), in obsega vse polkovine. Blok D obsega skupine od 3 do 12 (od 3B do 2B po Ameriškem številčenju skupin) in vsebuje vse prehodne kovine. Blok F, ki se ponavadi nahaja na spodnjem delu periodičnega sistema, obsega lantanide in aktinide [2].



Slika 1: Razdelitev na bloke po CAS sistemu

2.3.4 Združevajne elementov

Kemijski elementi so združeni skupaj tudi na drugačne načine. Nekateri načini združevanj so prikazani na periodičnem sistemu kot na primer prehodne kovine, slabe kovine in metaloidi. Obstajajo tudi druga neformalna združevanja kakor recimo platinasta skupina in plemenite kovine.

2.4 Periodičnost kemijskih lastnosti

Glavna prednost periodičnega sistema je zmožnost napovedi kemijskih lastnosti elementa glede na njegovo lokacijo v tabeli. Omeniti pa velja, da se lastnosti razlikujejo drugače, če se premikamo navpično vzdolž stolpcev tabele kot če se premikamo vodoravno vzdolž vrstic.

2.4.1 Trendi skupin

Moderne teorije kvantne mehanike o strukturi atoma razlagajo trende skupin s predlogom, da imajo elementi znotraj iste skupine enako konfiguracijo elektronov v njihovi valenčni lupini, kar je najpomembnejši faktor pri obračunavanju njihovih podobnih lastnosti. Elementi, ki so v isti skupini, kažejo vzorce v atomskem polmeru, ionizacijski energiji in elektronegativnosti.Če gremo v skupini od vrha navzdol se atomski polmeri elementov večajo. Ker je več prostora zapolnjenega z energijo so valenčni elektroni bolj oddaljeni od jedra. Od zgoraj ima vsak nadaljni element manjšo ionizacijsko energijo, ker je lažje odstraniti elektron, saj so atomi vezani manj trdno. Podobno je tudi z elektronegativnostjo saj se bo njena vrednost v skupini zmanjšala od zgoraj navzdol, ker se povečuje razdalja med valenčnim elektronom in jedrom.

2.4.2 Trendi period

Elementi v isti periodi kažejo trende v atomskem polmeru, ionizacijski energiji, elektronski privlačnosti in elektronegativnosti. S sprehodom od leve proti desni po periodi se atomski polmer ponavadi zmanjšuje. Do tega pride, ker ima vsak nadaljni element dodaten elektron in proton kar privede do tega, da se poveča privlak med elektronom in jedrom in se elektron približa jedru. To zmanjšanje v atomskem polmeru pravtako povzroči povečanje ionizacijske energije, ko se premikamo od leve proti desni po periodi. Bolj trdno je element vezan več energije je potrebne za odcepitev elektrona.Elektronegativnost se poveča podobno kakor ionizacijska energija zaradi velikega privlaka, ki ga jedro vrši na elektrone. Pravtako tudi elektronska privlačnost kaže manjši trend vzdolž periode. Kovine(na levi strani periode) imajo običajno manjšo elektronsko privlačnost kakor nekovine(na desni strani periode) z izjemo žlahtnih plinov.

2.5 Različice periodičnega sistema

Skozi čas so raziskovalci predlagali različne načine prikazovanja periodičnega sistema. Za zgled lahko vzamemo upokojenega profesorjakemije FernandaDufourja, ki je razvil tridimenzionalni periodični sistem, kateri prikazuje temeljno simetrijo periodičnega zakona za razliko od

standardnega dvodimenzionalnega prikaza periodičnega sistema [1]. Na enakem principu je narejen prikaz periodičnega sistema v obliki piramide, katero je nazadnje izpopolnil William B. Jensen.

Poleg tega so bili narejeni periodični sistemi, ki so ciljali na povzetek lastnosti spojin in ne elementov. Leta 1980 je Ray Hefferlin oblikoval periodični sistem za vse diatomične molekule, ki lahko nastanejo med prvimi 118. elementi. Njegova tabela razkrije, da se določene lasnosti molekul (razdalja med atomi in potrebna energija za ionizacijo molekule) pojavljajo v rednih vzorcih [1]. Ta tabela je znanstvenikom omogočila uspešno napovedati diatomične molekule.

Podobno klasifikacijo je naredil Jerry R., kjer je tip organske molekule poimenoval benzenoidaromatskih ogljikovodikov [1]. Njegov klasifikacijski sistem je podoben Döbereinerjevim triadam elementov [2]. Katera koli središčna molekula triade ima končno število ogljikovih in vodikovih atomov, ki so povprečje spremljevalnih vpisov navzdol in počez po tabeli. To shemo so uporabili pri sistematični študiji lastnosti benzenoidnih aromatskih ogljikovodikov, kjer so z uporabo teorije o grafih prišli do napovedi stabilnosti in reaktivnosti nekaterih od teh spojin.

3 Povzetek uporabljenih metod in orodij

V tem poglavju bomo opisali metodo RationalUnified Process (v nadaljevanju RUP) in programska orodja, ki smo jih uporabili pri izdelavi našega diplomskega dela.

Za izdelavo naše programske opreme smo se odločili za uporabo programskega jezika Java. Za pisanje programske kode pa smo si izbrali razvojno programsko okolje Eclipse. V programskem okolju Eclipse in programskem jeziku Java smo naredili skoraj celotno diplomsko nalogo.

Pri diplomski nalogi smo uporabili še program GIMP, v katerem smo naredili primerno ikono za naš program.

3.1 Opis metodologije RationalUnifiedProcess

Metodologijo RUP je razvilo podjetje Rational in se uporablja pri razvoju programske opreme.Ta metodologija uporablja za prikaz diagramov združljiv modelirni jezik (angl. UnifiedModelingLanguage, v nadaljevanju UML) poleg tega pa ima točno določene procesno razvojne postopke. Z medsebojno delitvijo izkušenj številnih podjetij je privedlo do šestih najboljših izkušenj (angl. *bestpractices*) katere se uporabljajo pri razvoju moderne programske opreme [14].

Najboljše izkušnje so:

- iterativni razvoj programske opreme,
- vodenje zahtev,
- uporaba komponente arhitekture,
- vizualno modelirajeprogramske opreme,
- preverjanje kakovosti programske opreme in
- nadzor sprememb programske opreme.

3.1.1 Iterativni razvoj programske opreme

Pri razvoju programske opreme je potreben iterativen pristop saj ne moremo že od začetka definirati celotnega problema, oblikovati rešitev, izdelati programsko opremo in šele na koncu produkt testirati. Ob vsaki iteraciji seza programsko opremo ustvariizvršljiva datoteka katero

testirajo končni uporabniki ali pa se testiranje izvede interno. Razvijalci s tem pridobijo povratne informacije, ki pripomorejo pri nadaljnem razvoju programske opreme [3].

3.1.2 Vodenje zahtev

RUP opisuje kako moramo zbirati, urejati in dokumentirati zahtevane funkcionalnosti ter njihove omejitve. Zahteve je treba čimbolj upoštevati ter dokumentirati sprejete kompromise in odločitve. Uporabe scenarija, ki je predpisan v procesu, se je izkazala za odlično rešitev pri zajemu funkcionalnih zahtevter pri zagotavljanju, da zahteve vplivajo na načrt, implementacijo in testiranje programske opreme s čimer povečamo možnosti, da bo končan sistem zadovoljil vse potrebe uporabnika [5].

3.1.3 Uporaba komponentne arhitekture

Ta proces se osredotoča na zgodnji razvoj in osnovo robustne izvršljive arhitekture predno dodeli vse vire za razvoj. Proces opisuje kako načrtovati prilagodljivo arhitekturo sistema, ki dovoljuje spremembe, jo intuitivno razumemo in omgoča ponovno rabo komponent programske opreme. Komponente so netrivialni moduli oziroma podsistemi, ki opravljajo določeno funkcijo [3].

3.1.4 Vizualno modeliranje programske opreme

S tem procesom prikažemo vizualni model programske opreme, kako je zajeta struktura ter kako se obnašajo arhitektura in komponente. S tem grafičnim prikazom se skrije vsa programska koda in podrobnosti. RUP za grafični opis modela sistema uporablja grafični jezik UML [3].UML se uporablja za vizualizacijo, specificiranje, izdelavo in dokumentiranje elementov programskega sistema. S tem se poenostavi opis sistema in olajša komunikacijo med razvijalciin uporabniki.

3.1.5 Preverjanje kakovosti programske opreme

V moderni programski opremi sta zelo pomembna dejavnika zanesljivost in zmogljivost, zato moramo vedno preveriti kakovost sistema. Na kakovost sistema moramo gledati z vidika zanesljivosti, funkcionalnosti teraplikacijske in sistemske zmogljivosti. Ocenjevanje kakovosti je vgrajeno v sam proces, v vse aktivnosti in vključuje vse sodelujoče ter uporablja nepristranske meritve in kriterije [5].

3.1.6 Nadzor sprememb programske opreme

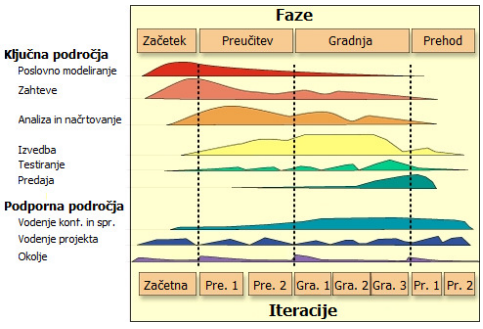
Ta proces opisuje kako je potrebno nadzorovati, slediti in spremljati spremembe, da zagotovimo uspešen iterativni razvoj. Poleg tega služi kot vodič pri vzpostavitvi varnega delovnega okolja za vsakega razvijalca posebej s tem, da mu nudi izolacijo pred narejenimi spremembami v drugih delovnih okoljih. Vse spremembe, ki so nastale v različnih delovnih okoljih, je potrebno nato tudi uskladiti, ko želimo izdati končno verzijo programske opreme [3].

3.1.7 Faze razvoja RUP

Metodologija RUP določa življenjski cikel projekta s štirimi fazami. Te faze omogočajo, da je proces predstavljen na visokem nivoju, čeprav je bistvo procesa v iteracijah razvoja, ki so prisotne znotraj faz. Vsaka izmed faz ima zastavljen nek cilj tako, da vemo, če smo cilj dosegli ali ne. Vizualizacijo faz razvoja RUP lahko predstavimo v dveh dimnezijah z uporabo grbavega (angl. *hump*) diagrama [4]. Vodoravna os prikazuje čas in vidike procesa, ki so povezani z njegovim življenskim ciklom. Navpična os pa prikazuje vsebino in področja, ki tvorijo logično vsebino procesa. V začetnih iteracijah je več pozornosti namenjene zahtevam, v nadaljevanju pa se več pozornosti nameni izvedbi.

Fazerazvoja RUP se delijo na:

- začetek,
- preučitev,
- gradnja in
- prehod.



Slika 2: Faze razvoja RUP

3.1.7.1 Začetek

Glavni cilj prve faze je ustrezno preučiti sistem, da dobimo vpogled v začetne stroške in končnega proračuna projekta. To vključuje vse od uspešnosti prodaje izdelka in prepoznavnosti na trgu do načrtovanja projekta, začetne ocene tveganja in opisa zahtev projekta.

3.1.7.2 Preučitev

V drugi fazi poskušamo ublažiti ključne probleme, ki smo jih identificirali z analiziranjem, še predno s fazo končamo. V tej fazi začenja projekt dobivati podobo. Tu se začne analiza problemov in arhitektura projekta dobi osnovno obliko.

3.1.7.3 Gradnja

V tretji fazi je cilj narediti sistem programske opreme. Še posebej se tu osredotočimo na izdelavo komponent in drugih funkcij sistema. Poleg tega pa je to faza, v kateri se največ programira. Zaradi obsega ima lahko več iteracij ter se ob zaključku vsake iteracije ustvari testna različica izdelka.

3.1.7.4 Prehod

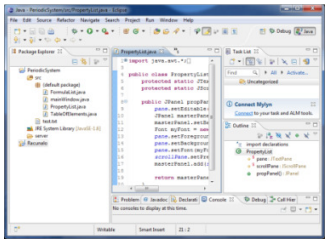
V zadnji fazi naredimo prehod med razvojem in produkcijo izdelka, kjer omogočimo uporabo in pomoč končnim uporabnikom. Aktivnosti, ki so ključne v tej fazi, so izobraževanje končnih uporabinkov in vzdrževalcev sistema ter testirajne sistema z mnenji uporabnikov, da vidimo, če smo dosegli zastavljene cilje iz začetne faze.

3.2 Razvojno okolje Eclipse

Eclipse je zastonjsko odprtokodno razvojno programsko okolje s podporo različnim programskim jezikom in vsebuje integrirano razvojno okolje (angl. IntegratedDevelopmentEnvironment, v nadaljevanju IDE) ter razširljiv sistem za vtičnike [6]. Program Eclipse je v večinskem delu napisan v programskem jeziku Java in se ga lahko uporabi za razvoj aplikacij v Javi. Zuporabo različnih vtičnikov pa lahko programiramo še v drugih programskih jezikih kot so recimo Ada, C, C++, COBOL, Perl, PHP, Python, R, Ruby(vključno z ogrodjem Ruby on Rails), Scala, Clojure in Scheme.

Prvotna koda razvojnega okolja Eclipseje temeljila na razvojnem okolju VisualAge, tako je bilo ime družini računalniško vgrajenega razvojnega okolja, ki ga je razvilo podjetje IBM in je imelo podporo programiranju v večih različnih programskih jezikih.Eclipse je v večjem delu namenjen

razvijalcem v programskem jeziku Java in vsebuje razvojna orodja za Javo (angl. Java DevelopmentTools, v nadaljevanju JDT). Uporabniki lahko njegove sposobnosti razširijo tako,da naložijo vtičnike za Eclipsino razvojno ogrodje kot so recimo razvojna orodja za druge programske jezike, lahko pa tudi napišejo in prispevajo svoje module za razne vtičnike.



Slika 3: Razvojno okolje Eclipse

3.2.1 Arhitektura

Razvojno okolje Eclipseuporablja vtičnike, da lahko ponuja popolno funkcionalnost vključno z zaganjalnim (angl. *runtime*) sistemom, za razliko od nekaterih aplikacij, kjer je funkcionalnost običajno že vprogramirana (angl. *hardcoded*). Eclipsinzaganjalni sistem temelji na Equinoxu, ki je implementiran v skladu z OSGi standardom [6].

Ta vtičniški mehanizem je majhna programska komponenta celotnega ogrodja. V Eclipsu lahko s pomočjo vtičnikov na primer delamo z urejevalniki besedila kot je recimo LaTeX, omrežnimi aplikacijami kot je telnet ter upravljanjem s podatkovnimi bazami. Arhitektura vtičnikov omogoča pisanje poljubnih dodatkov za okolje, kot na primer za nastavitve upravitelja. V Eclipsu je zagotovljena podpora Javi in CVSju preko vtičnikov pa je podprt tudi Subversion [6].

Z izjemo majhnega zagonskega jedra (angl. *kernel*) je vse v razvojnem okolju Eclipse vtičnik. To pomeni, da je vsak razvit vtičnik v Eclipsu implementiran na enak način kakor vsi ostali vtičniki tako, da so vse značilnosti narejene »enakovredno«.

Razvijalno okolje Eclipse vsebuje Eclipse JDT kateri ponuja IDE, z vgrajenim primarnim Java prevajalnikom, in celoten model Java izvorne kode. S tem so omogočeni naprednejši postopki prestrukturirajna in analiziranja kode.

3.2.2 Zgodovina

Program Eclipse seje začel kot projekt podjetja IBM Kanada. Razvilo ga je podjetje ObjectTechnologyInternational (v nadaljevanju OTI) kot javanski nadomestek za Smalltalk, ki je temeljil na družini VisualAge IDE izdelkov, katerega je pravtako razvilo podjetje OTI. Novembra 2001 je bil ustanovljen konzorcij, da bi se Eclipse še naprej razvijal kot odprtokoden program.Leta 2004 je izšla različica Eclipse 3.0 v kateri so za zaganjalno arhitekturo izbrali specifikacije OSGi storitvene platforme.

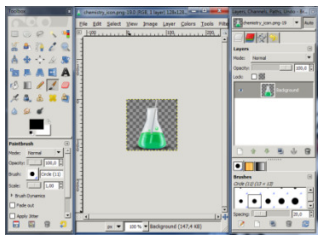
Program Eclipse je prvotno izšel pod skupno javno licenco (angl. CommonPublicLicense, okrajšava CPL), vendar so ga kasneje ponovno licencirali pod Eclipsovo javno licenco (angl. EclipsePublicLicense, okrajšava EPL) [6].

Po mnenju tedanjega tehnološkega vodje oddelka pri IBMovi racionalni diviziji, Leeja Nackmana, ime Eclipse ni bila besedna igra gleda na SunMicrosytems, saj je bil v tistem obdobju njihov glavni konkurent Microsoftov Visual Studio.

3.3 Program GIMP

GNUjev program za manipulacijo slik (angl. GNU ImageManipulation Program, v nadaljevanju GIMP) je zastonjski program za rastersko grafično oblikovanje [7]. Namenjen jepredvsem popravljanju slik in služi kot orodje za urejanje. Na voljo je brezplačno za večino popularnih operacijskih sistemov kot so Microsoft Windows, Apple Mac OS X in GNU/Linux.

Poleg tega, da lahko do podrobnosti spremenimo sliko in prosto rišemo, nam GIMP ponuja potrebna orodja za urejanje slik kot so na primer spreminjanje velikosti, spreminjanje izgleda in obrezovanje slik. Ponuja nam tudi fotomontažo večih slik in konverzijo med različnimi formati slik.



Slika 4: Program GIMP

3.3.1 Razvoj

Program GIMP v veliki večini razvijajo prostovoljci. Kljub temu pa ima projektGIMP razvijalskirazličici: nestabilno in stabilno različico. Nove značilnosti so dodane k GIMPovirazvijalskiveji. Razvijalci se lotijo izdelave nove različice programa, ko je dovolj novih značilnosti. Proceszačnejo tako, da najprej naredijo nestabilno različicoiz razvijalske različice. To različico nato stabilizirajo in zanjo delajo popravke, dokler ni pripravljenja zamenjati trenutne stabilne različice. GIMP je tako prevzel shemo, ki jo uporabljajo mnogi drugi brezplačni in odprtokodni programski (angl. Freeand open-sourcesoftware, okrajšava FOSS) projekti [7]. Drugo število v različici (na primer »6« v »2.6.11«) pove ali je izdana GIMP različica stabilna ali nestabilna. Liho število pomeni, nestabilno,sodo pa stabilno različico. Zadnje število pove, koliko popravkov je bilo po stabilni ali nestabilni različici. Od oktobra 2010 je najnovejša stabilna različica GIMPa 2.6.11.

3.3.2 Distribucija

GIMP izhaja brezplačno pod GNU GPL. Trenutna različica GIMPa deluje na številnih operacijskih sistemih vključno z GNU/Linux, Mac OS X in Microsoft Windows. Veliko distrubucij GNU/Linux vključuje GIMP kot del njihovega namiznega operacijskega sistema, kot sta na primer Debian in Fedora.

S prenosom programa GIMP na operacijski sistem Microsoft Windows je leta 1997 začel Tor Lillqvist, dokončal pa ga je Jernej Simončič. Za novejše različice GIMPa za sistemMacOS X je priporočen distributer MacPorts. GIMP je bilo veliko lažje prenesti na operacijski sistem Mac OS X kakor na Microsoft Windows, saj GTK+ (angl. GIMP Toolkit, prevedeno GIMP orodje) teče na X11 strežniku, ki ga lahko uporablja Mac OSX [7].

3.3.3 Zgodovina

Prvotni pomen kratice GIMP je bil: splošen program za manipulacijo slik (angl. General ImageManipulation Program). Ustvarjalca sta bila Spencer Kimball in Peter Mattis s Kalifornijske univerze v Berkleyju. GIMP sta začela razvijati leta 1995, že naslednje leto pa je izšla prva različica. Zatem je GIMP postal del GNU projekta.V tistem času je GIMP spremenil pomen črke »G« iz »splošno« (angl. *general*) v »GNU«. GIMP je bil narejen za operacijski

sistem UNIX, a so ga prostovoljci prevedli še na operacijska sistema Mac OS X in Microsoft Windows [7].

4 Razvoj od ideje do izdelka

V tem delu diplomske naloge bomo opisali idejo, ki smo jo razvili, prikazali ter razložili razvoj programa, ki je jedro diplomske naloge. Poleg tega se bomoosredotočili na opis postopnegareševanjater probleme, ki so nastali ob implementiranju novih funkcionalnosti.

4.1 Opis ideje

Rešitev diplomske naloge je bila preprosta, zanimiva poleg tega pa se jo lahko v prihodnosti še nadgradi z dodatno funkcionalnostjo.

Sprva smo si zamislili preprosto namizno aplikacijo, ki prikazuje periodični sistem kemijskih elementov in uporabniku ponuja dve interaktivni možnosti. Prva možnost je ta, dauporabniku ob izbiri enega izmed kemijskih elementov prikaže podroben opis tega elementa. Druga možnost pa (ob vklopljeni iskalni funkciji) uporabniku omogoča iskanje po spojinah, katere so shranjene v seznamu. Uporabniku se nato prikaže seznam spojin, ki vsebujejo tiste kemijske elemente, po katerih je uporabnik izvedel iskanje.

Ugotovili smo, da bi bilo zelo priročno, če bi imeli bazo uporabnikov. Tako uporabniku omogočimo dodajanje, brisanje ter spreminjanje kemijskih spojin iz podatkovne baze. Poleg tega tudi preprečimo nevšečnosti, če recimo en računalnik uporablja več uporabnikov in nekdo izmed njih po pomoti izbriše seznam kemijskih spojin ali pa ga morda celo namenomaspremeni.

Zahteve, katerim smo skušali ugoditi pri reševanju diplomske naloge, so funkcionalne in nefunkcionalne.

- Funkcionalne zahteve:
    - iskanje kemijskih spojin,
    - prikaz opisov kemijskih elementov in spojin,
    - uvoz in izvoz podatkov,
    - dodajanje, urejanje in brisanje podatkov,
    - tiskanje podatkov,
    - prenosljivost podatkov in
    - grafični prikaz kemijskih spojin.

- Nefunkcionalne zahteve:
    - podpora različnim uporabnikom,
    - enostavna uporaba,
    - hiter odziv,
    - zagotovljena varnost podatkov in
    - možnost prilagoditve jezika

4.2 Načrtovanje projekta

V tem poglavju bomo predstavili naš pristop k reševanju problema podatkovnih baz. To so podatkovne baze, ki vsebujejo podatke o kemijskih elementih, spojinah teruporabnikih.

4.2.1 Načrt podatkov

Predno smo ustvarili podatkovne baze, smo se morali odločiti kakšne podatke bodo vsebovale ter koliko jih bomo sploh potrebovali.Podatke smo morali najprej smiselno ločiti med seboj, kar sedaj pri uporabi aplikacije vpliva na manjši čas iskanja. Ob tehtnem premisleku smo se nato odločili za tri podatkovne baze, čeprav bi lahko imeli tudi štiri.

Podatkovne baze smo razdelili na:

- opise kemijskih elementov,
- formule, imena ter opise kemijskih spojin in
- uporabniška imena, gesla ter privilegije.

4.2.1.1 Baza kemijskih elementov

Ta podatkovna baza se uporabi samo kadar uporabink zahteva opis kemijskega elementa, zato je smiselno, da so podatki ločeni od kemijskih spojin, saj bi bil izvajalni čas funkcij, ki obdelujejo podatke, daljši. Da smo zagotovili še krajši čas izvajanja funkcij smo za vsak kemijski element naredili datoteko z njegovim pripadajočim opisom. Datoteke elementov imajo obliko (na primer »`1 - H.txt`«). Takšno obliko zapisa smo si izbrali zaradi preglednosti, saj se ime datoteke začne s številom, ki ustreza atomskemu številu kemijskega elementa, ter nadaljuje z oznako elementa.

Implementacije smo se lotili tako, da se ob izbiri kemijskega elementa pridobi niz označenega gumba in iz njega pridobi število ter oznako elementa. To dvoje nato združimo v obliko kakršno

imajo datoteke z opisi elementov, poskrbimo za pravilno lokacijo teh datotek ter nadaljujemo z obdelavo. Ob izbiri elementa se poišče ustrezno datoteko, katero se odpre, pregleda vrstico za vrstico ter vsebino sproti izpiše na grafični element `JTextPane`. Za lepši prikaz ter boljše razločanje besedila, smo v opisih kemijskih elementov [9] spremenili barvo pomembnejšim delom opisa. To smo dosegli z zaporedjem dveh znakov in sicer »`..`« in »`--`«.

Uporabniki lahko spremenijo opise kemijskih elementov, v kolikor jim trenutni ne zadoščajo, vendar morajo to storiti ročno, saj tega preko aplikacije ne dopuščamo.

4.2.1.2 Baza kemijskih spojin

To podatkovno bazo bi lahko razdelili na dve bazi. V eni bi bile formule ter imena kemijskih spojin, v drugi pa imena ter opis kemijskih spojin.

Na začetku smo si zamislili le standardno bazo kemijskih spojin, katero lahko uporablja vsak uporabnik. Ker smo nato uporabnikom ponudili enoličnost, smo morali narediti tudi sistem, ki bo omogočal uporabnikom hraniti njihove podatkovne baze kemijskih spojin. Za hrambo uporabniških podatkovnih baz smo si zamislili nek oddaljen strežnik, vendar smo pri diplomski nalogi predstavili le koncept strežnika na trenutno uporabljenem računalniku. Uporabnik bi lahko tako z ene lokacije naredil spremembe na svoji bazi, jo posodobil z izvozom na strežnik, ter nadaljeval z delom na posodobljeni bazi z druge lokacije, ko bi bazo uvozil.

Ker podatkovna baza kemijskih spojin vsebuje tri elemente (formulo, ime in opis), smo morali najti ustrezno rešitev kako te tri elemente ločiti med seboj. Kakor pri podatkovni bazi kemijskih elementov, smo se tudi tukaj poslužili zaporedja znakov. Vnosi v bazi imajo obliko (na primer »`..Co(NO_3)_2+6H_2O::Cobalt(II)Nitrate Hexahydride`«) zatem v naslednjih vrsticah sledi opis spojine na koncu pa se zaključi z zaporedjem znakov »`;;`«.

Kemijske spojine [8] iščemo s pomočjo začetnega niza »`..`«, ko ga najdemo niz razdelimo na dva dela z uporabo funkcije `split()`, kjer za delitev uporabimo niz »`::`«. S tem dobimo kemijsko formulo, katero moramo še preoblikovati, ter ime spojine. Kadar se v formuli pojavi znak »_« pomeni, da bo naslednje število zapisano z manjšo pisavo, če pa se pojavi znak »+« pomeni, da bo število normalne velikosti. Iz zgornjega primera bi se tako v grafični element `JList` zapisala urejena formula »$Co(NO_3)_2 6H_2O$« ter poleg nje še nespremenjeno ime »`Cobalt(II) Nitrate Hexahydride`«, ki ju ločuje znak »-«. Za kemijske spojine smo

naredili primer opisa. Vsem spojinam nismo dodali opisa, saj smo želeli to prepustiti uporabnikom.

V trenutku, ko se uporabnik prijavi v aplikacijo, se iz strežnika v delovni direktorij prepiševsebina njegove podatkovne baze, v kolikor ta obstaja, nad katero dela dokler se ne odjavi ali zapre aplikacijo. Takrat se vsebina podatkovne baze tudi izbriše iz delovnega direktorija. To vsebino lahko uporabnik ureja tako, da dodaja, spreminja ali briše kemijske spojine. V vseh primerih se pred dokončno potrditvijo spremembe ustvarijo začasne datoteke, ki vsebujejo vse nespremenjene podatke do mesta, kjer je do spremembe prišlo. Če spremembo potrdimo, se začasnipodatki podatkovne baze zapišejo do konca ter prepišejo vsebino prvotne podatkovne baze, v nasprotnem primeru pa se izbrišejo začasnipodatki podatkovne baze.

4.2.1.3 Baza uporabnikov

Podatke o registriranih uporabnikih smo morali nekje shraniti, zato smo se odločili za strežnik. Podatkovna baza uporabnikov je zelo preprosta in služi le za predstavitev osnovne rešitve problema večih uporabnikov.

Podatki so tu shranjeni v obliki (na primer »`adminN::adminP::1`«). Ko se uporabnik registrira se doda zapis podoben zgornjemu primeru. Če razčlenimo zgornji niz po nizu »`::`« na tri dele dobimo od leve proti desni: uporabniško ime (»`adminN`«), uporabniško geslo (»`adminP`«) in pravice administratorja (»`1`«). Navadni uporabinki nimajo pravic administratorja (»`0`«), če jih želijo pridobiti jim mora to omogočiti oseba, ki ima dostop do strežnika.

4.2.2 Načrt uporabnikov

Vsak sistem potrebuje nekoga, ki ima celoten pregled nad podatkovnimi bazami, zato smo se odločili, da uporabnike razdelimo na dve skupini: navadni uporabniki in administratorji.

Funkcije, ki se jih lahko poslužujejo navadni uporabniki so:

- registracija,
- prijava in odjava,
- uvoz in izvoz podatkovne baze,
- spreminjanje podatkov v podatkovni bazi in
- posodobitev podatkov lokalne standardne podatkovne baze.

Administratorji imajo trenutno na voljo le eno dodatno funkcijo. Ta je:

- posodobitev podatkov standardne podatkovne baze, ki je na strežniku, z izvozom.

4.2.3 Načrt funkcij

V diplomski nalogi smo morali implementirati kar nekaj funkcij, ki služijo končnim uporabnikom. Nekatere izmed teh funkcij so predstavljene v prejšnjem poglavju. Poleg teh funkcij pa smo morali poskrbeti še za:

- prikaz opisov kemijskih elementov in spojin,
- iskanje po kemijskih spojinah,
- prikaz kemijskih spojin glede na izbrano črko angleške abecede.

4.3 Postopna izdelavaprojekta

4.3.1 Velikost aplikacije

Našega projekta smo se lotili tako, da smo najprej delali na velikosti same aplikacije. Ker so razni grafični elementi, ki smo jih uporabili pri izdelavi aplikacije, zasegli dobršen del zaslona, smo se po tehtnem premisleku odločili, da bo velikosti `1024x768`točk.

4.3.2 Razčlenitev programa na razrede

Našo programsko kodo smo v končni fazi razčlenili na štiri razrede in sicer: glavni razred, razred periodičnega sistema, razred prikaza opisov ter razred prikaza formul kemijskih spojin.

V glavnem razredu smo poskrbeli za prikaz in razvrstitev ostalih treh razredov znotraj ustvarjenega okna aplikacije.

V drugem razredu periodičnega sistema smo tvorili periodični sistem ter poskrbeli za uporabnikovo interakcijo z grafičnimi elementi, ki vsebujejo okrajšave kemijskih elementov, gumbom in potrditvenim poljem.

V tretjem (najobsežnejšem) razredu, kjer smo imeli sprva le prikaz formul kemijskih spojin, smo se nato zaradi že izbranega razporejevalnika v glavnem razredu odločili izkoristiti nezaseden prostor razporejevalnika v razredu prikaza formul kemijskih spojin. Tu smo zato ustvarili nov razporejevalnik, na katerega smo postavili devet gumbov. Ti gumbi služijo interakciji z

uporabnikom. V večjem delu služijo pri upravljanju s podatki uporabnikove podatkovne baze kemijskih spojin, v manjšem delu pa služijo pri upravljanju uporabniškega računa.

Zadnji (najmanjši) razred prikaza opisov trenutno skrbi le za to, vendar se lahko tudi za ta grafični element doda kakšna interakcija, v kolikor bi bila potreba po tem.

4.3.2.1 Glavnookno aplikacije

Programiranja aplikacije smo se lotili tako, da smo najprej naredili glavni razred programa, kateri je ob zagonu aplikacije izrisal prazno okno v okolju Microsoft Windows. Prva stvar, ki nas je motila ob zagonu naše aplikacije je bila prikazana ikona. Standardno Java ikono smo zato zamenjali s kemijsko ikono, ki smo jo ustvarili v grafično oblikovalnem programu GIMP. V končni fazi, ko smo aplikacijo testirali, smo ugotovili, da se grafični elementi ne izrišejo pravilno v okolju Microsoft Windows XP za razliko od okolja Microsoft Windows 7. Ugotovili smo, da je razlika v dveh točkah po ordinatni osiy. Zaradi tega, ker Windows 7 uporablja drugačen koordinatni sistemkakor Windows XP. Za razporejevalnik smo v glavnem razredu uporabili `GridBagLayout`, ki sicer ni enostaven razporejevalnik za uporabo, a je najbolj prilagodljiv.

```
if(System.getProperty("os.name").equals("Windows XP"))
       y = 772;
Containercontent = window.getContentPane();
Imageimage = newImageIcon("./local/chemistry_icon.png").getImage();
content.setLayout(newGridBagLayout());
GridBagConstraintsgbc = newGridBagConstraints();
TableOfElementsToE = newTableOfElements();
gbc.gridx = 0;
gbc.gridy = 0;
gbc.gridwidth = 1;
content.add(ToE.table(), gbc);
FormulaListfL = newFormulaList();
gbc.gridx = 1;
gbc.gridy = 0;
gbc.gridheight = 2;
content.add(fL.listPanel(), gbc);
PropertyListpL = newPropertyList();
gbc.gridx = 0;
gbc.gridy = 1;
content.add(pL.propPanel(), gbc);
window.setTitle("Periodic Table ofElements");
window.setIconImage(image);window.pack();
window.setSize(x, y);
window.setResizable(false);
window.setLocationRelativeTo(null);
window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
window.setVisible(true);
```

Slika5: Glavni razred

Slika 5 prikazuje bistven del kode v glavnem razredu. Najprej smo poskrbeli za velikost okna, potem smo pridobili ikono za aplikacijo.Sledi razporeditev ostalih treh razredov po razporejevalnikuGridBagLayout, na koncu pa sledijo še nastavitve glavnega okna aplikacije.

```
Runtime.getRuntime().addShutdownHook(newThread(){
publicvoid run(){
	File directory = new File("./local/"+FormulaList.userNameField.getText());

	if(FormulaList.userNameField.getText().length()>0&&directory.exists()){
		String[] dir = directory.list();
		for(int i=0; i<dir.length; i++){
			File currentFile = new
			File("./local/"+FormulaList.userNameField.getText()+"/"+dir[i]);
			currentFile.delete();
		}
		directory.delete();
	}
}});
```

Slika 6: Izbris datotek in direktorija

Slika 6 prikazuje kodo, ki je bila dodana čisto na koncu glavnemu razredu v testni fazi razvojaaplikacije. Ta koda poskrbi, da se vse datoteke in direktorij trenutno prijavljenega uporabnika v naši aplikaciji izbrišejo iz delovnega direktorija v primeru, ko se je uporabnik pozabil odjaviti iz aplikacije, ko je le-to zaprl s pritiskom na gumb »X« v zgornjem desnem kotu programa. S tem smo preprečili enostavno zlorabo podatkov podatkovne baze neodjavljenegauporabinka.

4.3.2.2 Izdelava periodičnega sistema

Ko smo imeli osnovo, smo se lotili kreiranja periodičnega sistema kemijskih elementov. Začetek tega koraka je bil počasenpredvsem zaradi tega, ker smo morali preveriti obnašanje nekaterih razporejevalnikov (angl. LayoutManager) ter nato izbrati ustreznega. Ker smo imeli vse grafične elemente fiksne smo zapovršino, na katero smo odlagali razne elemente, za razporejevalnik izbrali absolutno pozicioniranje z ukazom `setLayout(null)`. To pomeni, da razporejamo grafične elemente na točno določeno mesto v razporejevalniku glede na podane podatke.Običajno sta to ordinatna in abscisna os izraženi v točkah. Tu bi bilo pomembno poudariti, da je pri računalniški grafiki pozitivna smer abscisne osi x od leve proti desni, pozitivna smer ordinatne y osi pa od zgoraj navzdol. Izhodišče koordinatnega sistema pri računalniški grafiki je tako zgornji levi kot pravokotnika.

Izdelave periodičnega sistema smo se lotili tako, da smo najprej v zanki tvorili gumbe, ki predstavljajo kemijske elemente.V zanki smo gumbom nastavili številnelastnosti med katerimi sta bili tudi barva ozadja ter oznake kemijskih spojin. Barva je bila izbrana v skladu z barvno shemo, ki jo uporabljajo periodični sistemi.Zraven smo dodali še barvno legendo s pripadajočimi imeni skupin, tako da uporabnik ve, kateri skupini pripada določen element.

```
protectedstaticvoidbuttonColor(int n){
	for(int i=0; i<orderList.length; i++){
		for(int j=0; j<orderList[i].length; j++){
			if(orderList[i][j] == n+1)
				buttons[n].setBackground(buttonColor[i]);
		}
	}
}
```

Slika 7: Nastavitev barve gumbov

Zgornja koda prikazuje funkcijo, ki skrbi za barvo gumbov. Spremenljivka `orderList`je dvodimenzionalna tabela celih števil. V prvi dimenziji hrani seznam desetih skupin v drugi dimenziji pa kemijske elemente urejene po atomskem številu, ki pripadajo določeni skupini. Nakoncuse trenutnemu gumbu določi ena izmed desetih barv, katere soshranjene v enodimenzionalni spremenljivki `buttonColor`.

Preden smo gumbe dokončno dodali na razporejevalnik periodičnega sistema, smo jim dodali še besedilo. Gumbom smo dodali atomsko število ter pripadajočo oznako kemijskega elementa. Ker smo želeli, da je atomsko število prikazano nad oznako elementa, smo se morali poslužiti HTML oblikovanja gumbov. HTML oblikovanje med številnimi možnostmi omogoča tudi večvrstični prikaz besedila v gumbih ter drugimi grafičnimi elementi.

```
buttons[i].setText("<html><div>"+(i+1)+"</div><div>"+elementSigns[i]+"</div></html>");
```

Slika 8: HTML oblikovan gumb

Slika 8 prikazuje, kako smo s HTMLjem uredili napise na gumbih.Koda zaradi preglednosti ne vsebuje uporabljenega stila, s katerim smo nastavili pisavo ter pozicijo besedila.

V tem razredu smo nato implementirali še najpomembnejši del diplomske naloge, iskanje spojin glede na njihovo kemijsko sestavo. Implementacije smo se lotili tako, da smo najprej ustvarili potrditveno polje ter iskalni gumb. S potrditvenim poljem smo določili, da lahko uporabnik išče

po kemijskih spojinah le v primeru, ko je to polje označeno. Uporabnik lahko nato izbere poljubno število kemijskih elementov in išče morebitne kemijske spojine, ki vsebujejo te elemente. V nasprotnem primeru se uporabniku ob kliku na poljuben kemijski element prikaže podroben opis izbranega elementa. Poleg tega pa tu ni mogočeizbrati več elementov hkrati, saj se prejšnji element ob izbiri novega odznači.

```
ArrayList<String>buttonNameList = newArrayList<String>();
FormulaList.searchComboBox.setSelectedIndex(0);
for(int i=0; i<buttons.length; i++){
       if(buttons[i].getBackground() == Color.ORANGE){
              intsignCounter = 0;
              StringBufferelementName = newStringBuffer();
              for(int j=0; j<buttons[i].getText().length(); j++){
                     if(buttons[i].getText().charAt(j) == '>')
                            signCounter++;
                     if(signCounter == 4){
                            while(buttons[i].getText().charAt(j+1)!='<')
                                   elementName.append(buttons[i].getText().charAt(++j));
                     buttonNameList.add(elementButtonName.toString());
                     signCounter = 0;
                     }
              }
       }
}
```

Slika 9: Seznam izbranih elementov

Slika 9 predstavlja uvod v naš problem iskanja kemijskih spojin. Tu smo poskrbeli, da smo iz označenih kemijskih elementov pridobili njihove okrajšave ter jih shranili v seznam, katerega smo uporabili v nadaljevanju.Označeni elementi imajo oranžno barvo ozadja, zato smo elemente poiskali po tej barvi. Ko smo našli gumb, ki je ustrezal tej zahtevi, smo pridobili njegov celoten HTML niz, ga pregledali in iz njega izluščili kemijsko okrajšavo elementa. Pridobljene okrajšave elementov smo nato shranili v seznam nizov.

Na podoben način smo naredili še seznam nizov iz trenutno uporabljenih podatkov podatkovne baze kemijskih spojin. Iz baze smo pregledali spojino za spojino, kjer smo vsako razstavili na njene kemijske elemente in jih shranili v seznam. V tem seznamu smo nato iskali elemente iz prvega seznama. V kolikor so bili vsi elementi iz prvega seznama vsebovani v drugem seznamu, smo nadaljevali s preoblikovanjem prvotne oblike zapisa kemijske spojine v HTML oblikovan zapis katerega smo uporabili za prijaznejši prikaz uporabniku.

```
booleanelementFound = false;
for(int k=0; k<buttonPressedNameList.size(); k++){
	if(formulaList.contains(buttonNameList.get(k)))
		elementFound = true;
	else{
		elementFound = false;
		break;
	}
}
if(elementFound){
	intnumOfChars = 0;
	intnumOfDashes = 0;
	StringBufferoutput = newStringBuffer("<html><div>");
	for(int i=0; i<formulaName[0].length(); i++){
	if(formulaName[0].charAt(i) == '_'){
	numOfChars++;
	numOfDashes++;
		output.append("<sub>"+formulaName[0].charAt(++i)+"</sub>");
		if(++i <formulaName[0].length()){
			if(formulaName[0].charAt(i)>='0'&&formulaName[0].charAt(i)<='9'){
				output.append("<sub>"+formulaName[0].charAt(i)+"</sub>");
				numOfChars++;
				i++;
			}
			i--;
		}
	}
	elseif(formulaName[0].charAt(i) == '+')
	numOfDashes++;

	else{
		outputLine.append(formulaListString[0].charAt(i));
		numOfChars++;
	}}
	while(numOfDashes-- >0){
		outputLine.append(" ");
		numOfChars++;
	}
	while(numOfChars++<22)
		outputLine.append(" ");
	tempFormulaList.put("- "+formulaName[1]+"</div></html>",output.toString());
}
```

Slika 10: Iskanje in prikaz kemijskih spojin

Slika 10 prikazuje naš pristop pri reševanju glavnega problema.Za primerjavo dveh seznamov kemijskih elementovsmo uporabili funkcijo `contains()`,s pomočjo katere smo iskali nize. Ko smo uspešno našli vse nize iz prvega seznama, smo na koncu preoblikovane nize shranili v slovar. Slovar je podatkovna struktura, ki vsebuje ključe in vrednosti. Tu smo za podatkovno strukturo uporabili `TreeMap`, kar nam je kasneje olajšalo delo, ko smo nadseznamom izvedli iskanje po ključih. Za ključe smo uporabili imena kemijskih spojin,za vrednosti pa njihove kemijske formule.

4.3.2.3 Prikaz spojin

V naslednjem razredu smo morali poskrbeti za dve stvari. Prva je bila omogočiti prikaz spojin uporabniku, druga pa je bila omogočiti upravljanje z uporabniškimi podatkovnimi bazami ter uporabniškimi računi.

Najprej smo se lotili implementacije prikaza kemijskih spojin. To smo storili tako, da smo na razporejevalnik dodali grafični element `JList`. Temu elementu smo morali nastaviti nekaj lastnosti ter dodati dve pomembni funkcionalnosti. Prva je bila, da se ob izbiri enega elementa v seznamu na polju za opis prikaže opis izbrane kemijske spojine. Druga pa je bila, da smo morali elementu `JList` dodati element `JscrollPane`, s katerim smo uporabniku omogočili sprehod čez celoten seznam spojin, ki je trenutno prikazan. Poleg prikaza seznama spojin smo se odločili, da lahko uporabnikom ponudimo še iskanje oziroma prikaz spojin po njihovih imenih. Tako lahko uporabnik prikaže spojine, ki se začnejo na določeno črko angleške abecede ali pa pač prikaže vse spojine, ki so trenutno shranjene v podatkovni bazi. V kolikor uporabnik išče kemijske spojine glede na njihovo kemijsko sestavo, se v začasen seznam za obdelavo shranijo vse spojine, ki pogojem ustrezajo, katere pa lahko uporabnik pravtako sortira po abecedi.

4.3.2.4 Upravljajne s podatki

Ker smo želeli uporabnikom omogočiti številne operacije, ki jih lahko izvaja nad podatkovno bazo, smo jim morali ponuditi enoličnost.To smo dosegli tako, da smo uporabnikom omogočili registracijo v sistem. Registracija ponuja številne možnosti uporabnikom, vendar so te namenjene izključno kreiranju in spreminjanju podatkov uporabnikove baze kemijskih spojin. Če uporabnik aplikacije nima ustvarjenega računa ali želi le kaj pogledati in meni, da podatki standardne podatkovne baze spojin niso posodobljeni, lahko to stori s pritiskom na gumb »Posodobi«. Ostale možnosti, ki so na voljo prijavljenim uporabnikom, so še dodaj, odstrani in spremeni spojino ter

uvoz ali izvoz podatkov uporabnikove podatkovne baze. Posebna možnost je izvoz saj lahko administrator izvozi podatke svoje trenutne podatkovnebaze v mapo, kjer se nahajajo podatki standardne podatkovne baze ter s tem vsebino stare bazezamenja z novo. Ob izbiri gumbov »Dodaj« za dodajanje ter »Uredi« za urejanje se prikaže novo okno, ki vsebuje vnosna polja za ime kemijske spojine, njeno formulo ter njen opis. Pri dodajanju nove spojine je trenutna omejitev le njeno ime. Če recimo spojina z enakim imenom že obstaja, le-ta ne bo dodana v podatkovno bazo. Ob pritisku na gumb »Odstrani« za odstranitev spojine iz seznama se pojavi potrditveno okno, ki uporabnika obvesti o morebitnem dokončnem izbrisu označene spojine iz podatkovne baze. Gumba »Uvozi« za izvoz in »Izvozi« za izvoz uporabniku ponujata uvoz ali izvoz podatkov njihove podatkovne baze, administratorju pa gumb »Izvoz« poleg izvoza v svojo podatkovno bazo omogoča še izvoz podatkov v standardno podatkovno bazo.

Ostali trije gumbi »Registriraj« za registracijo, »Prijavi« za prijavo ter »Odjavi«za odjavo služijo pri upravljanju z uporabniškimi računi.Ob registraciji ali prijavi se uporabniku prikažeta dve vpisni polji in sicer za uporabniško ime ter geslo. Pri registraciji je ena izmed dveh omejitev ta, da mora uporabnik vnesti najmanj pet znakov za uporabniško ime in geslo, če pogoju ni zadoščeno se prikaže obvestilo o napaki. Druga omejitev pa je, da se ob registraciji pregleda vnešeno uporabniško ime in če uporabnik s tem imenom že obstaja se izpiše napaka. Podobni pravili sta pri prijavi uporabnika, s tem da se ob neuspešni prijavi izpiše, če uporabnik obstaja ali ne, ter ali je za vnešeno uporabniško ime vnešeno napačno geslo.

4.3.2.4.1 Možnosti izboljšavehrambe podatkov

Ker je naša rešitev preprosta, kar se tiče podatkov v podatkovni bazi, je edino primerno, da omenimo izboljšave, ki bi jih lahko naredili.

V kolikor bi aplikacijo naredili dostopno javnosti bi morali vse podatke podatkovnih baz obvezno spremeniti v obliko XML.

```
<compoundList>
	<compound>
		<formula>H_2O</formula>
		<name>Water</name>
		<description>Descriptionforwater</description>
	</compound>
</compoundList>
```

Slika 11: Primer baze kemijskih spojin v XML

Slika 11 prikazuje eno od možnosti, ki bi jo lahko uporabili, pri podatkovni bazi kemijskih spojin napisani v obliki XML.

```
<elementList>
	<element>
		<name>Oxygen</name>
		<description>Descriptionforoxygen</description>
	</element>
</elementList>
```

Slika 12: Primer baze kemijskih elementov v XML

Podatkovno bazo kemijskih elementov bi lahko imeli urejeno kakor prikazuje slika 12. V tem primeru bi imeli vse elemente shranjene v eni podatkovni bazi, in ne vsak element posebej kakor sedaj.

```
<userList>
	<user>
		<name>Newuser</name>
		<password>Newpass</password>
		<privileges>0</privileges>
	</user>
</userList>
```

Slika 13: Primer baze uporabnikov v XML

Zgornja slika prikazuje izgled XML oblikovane baze uporabnikov, s podatki, ki jih v aplikaciji trenutno uporabljamo.

Od uporabnikov bi lahko ob registraciji zahtevali njihov elektronski naslov kamor bi jim poslali potrditveno registracijsko kodo, ki bi služila aktivaciji računa. V kolikor bi uporabnikom nudili plačljivo funkcionalnost aplikacije, bi od njih ob registraciji zahtevali še enoličen registracijski ključ. Poleg tega bi lahko hranili še recimo datum registracije, datum zadnje prijave ter število mesečnih in skupnih prijav, kar bi nam lahko služilo za statistiko.

4.3.2.4.2 Dodatne funkcionalnosti

Aplikacijo bi lahko izboljšali še z implementacijo dodatnih funkcionalnosti za uporabnika.Tri stvari bi lahko dodali na področju uporabniških računov. To so:

- pošiljanje gesel ali uporabniških imen,
- sprememba gesla in
- izbris računa.

Velikokrat se zgodi, da uporabnik pozabi uporabniško ime ali geslo. V takšnih primerih je zato dobrodošla možnost pridobitve ene od teh dveh stvari. Uporabniku bi tako nudili možnost pridobitve njegovega uporabniškega imena ali gesla tako, da bi mu na elektronski naslov poslali ustrezne podatke. Poleg tega je smiselno, da bi uporabniku omogočili tudi spremembo gesla, ker se lahko s tem uporabnik zaščiti, če je recimo neka tretja oseba prišla do njegovega gesla. Uporabniku bi lahko ob daljši neaktivnosti poslali obvestilo o morebitnem izbrisu računa, vključno z uporabnikovo podatkovno bazo, v kolikor se ne bi prijavil v sistem v določenem roku.

Aplikacijo bi lahko naredili s podporo več jezikom, saj angleščine ne razume vsak. To bi storili tako, da bi vse grafične elemente, ki trenutno vsebujejo angleško besedilo, razvrstili v neko XML datoteko. Prostovoljci bi nato to datoteko prevedli v druge jezike. Vsaka XML datoteka bi imela v imenu podano kratico jezika v katerem je napisana, za lažje razločanje. V aplikaciji bi nato uporabniku ponudili možnost izbire enega izmed ponujenih jezikov. Ob tej izbiri bi se besedilo vseh grafičnih elementov spremenilo iz angleškega v izbran jezik. Izbira jezika bi se naredila samo enkrat, saj bi se ob ponovnem zagonu aplikacije izbira jezika ohranila.

Poleg teh dveh sprememb pa bi lahko uporabniku nudili še možnost tiskanja. Natisnil bi lahko trenutno prikazan opis kemijskega elementa ali spojine ter trenutno prikazan seznam kemijskih spojin. Rešitve tega problema bi se lotili z implementacijo novega gumba, ki bi uporabniku ponudil dve možnosti. Uporabnik bi nato izbral, iz katerega grafičnega elementa želi tiskati: iz elementa za prikaz opisov ali elementa za prikaz seznama. Ena možnost tiskanja bi bila, da bi se vsebina izbranega elementa prepisala v začasno tekstovno datoteko, katera bi se uporabila pri tiskanju. To datoteko bi potem pregledovali vrstico za vrstico, katere bi sproti tiskali. Druga (hitrejša) možnost bi bila, da bi vsebino izbranega elementa pregledovali vrstico za vrstico ter jih sproti tiskali. Java omogoča tiskanje na tiskalnik, na enak način, kakor se zapisuje v tekstovne datoteke.

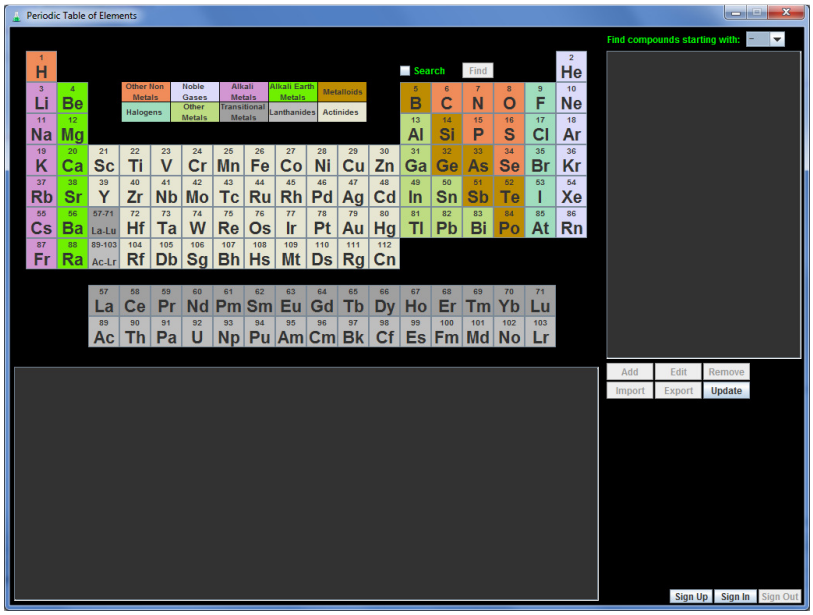
```
try {
        FileWriterout = newFileWriter("lpt1");
        out.write("Helloworld");
        out.write(0x0D); // CR
        out.close();
}
catch (IOException e) {
        e.printStackTrace();
}
```

Slika 14: Primer tiskanja v Javi v okolju Microsoft Windows

Slika 14 prikazuje enostaven način tiskanja tekstovnih besedil v programskem jeziku Java.

5 Primer uporabe programske opreme

V tem poglavju bomo prikazali delovanje našega programa ter njegovo splošno uporabo.

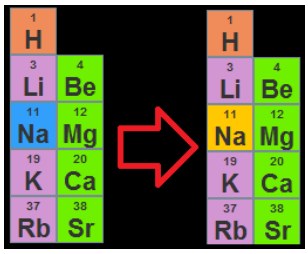


Slika 15: Glavno okno ob zagonu programa

Slika 15 prikazuje zagon našega programa v okolju Microsoft Windows 7. Na sliki lahko vidimo periodični sistem, pod njim pa je polje, kjer se izpišejo podatki o elementih ali kemijskih spojinah. Na levi strani imamo zgoraj polje, kjer se uporabniku izpišejo kemijske spojine, spodaj pa imamo že v prejšnjem poglavju omenjene gumbe.

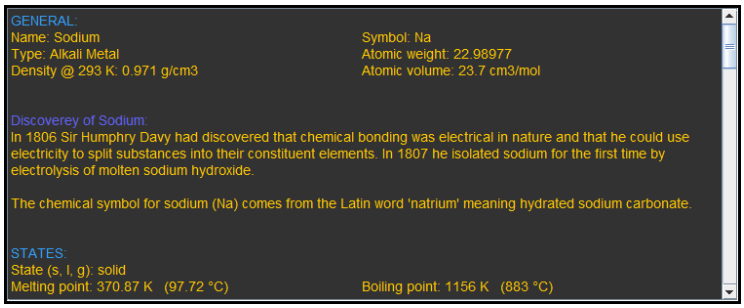
Slika 16: Barvna legenda skupin

Slika 16 prikazuje zgornji del periodičnega sistema, kjer lahko uporabnik vidi legendo skupin v katere so razvrščeni kemijski elementi.



Slika 17: Izbira elementa

Na sliki 17lahkona levi strani vidimo sprehod po periodičnem sistemu z uporabo računalniške miške. Tam, kjer se uporabnik trenutno nahaja z miškinim kazalcem, se element obarva svetlo modro.Ob izbiri elementa, s klikom na miškin gumb pa se element obarva oranžno.

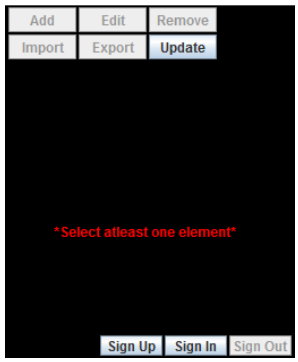


Slika 18: Opis kemijskega elementa

Ko uporabnik nima izbrane funkcije za iskanje po spojinah, se mu ob kliknu na element v spodnjem delu programa v oknu prikaže opis izbranega elementa kakor to prikazuje slika 18.

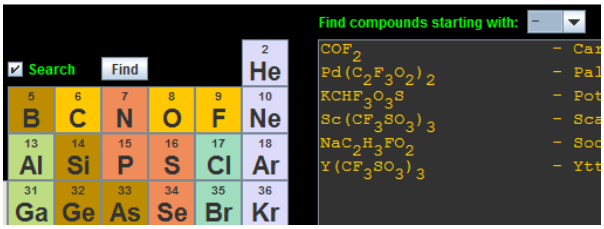
Slika 19: Izbira funkcije iskanja

Na sliki 19 lahko na levi strani vidimo, da se ob neoznačenem polju »Išči« gumba »Najdi« ne da sprožiti. To pomeni, da uporabnik ne more iskati po spojinah. Na desni strani pa vidimo, da sedaj uporabnik lahko išče po kemijskih spojinah, ki so shranjene v podatkovni bazi.



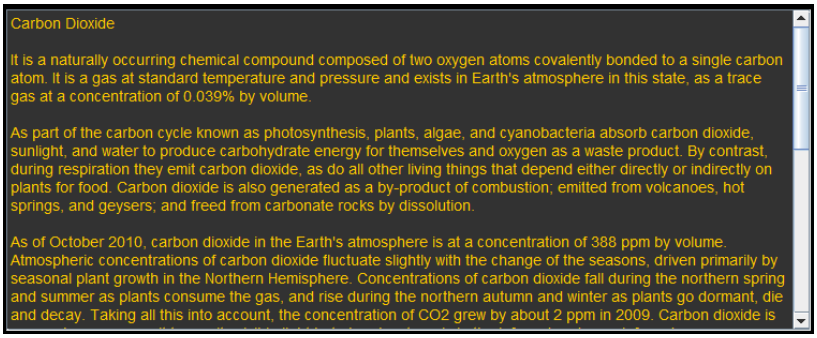
Slika 20: Napačna izbira iskanja

Slika 20 prikazuje napako pri napačni izbiri iskanja kemijskih spojin in sicer uporabnika obvesti, da ni izbral nobenega elementa za iskanje. Vse napake, do katerih lahko pride, se uporabniku izpišejo v tem delu aplikacije.



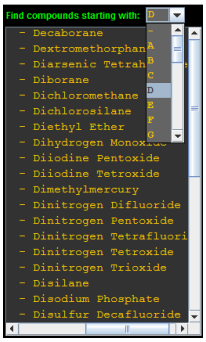
Slika 21: Izpis iskanih spojin

Slika 21 prikazuje izpis vseh kemijskih spojin, ki v tem primeru vsebujejo elemente »C«, »O« in »F«. Kemijskih spojin, ki vsebujejo izbrane elemente, je morda več, vendar so v naši podatkovni bazi shranjene le-te prikazane spojine, ki ustrezajo iskalnim pogojem.



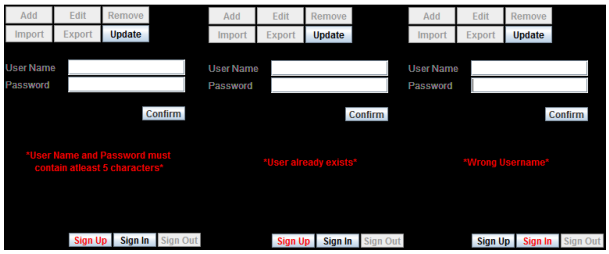
Slika 22: Opis izbrane kemijske spojine

Ob izbiri kemijske spojine (v tem primeru ogljikov dioksid CO_2)se uporabniku na mestu za izpis, kot prikazuje slika 22, pokaže opis izbrane kemijske spojine v kolikor ima kemijska spojina podan opis, v nasprotnem primeru se izpiše le ime kemijske spojine.



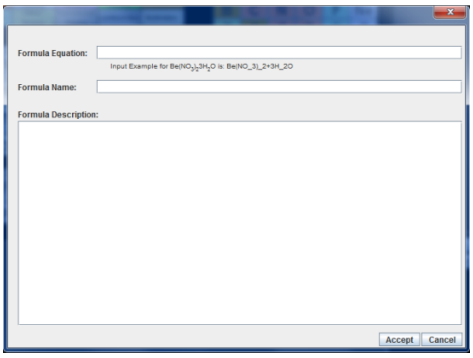
Slika 23: Sortiranje spojin

V kolikor si uporabnik želi prikazati le kemijske spojine, ki se začnejo na določeno črko angleške abecede, lahko to stori z uporabo izbirnega polja kot prikazuje slika 23. Ob izbiri znaka »-« (vezaj oziroma pomišljaj) se izpišejo vse spojine, ki so shranjene v podatkovni bazi, ki je trenutno v uporabi.



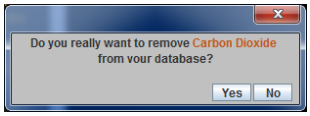
Slika 24: Napake pri registraciji ali prijavi uporabnika

Na sliki 24 lahko vidimo tri vrste napak pri registraciji ali prijavi uporabnika. Na skrajni levi je prikazana napaka, ki je prisotna pri registraciji in prijavi uporabnika ter uporabnika obvesti, da mora uporabiti daljše uporabniško ime ali geslo. Na sredini je napaka, ki se izpiše ob registraciji, kadar se želi uporabnik registrirati, a je uporabniško ime že zasedeno. Na skrajni desni pa je prikazana napaka, ko se želi uporabnik prijaviti, a je vnesel napačno uporabniško ime, saj le to ni bilo registrirano. Poleg tega pa se pri prijavi v sistem lahko izpiše še ena napaka in sicer, ko uporabnik vnese napačno geslo pri pravilno vnešenem uporabniškem imenu.



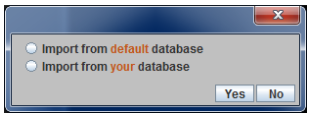
Slika 25: Dodajanje ali spreminjanje kemijske spojine

Zgornja slika prikazuje okno, ki se pokaže ob izbiri gumba »Dodaj« ali »Uredi«. Razlika med njima je ta, da je okno popolnoma prazno, kadar želi uporabnik dodati novo spojino, če pa želi uporabnik katero spojino spremeniti, se ob kreiranju zgornjega okna avtomatično vnese formula kemijske spojine, njeno ime ter njen opis, v kolikor ta obstaja.



Slika 26: Potrditev izbrisa

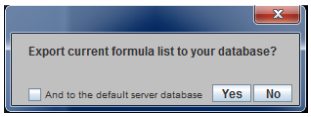
Slika 26 prikazuje potrditveno okno, ki se pokaže kadar želi uporabnik izbrisati označeno kemijsko spojino iz svoje podatkovne baze ob pritisku na gumb »Odstrani« (v tem primeru je to ogljikov dioksid).



Slika 27: Možnost uvoza

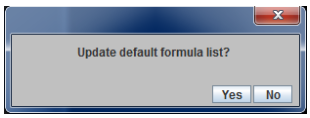
Slika 27 prikazuje dve možnosti uvoza podatkov podatkovnih baz ob pritisku na gumb »Uvozi«. Prva je uvozpodatkov iz standardne podatkovne baze v primeru, ko uporabnik svoje še nima narejene ali jo je pomotoma izbrisal ali pa je naredil kakšno neželeno spremembo. Druga možnost je, da uporabnik uvozi svojo podatkovno bazo, nad katero bo lahko delal. Obe možnosti uvoza pomenita le, da bo uporabnik ob trenutni uporabi aplikacije delal z začasnimipodatki

podatkovne baze, ki se ob odjavi ali zaprtju aplikacije izbrišejo, v kolikor jih prej ne izvozi v svojo podatkovno bazo.



Slika 28: Možnost izvoza

Slika 28 prikazuje možnost izvoza za administratorje, ko administrator uporabi gumb »Izvozi«. Spodnja možnost, omogoča administratorju posodobitev podatkov standardne podatkovne baze, poleg tega pa hkrati posodobi tudi podatke svoje podatkovne baze. Navadni uporabniki spodnjega potrditvenega okna ne vidijo, kar jim prepreči vsakršno poseganje v podatke standardne podatkovne baze kemijskih spojin.



Slika 29: Možnost posodobitve

Zgornja slika prikazuje možnost posodobitve podatkov standardne podatkovne baze iz glavne standardne baze, ki je shranjena na strežniku. To okno se pokaže, ko uporabnik izbere gumb »Posodobi«.

6 Sklep

Z izdelavo diplomske naloge smo si pridobili veliko izkušenj, še posebno tistih s stališča samega razvoja aplikacij, saj sedaj približno vemo, na kakšne težave je dobro že vnaprej pomisliti in jih odpravitipredno se lotimo dela na projektu.Podobne aplikacije za osebne računalnike nismo našli, smo pa zasledili nekaj aplikacij za iPad, vendar nobena ne ponuja iskanja po kemijskih spojinah kot smo to storili mi.

Kakor smo omenili že v prejšnjih poglavjih, je ta aplikacija trenutno najbolj primerna za domačo rabo. Za širšo rabo še ni dovolj zmogljiva in izpopolnjena.

Da bi dosegli nivo profesionalnega programa bi lahko implementirali še več idej, poleg že prej omenjenih. Najprej bi lahko implementirali kup možnosti načina prikaza kemijskih elementov.Ena izmed njih je grafični (barvni) prikaz stanja kemijskih elementov glede na izbrano temperaturo. Poleg tega bi lahko prikazali 3D modele kemijskih spojin, za kar bi uporabili OpenCL. Nenazadnjebi lahko razvili algoritem, ki bi ob izbranem kemijskem elementu pokazal primerne elemente, s katerimi se lahko veže trenutno označen element.

Registracijo uporabnikov bi morali še dodelati tako, da bi nekatere znake prepovedali ter zahtevali vsaj eno veliko črko in eno cifro.

Čeprav vseh idej nismo mogli implementirati je bilo delo na tem projektu izredno zanimivo in poučno. Kljub vsem problemom, ki so se pojavili med samim razvojem, smo jih s trudom odpravili in pridobili izkušnje, ki nam bodo služile pri nadaljnem razvoju programske opreme.

Viri

[1] (2011) Članek o razvoju periodičnega sistema. Dostopno na:
http://www.scientificamerican.com/article.cfm?id=the-evolution-of-the-periodic-system

[2](2011) Opis periodičnega sistema. Dostopno na:
http://en.wikipedia.org/wiki/Periodic_table

[3]Philippe Kruchten, *RationalUnifiedProcess-An Introduction,*Addison Wesley, 1999.

[4] (2011) Metodologija RUP. Dotopno na:
http://en.wikipedia.org/wiki/Rational_Unified_Process

[5] (2011) Podrobnejši opis metodologije RUP. Dostopno na:
http://www.augustana.ab.ca/~mohrj/courses/2000.winter/csc220/papers/rup_best_practices/rup_bestpractices.html

[6](2011) Opis programaEclipse. Dostopno na:
http://en.wikipedia.org/wiki/Eclipse_(software)

[7](2011) Opis programa GIMP. Dostopno na:
http://en.wikipedia.org/wiki/GIMP

[8] (2011) Seznam nekaterih kemijskih sopjin. Dostopno na:
http://www.convertunits.com/compounds/

[9] (2011) Opis kemijskih elementov. Dostopno na:
http://www.chemicool.com/

[10] Horvitz Leslie, *Eureka!: Scientific Breakthroughs That Changed The World*,New York: Wiley John(2002).

[11] Bryson Bill, *A Short History of Nearly Everything*, London: Black Swan (2004).

[12] Atkins P. W., *The Periodic Kingdom*, Collins Harper (1995).

[13] Newlands John A. R., *On the Law of Octaves*, Chemical News(1865).

[14]Pičulin Mitja, *Sinhronizacija večkrat nameščenih aplikacij*, Diplomsko delo, Ljubljana (2008)

[15] Weinhold Frank, Landis Clark R.,*Valency and bonding*, Cambridge: Cambridge University Press (2005)

IZVORNA KODA

Razred MainWindow

```
import java.io.*;
import java.awt.*;
import javax.swing.*;

public class mainWindow{
        private static int x=1024, y=768;
        protected static JFrame window = new JFrame();
        public static void main(String[] args){
                if(System.getProperty("os.name").equals("Windows XP"))
                        y = 772;

                Container content = window.getContentPane();
                Image image = new ImageIcon("./local/chemistry_icon.png").getImage();

                content.setLayout(new GridBagLayout());
                GridBagConstraints gbc = new GridBagConstraints();
                TableOfElements ToE = new TableOfElements();
                gbc.gridx = 0;
                gbc.gridy = 0;
                gbc.gridwidth = 1;
                content.add(ToE.table(), gbc);

                FormulaList fL = new FormulaList();
                gbc.gridx = 1;
                gbc.gridy = 0;
                gbc.gridheight = 2;
                content.add(fL.listPanel(), gbc);

                PropertyList pL = new PropertyList();
                gbc.gridx = 0;
                gbc.gridy = 1;
                content.add(pL.propPanel(), gbc);

                window.setTitle("Periodic Table of Elements");
                window.setIconImage(image);
                window.pack();
                window.setSize(x, y);
                window.setResizable(false);
                window.setLocationRelativeTo(null);
                window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        window.setVisible(true);

        //deletes the logged in user directory and it's content when program terminates
        Runtime.getRuntime().addShutdownHook(new Thread(){
            public void run(){
                File directory = new File("./local/"+FormulaList.userNameField.getText());
                if(FormulaList.userNameField.getText().length()>0 && directory.exists()){
                        String[] fileInDirectory = directory.list();
                                for(int i=0; i<fileInDirectory.length; i++){
                                    File currentFile = new
File("./local/"+FormulaList.userNameField.getText()+"/"+fileInDirectory[i]);
                                        currentFile.delete();
                                }
                                directory.delete();
                }
            }
        });
        }
}
```

Razrded TableOfElements

```
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.text.*;
import javax.swing.border.*;

public class TableOfElements implements ActionListener{

        protected static JButton[] buttons = new JButton[112];
        private JButton[] elementSign = new JButton[2];
        protected static JCheckBox searchCheckBox = new JCheckBox("Search");
        private JButton searchButton = new JButton("Find");
        private int rowNumber = 0;
        protected static JPanel warningNoElementPanel = new JPanel();
        protected static TreeMap<String, String> tempSearchFormulaList = new TreeMap<String,
String>();

        private String[] elementSigns =
{"H","He","Li","Be","B","C","N","O","F","Ne","Na","Mg","Al","Si","P","S","Cl","Ar","K","Ca","Sc",
"Ti","V","Cr","Mn","Fe",

        "Co","Ni","Cu","Zn","Ga","Ge","As","Se","Br","Kr","Rb","Sr","Y","Zr","Nb","Mo","Tc","Ru","
Rh","Pd","Ag","Cd","In","Sn",

        "Sb","Te","I","Xe","Cs","Ba","La","Ce","Pr","Nd","Pm","Sm","Eu","Gd","Tb","Dy","Ho","Er","
Tm","Yb","Lu","Hf","Ta","W",

        "Re","Os","Ir","Pt","Au","Hg","Tl","Pb","Bi","Po","At","Rn","Fr","Ra","Ac","Th","Pa","U","
Np","Pu","Am","Cm","Bk","Cf",

        "Es","Fm","Md","No","Lr","Rf","Db","Sg","Bh","Hs","Mt","Ds","Rg","Cn"};

        private static int[][] orderList = {{1,6,7,8,15,16,34},
                                                {2,10,18,36,54,86},
                                                {3,11,19,37,55,87},
                                                {4,12,20,38,56,88},
                                                {5,14,32,33,51,52,84},
                                                {9,17,35,53,85},
                                                {13,31,49,50,81,82,83},

        {57,58,59,60,61,62,63,64,65,66,67,68,69,70,71},

        {89,90,91,92,93,94,95,96,97,98,99,100,101,102,103},

        {21,22,23,24,25,26,27,28,29,30,39,40,41,42,43,44,45,46,47,48,72,73,74,75,76,77,78,79,80,10
4,105,106,107,108,109,110,111,112}
                                                };

        private static Color[] buttonColor = {new Color(240, 140, 90),
                                                   new Color(220, 220, 250),
                                                   new Color(210, 150, 210),
                                                   new Color(110, 240, 0),
                                                   new Color(190, 140, 0),
                                                   new Color(160, 220, 190),
                                                   new Color(190, 220, 130),
                                                   new Color(160, 160, 160),
                                                   new Color(190, 190, 190),
                                                   new Color(230, 230, 210)};

        //sets color for specific button
        protected static void buttonColor(int n){
                for(int i=0; i<orderList.length; i++){
                        for(int j=0; j<orderList[i].length; j++){
                                if(orderList[i][j] == n+1)
                                        buttons[n].setBackground(buttonColor[i]);
                        }
```

```
                }
        }

        public JPanel table(){
                JPanel elements = new JPanel();
                elements.setLayout(null);
                elements.setBackground(Color.BLACK);

                JPanel masterPanel = new JPanel();
                masterPanel.setPreferredSize(new Dimension(760, 430));
                masterPanel.setLayout(new BorderLayout());

                for(int i=0; i<buttons.length; i++){
                        buttons[i] = new JButton();
                        buttons[i].setMargin(new Insets(0, 0, 0, 0));
                        buttons[i].setFocusable(false);
                        buttons[i].setBorder(new MatteBorder(1, 1, 1, 1, new Color(122, 138,
153)));
                        buttons[i].addActionListener(this);
                        final int j = i;

                        //sets button color on mouse entered/exited
                        buttons[i].addMouseListener(new MouseAdapter(){
                                public void mouseEntered(MouseEvent evt){
                                        if(!(buttons[j].getBackground() == Color.ORANGE))
                                                buttons[j].setBackground(new Color(50, 160, 250));
                                }
                                public void mouseExited(MouseEvent evt){
                                        if(!(buttons[j].getBackground() == Color.ORANGE))
                                                buttonColor(j);
                                }
                        });
                        buttonColor(i);

                        //creates buttons and places them to JPanel elements
                        switch(rowNumber){
                                case 0:
                                        if(i == 0)
                                                buttons[i].setBounds(20, 30, 40, 40);
                                        else{
                                                buttons[i].setBounds(700, 30, 40, 40);
                                                rowNumber++;
                                        }
                                        break;
                                case 1:
                                        if(i < 4)
                                                buttons[i].setBounds(20+40*(i-2), 70, 40, 40);
                                        else{
                                                buttons[i].setBounds(500+40*(i-4), 70, 40, 40);
                                                if(i == 9)
                                                        rowNumber++;
                                        }
                                        break;
                                case 2:
                                        if(i < 12)
                                                buttons[i].setBounds(20+40*(i-10), 110, 40, 40);
                                        else{
                                                buttons[i].setBounds(500+40*(i-12), 110, 40, 40);
                                                if(i == 17)
                                                        rowNumber++;
                                        }
                                        break;
                                case 3:
                                        buttons[i].setBounds(20+40*(i-18), 150, 40, 40);
                                        if(i == 35)
                                                rowNumber++;
                                        break;
                                case 4:
                                        buttons[i].setBounds(20+40*(i-36), 190, 40, 40);
                                        if(i == 53)
                                                rowNumber++;
```

```
                                        break;
                                case 5:
                                        if(i < 56)
                                                buttons[i].setBounds(20+40*(i-54), 230, 40, 40);
                                        else if(i < 71){
                                                if(i == 56){
                                                        elementSign[0] = new JButton();
                                                        elementSign[0].setMargin(new Insets(0, 0, 0,
0));
                                                        elementSign[0].setBorder(new MatteBorder(1,
1, 1, 1, new Color(122, 138, 153)));
                                                        elementSign[0].setEnabled(false);
                                                        elementSign[0].setText("<html><div
style=font-size:10;font-family:Arial;margin-top:-2;margin-bottom:8;text-align:center>57-
71</div><div style=font-size:12;font-family:SansSerif;text-align:center>La-Lu</div></html>");
                                                        elementSign[0].setBackground(new Color(160,
160, 160));
                                                        elementSign[0].setBounds(20+40*(i-54), 230,
40, 40);
                                                        elements.add(elementSign[0]);
                                                }
                                                buttons[i].setBounds(100+40*(i-56), 330, 40, 40);
                                        }
                                        else{
                                                buttons[i].setBounds(140+40*(i-71), 230, 40, 40);
                                                if(i == 85)
                                                        rowNumber++;
                                        }
                                        break;
                                case 6:
                                        if(i < 88)
                                                buttons[i].setBounds(20+40*(i-86), 270, 40, 40);
                                        else if(i < 103){
                                                if(i == 88){
                                                        elementSign[1] = new JButton();
                                                        elementSign[1].setMargin(new Insets(0, 0, 0,
0));
                                                        elementSign[1].setBorder(new MatteBorder(1,
1, 1, 1, new Color(122, 138, 153)));
                                                        elementSign[1].setEnabled(false);
                                                        elementSign[1].setText("<html><div
style=font-size:10;font-family:Arial;margin-top:-2;margin-bottom:8;text-align:center>89-
103</div><div style=font-size:12;font-family:SansSerif;text-align:center>Ac-Lr</div></html>");
                                                        elementSign[1].setBackground(new
Color(190,190,190));
                                                        elementSign[1].setBounds(20+40*(i-86), 270,
40, 40);
                                                        elements.add(elementSign[1]);
                                                }
                                                buttons[i].setBounds(100+40*(i-88), 370, 40, 40);
                                        }
                                        else
                                                buttons[i].setBounds(140+40*(i-103), 270, 40, 40);
                                        break;
                                default: System.out.println("Error creating table of elements!");
                        }
                        buttons[i].setText("<html><div style=font-size:10;font-family:Arial;margin-
top:1;margin-bottom:-3;text-align:center>"+(i+1)+"</div><div style=font-size:22;font-
family:SansSerif;text-align:center>"+elementSigns[i]+"</div></html>");
                        elements.add(buttons[i]);
                }

                //Search panel with checkbox and button
                JPanel searchPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT, 20, 0));
                searchCheckBox.setForeground(Color.GREEN);
                searchCheckBox.setBackground(Color.BLACK);
                searchCheckBox.setFocusable(false);
                searchCheckBox.setMargin(new Insets(0, 0, 0, 0));
                searchCheckBox.addItemListener(new ItemListener(){
                        public void itemStateChanged(ItemEvent e){
                                for(int i=0; i<buttons.length; i++){
```

```
                                        if(buttons[i].getBackground()==Color.ORANGE)
                                                buttonColor(i);
                                }
                                if(searchCheckBox.isSelected()){
                                        searchButton.setEnabled(true);
                                        tempSearchFormulaList.clear();
                                        FormulaList.searchComboBox.setSelectedIndex(0);
                                        PropertyList.pane.setText("");
                                }
                                else{
                                        warningNoElementPanel.setVisible(false);
                                        searchButton.setEnabled(false);
                                        tempSearchFormulaList.clear();
                                }
                                FormulaList.model.clear();
                                FormulaList.selectedItemIndex = -1;
                        }
                });
                searchPanel.add(searchCheckBox);

                warningNoElementPanel.setLayout(null);
                warningNoElementPanel.setBounds(5, 48, 250, 240);
                warningNoElementPanel.setBackground(Color.BLACK);
                warningNoElementPanel.setVisible(false);
                final JLabel warningLabel = new JLabel();
                warningLabel.setForeground(Color.RED);
                warningLabel.setBounds(20, 120, 200, 60);
                warningNoElementPanel.add(warningLabel);
                FormulaList.buttonPanel.add(warningNoElementPanel);

                searchButton.setMargin(new Insets(0, 0, 0, 0));
                searchButton.setPreferredSize(new Dimension(40, 20));
                searchButton.setFocusable(false);
                searchButton.setEnabled(false);
                searchButton.addMouseListener(new MouseAdapter(){
                        public void mouseClicked(MouseEvent evt){
                                if(searchCheckBox.isSelected()){
                                        if(FormulaList.formulaButtons[6].getForeground() ==
Color.RED || FormulaList.formulaButtons[7].getForeground() == Color.RED){
                                                FormulaList.regSignPanel.setVisible(false);

        FormulaList.formulaButtons[6].setForeground(Color.BLACK);

        FormulaList.formulaButtons[7].setForeground(Color.BLACK);
                                        }
                                        ArrayList<String> buttonPressedNameList = new
ArrayList<String>();
                                        FormulaList.searchComboBox.setSelectedIndex(0);
                                        for(int i=0; i<buttons.length; i++){
                                                if(buttons[i].getBackground() == Color.ORANGE){
                                                        int signCounter = 0;
                                                        StringBuffer elementButtonName = new
StringBuffer();
                                                        for(int j=0; j<buttons[i].getText().length();
j++){
                                                                if(buttons[i].getText().charAt(j) ==
'>')
                                                                        signCounter++;
                                                                if(signCounter == 4){

        while(buttons[i].getText().charAt(j+1) != '<')

        elementButtonName.append(buttons[i].getText().charAt(++j));

        buttonPressedNameList.add(elementButtonName.toString());
                                                                        signCounter = 0;
                                                                }
                                                        }
                                                }
                                        }
                                        if(buttonPressedNameList.isEmpty()){
```

```
                                        FormulaList.model.clear();
                                        warningLabel.setText("<html><div style=padding-
left:20>*Select atleast one element*</div></html>");
                                        warningNoElementPanel.setVisible(true);
                                }
                                else{
                                        FormulaList.model.clear();
                                        warningNoElementPanel.setVisible(false);
                                        tempSearchFormulaList.clear();
                                    try{
                                        Scanner sc;
                                        if(FormulaList.signedIn){
                                                try{
                                                        sc = new Scanner(new
File("./local/"+FormulaList.userNameField.getText()+"/compoundsList.txt"));
                                                }
                                                catch(FileNotFoundException exc){
                                                        sc = new Scanner(new
File("./local/default/compoundsList.txt"));
                                                }
                                        }
                                        else{
                                                sc = new Scanner(new
File("./local/default/compoundsList.txt"));
                                        }
                                                while(sc.hasNextLine()){
                                                        String tempLine = sc.nextLine();
                                                        if(tempLine.length() > 2){
                                                                if(tempLine.substring(0,
2).compareTo("..") == 0){
                                                                        ArrayList<String>
formulaElementsList = new ArrayList<String>();
                                                                        StringBuffer
elementFormulaName = new StringBuffer();
                                                                        tempLine =
tempLine.substring(2, tempLine.length());
                                                                        String[]
formulaListString = tempLine.split("::");
                                                                        for(int i=0;
i<formulaListString[0].length(); i++){
        if(formulaListString[0].charAt(i) == '_' ||
        (formulaListString[0].charAt(i) > 47 && formulaListString[0].charAt(i) < 58)){
        if(elementFormulaName.length() > 0){
        formulaElementsList.add(elementFormulaName.toString());
        elementFormulaName.delete(0, elementFormulaName.length());
                                                                                }
                                                                        }
                                                                        else{
        if(formulaListString[0].charAt(i) > 64 && formulaListString[0].charAt(i) < 91){
        if(elementFormulaName.length() > 0){
        formulaElementsList.add(elementFormulaName.toString());
        elementFormulaName.delete(0, elementFormulaName.length());
        elementFormulaName.append(formulaListString[0].charAt(i));
        }
        else{
        elementFormulaName.append(formulaListString[0].charAt(i));
        }
```

```
                                                                                }
                                                                                else

        elementFormulaName.append(formulaListString[0].charAt(i));
                                                                                if(i ==
formulaListString[0].length()-1)

        formulaElementsList.add(elementFormulaName.toString());
                                                                        }
                                                                }
                                                                boolean elementFound =
false;
                                                                for(int k=0;
k<buttonPressedNameList.size(); k++){

        if(formulaElementsList.contains(buttonPressedNameList.get(k))){

        elementFound = true;
                                                                        }
                                                                        else{

        elementFound = false;
                                                                                break;
                                                                        }
                                                                }
                                                                if(elementFound){
                                                                        int numOfChars
= 0;
                                                                        int numOfDashes
= 0;
                                                                        StringBuffer
outputLine = new StringBuffer("<html><div>");
                                                                        for(int i=0;
i<formulaListString[0].length(); i++){

        if(formulaListString[0].charAt(i) == '_'){

        numOfChars++;

        numOfDashes++;

        outputLine.append("<sub style=font-size:17>"+formulaListString[0].charAt(++i)+"</sub>");

        if(++i < formulaListString[0].length()){

        if(formulaListString[0].charAt(i) >= '0' && formulaListString[0].charAt(i) <= '9'){

                outputLine.append("<sub style=font-
size:17>"+formulaListString[0].charAt(i)+"</sub>");

                numOfChars++;

                i++;

        }

        i--;

        }
                                                                                }
                                                                                else
if(formulaListString[0].charAt(i) == '+'){

        numOfDashes++;
                                                                                }
                                                                                else{

        outputLine.append(formulaListString[0].charAt(i));

        numOfChars++;
                                                                                }
```

```
                                                                                }

        while(numOfDashes-- >0){

        outputLine.append(" ");

        numOfChars++;
                                                                                }

        while(numOfChars++<22)

        outputLine.append(" ");

        tempSearchFormulaList.put("- "+formulaListString[1]+"</div></html>",
outputLine.toString());
                                                                        }
                                                                }
                                                        }
                                                }
                                                sc.close();
                                        }
                                        catch(FileNotFoundException exc){
                                                exc.printStackTrace();
                                            }

                                        if(!tempSearchFormulaList.isEmpty()){
                                            Object[] key =
TableOfElements.tempSearchFormulaList.keySet().toArray();
                                                //Arrays.sort(key);
                                                for (int i=0; i<key.length; i++){

        FormulaList.model.add(FormulaList.list.getModel().getSize(),
TableOfElements.tempSearchFormulaList.get(key[i])+key[i]);
                                                }
                                        }
                                        else if(tempSearchFormulaList.isEmpty()){
                                            warningLabel.setText("<html><div style=padding-
left:20;text-align:center>*Compounds with selected elemetns not found*</div></html>");
                                                warningNoElementPanel.setVisible(true);
                                        }
                                    }
                                }
                        }
                });
                searchPanel.add(searchButton);
                searchPanel.setBounds(490, 45, 150, 20);
                searchPanel.setBackground(Color.BLACK);
                elements.add(searchPanel);

                //Element legend
                String[] names = {"Other Non Metals","Noble Gases","Alkali Metals","Alkali Earth
Metals","  Metalloids",
                                                "  Halogens","Other
Metals","Transitional Metals","Lanthanides","  Actinides"};
                JLabel[] nameLabels = new JLabel[10];
                for(int i=0; i<names.length; i++){
                        nameLabels[i] = new JLabel("<html><div style=font-size:10;text-
align:center>"+names[i]+"</div></html>");
                        nameLabels[i].setOpaque(true);
                        nameLabels[i].setBackground(buttonColor[i]);
                        if(i<5)
                                nameLabels[i].setBounds(143+63*i, 70, 62, 25);
                        else
                                nameLabels[i].setBounds(143+63*(i-5), 96, 62, 25);
                        elements.add(nameLabels[i]);
                }

                masterPanel.add(elements, BorderLayout.CENTER);
                return masterPanel;
        }
```

```
        //when an element button is clicked in non-search mode it's content is written to
jtextpane
        public void actionPerformed(ActionEvent d){
                Object source = d.getSource();
                for(int i=0; i<buttons.length; i++){
                        if(source == buttons[i]){
                                if(buttons[i].getBackground() == Color.ORANGE){
                                        buttonColor(i);
                                        PropertyList.pane.setText("");
                                }
                                else{
                                        if(!searchCheckBox.isSelected()){
                                                for(int j=0; j<buttons.length; j++)
                                                        if(buttons[j].getBackground() ==
Color.ORANGE)
                                                                buttonColor(j);
                                                StyledDocument doc =
PropertyList.pane.getStyledDocument();
                                                Style styleOrange = doc.addStyle("Orange", null);
                                                Style styleBlue = doc.addStyle("Blue", null);
                                                Style styleSub = doc.addStyle("Italic", null);
                                                StyleConstants.setForeground(styleOrange,
Color.ORANGE);
                                                StyleConstants.setForeground(styleBlue, new
Color(50, 160, 250));
                                                StyleConstants.setForeground(styleSub, new
Color(100, 100, 250));
                                                Style tempStyle;

                                                try{
                                                        doc.remove(0, doc.getLength());
                                                        File inFile = new
File("./local/elements/"+Integer.toString(i+1)+" - "+elementSigns[i]+".txt");
                                                        Scanner sc = new Scanner(inFile);
                                                        String line = null;
                                                        while(sc.hasNextLine()){
                                                                line = sc.nextLine();
                                                        if(line.length()>0 && line.substring(0,
2).equals("..")){
                                                                line=line.substring(2,
line.length());
                                                                tempStyle = styleBlue;
                                                        }
                                                        else if(line.length()>0 && line.substring(0,
2).equals("--")){
                                                                line=line.substring(2,
line.length());
                                                                tempStyle = styleSub;
                                                        }
                                                        else
                                                                tempStyle = styleOrange;
                                                        doc.insertString(doc.getLength(), line+"\n",
tempStyle);
                                                        }
                                                        PropertyList.pane.setCaretPosition(0);
                                        }
                                        catch(BadLocationException exc){
                                            exc.printStackTrace();
                                        }
                                        catch(IOException exc){
                                                exc.printStackTrace();
                                        }
                                        }
                                        buttons[i].setBackground(Color.ORANGE);
                                }
                        }
                }
        }
}
```

Razred FormulaList

```
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import java.nio.channels.*;
import java.util.*;
import javax.swing.*;
import javax.swing.text.*;
import javax.swing.event.*;

public class FormulaList{
	protected static DefaultListModel model = new DefaultListModel();
	protected static JList list = new JList(model);
	private JScrollPane listScroller = new JScrollPane(list);
	private String[] alphabet = {"-
","A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S","T","U","V","W","X"
,"Y","Z"};
	protected static boolean signedIn = false;
	protected static final JTextField userNameField = new JTextField();
	private JPasswordField userPassField = new JPasswordField();
	protected static JPanel regSignPanel = new JPanel();
	protected static JButton[] formulaButtons = new JButton[9];
	private JLabel warning = new JLabel();
	protected static int selectedItemIndex = 0;
	protected static int adjusting = 0;
	protected final static JPanel buttonPanel = new JPanel();
	private StringBuffer name;
	protected int privileges = 0;
	private JLabel error = new JLabel();
	private PrintWriter pw = null;
	protected static JComboBox searchComboBox;

	public JPanel listPanel(){
		JPanel masterPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 0, 0));
		masterPanel.setPreferredSize(new Dimension(258, 740));
		masterPanel.setBackground(Color.BLACK);

		//search panel
		JPanel searchPanel = new JPanel();
		searchPanel.setPreferredSize(new Dimension(240,25));
		searchPanel.setBackground(Color.BLACK);
		JLabel searchLabel = new JLabel("Find compounds starting with: ");
		searchLabel.setForeground(Color.GREEN);
		searchPanel.add(searchLabel);
		searchComboBox = new JComboBox(alphabet);
		searchComboBox.addActionListener(new ActionListener(){
			public void actionPerformed(ActionEvent evt){
				JComboBox jcb = (JComboBox)evt.getSource();
			String firstChar = (String)jcb.getSelectedItem();
			model.clear();
			PropertyList.pane.setText("");

			if(!TableOfElements.searchCheckBox.isSelected())
				for(int i=0; i<TableOfElements.buttons.length; i++)
					if(TableOfElements.buttons[i].getBackground() ==
Color.ORANGE)
						TableOfElements.buttonColor(i);
			listScroller.getViewport().setViewPosition(new Point(0,0));
			try{
				if(!TableOfElements.tempSearchFormulaList.isEmpty()){
					Object[] key =
TableOfElements.tempSearchFormulaList.keySet().toArray();
					Arrays.sort(key);
					for (int i = 0; i < key.length; i++) {
						if(firstChar.charAt(0) ==
key[i].toString().substring(2).charAt(0))
							model.add(list.getModel().getSize(),
TableOfElements.tempSearchFormulaList.get(key[i])+key[i]);
						else if(firstChar.charAt(0) == '-')
```

```
                                                        model.add(list.getModel().getSize(),
TableOfElements.tempSearchFormulaList.get(key[i])+key[i]);
                                                }
                                }
                                else{
                                        Scanner sc;
                                        if(signedIn){
                                                try{
                                                        sc = new Scanner(new
File("./local/"+userNameField.getText()+"/compoundsList.txt"));
                                                }
                                                catch(FileNotFoundException exc){
                                                        sc = new Scanner(new
File("./local/default/compoundsList.txt"));
                                                }
                                        }
                                        else
                                                sc = new Scanner(new
File("./local/default/compoundsList.txt"));

                                                while(sc.hasNextLine()){
                                                        String tempLine = sc.nextLine();
                                                        if(tempLine.length()>2){
                                                                if(tempLine.substring(0,
2).compareTo("..") == 0){
                                                                        int numOfChars = 0;
                                                                        int numOfSigns = 0;
                                                                        tempLine =
tempLine.substring(2, tempLine.length());
                                                                        String[] formulaListString =
tempLine.split("::");
                                                                        StringBuffer outputLine = new
StringBuffer("<html><div>");
                                                                        for(int i=0;
i<formulaListString[0].length(); i++){

        if(formulaListString[0].charAt(i) == '_'){
                                                                                        numOfChars++;
                                                                                        numOfSigns++;

        outputLine.append("<sub style=font-size:17>"+formulaListString[0].charAt(++i)+"</sub>");
                                                                                        if(++i <
formulaListString[0].length()){

        if(formulaListString[0].charAt(i) >= '0' && formulaListString[0].charAt(i) <= '9'){

        outputLine.append("<sub style=font-size:17>"+formulaListString[0].charAt(i)+"</sub>");

        numOfChars++;

        i++;
                                                                                                }
                                                                                                i--;
                                                                                        }
                                                                                }
                                                                                else
if(formulaListString[0].charAt(i) == '+'){
                                                                                        numOfSigns++;
                                                                                }
                                                                                else{

        outputLine.append(formulaListString[0].charAt(i));
                                                                                        numOfChars++;
                                                                                }
                                                                        }
                                                                        while(numOfSigns-- > 0){

        outputLine.append(" ");
                                                                                numOfChars++;
                                                                        }
                                                                        while(numOfChars++ < 22)
```

```
        outputLine.append(" ");

        TableOfElements.tempSearchFormulaList.put("- "+formulaListString[1]+"</div></html>",
outputLine.toString());
                                                        }
                                                }
                                        }
                                        sc.close();

                                        Object[] key =
TableOfElements.tempSearchFormulaList.keySet().toArray();
                                        Arrays.sort(key);
                                        for (int i = 0; i < key.length; i++){
                                                if(firstChar.charAt(0) ==
key[i].toString().substring(2).charAt(0))
                                                        model.add(list.getModel().getSize(),
TableOfElements.tempSearchFormulaList.get(key[i])+key[i]);
                                                else if(firstChar.charAt(0) == '-')
                                                        model.add(list.getModel().getSize(),
TableOfElements.tempSearchFormulaList.get(key[i])+key[i]);
                                        }
                                }
                                }
                                catch(FileNotFoundException exc){
                                        exc.printStackTrace();
                                }
                        }
                });
                searchComboBox.setPreferredSize(new Dimension(50, 20));
                Font comboFont = new Font("Courier New", Font.BOLD, 14);
                searchComboBox.setFont(comboFont);
                searchComboBox.putClientProperty("JComboBox.isTableCellEditor",
Boolean.TRUE);//prevents triggering jcombobox with keyboard unless you press enter
                Component checkBoxButton = searchComboBox.getComponent(0);
                Color defaultColor=checkBoxButton.getBackground();
                searchComboBox.setBackground(new Color(100, 100, 100));
                checkBoxButton.setBackground(defaultColor);
                searchComboBox.setForeground(Color.ORANGE);
                searchPanel.add(searchComboBox);
                masterPanel.add(searchPanel);

                //formula list panel
                list.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
                list.setLayoutOrientation(JList.VERTICAL);
                list.setBackground(new Color(50, 50, 50));
                Font listFont = new Font("Courier New", Font.PLAIN, 15);

                list.setFont(listFont);
                list.setForeground(Color.ORANGE);
                list.setFixedCellHeight(20);
                list.addListSelectionListener(new ListSelectionListener(){
                        public void valueChanged(ListSelectionEvent evt){
                                if(!evt.getValueIsAdjusting())
                                        adjusting = evt.getLastIndex();
                                else{
                                        if((selectedItemIndex = evt.getFirstIndex()) > adjusting)
                                                selectedItemIndex = adjusting;
                                        else
                                                selectedItemIndex = evt.getFirstIndex();
                                        String temp = list.getSelectedValue().toString();
                                        name = new StringBuffer();
                                        for(int i=0; i<temp.length(); i++){
                                                if(temp.charAt(i) == '-' && temp.charAt(++i) == '
'){
                                                        for(int j=i+1; j<temp.length(); j++){
                                                                if(temp.charAt(j) == '<')
                                                                        break;
                                                                name.append(temp.charAt(j));
                                                        }
                                                }
```

```
                                        }
                                        File inFile;
                                        if(signedIn)
                                                inFile = new
File("./local/"+userNameField.getText()+"/compoundsList.txt");
                                        else
                                                inFile = new
File("./local/default/compoundsList.txt");
                                        BufferedReader br = null;
                                        try{
                                                if(!TableOfElements.searchCheckBox.isSelected())
                                                        for(int i=0;
i<TableOfElements.buttons.length; i++)
                                                        if(TableOfElements.buttons[i].getBackground()
== Color.ORANGE)
                                                                TableOfElements.buttonColor(i);

                                                br = new BufferedReader(new FileReader(inFile));
                                                String line = null;
                                                while((line = br.readLine()) != null){
                                                        if(line.length()>2 && line.substring(0,
2).equals("..")){
                                                                String[] splitLine =
line.split("::");

        if(splitLine[1].equalsIgnoreCase(name.toString())){
                                                                        Document doc =
PropertyList.pane.getDocument();
                                                                        try{
                                                                                doc.remove(0,
doc.getLength());

        doc.insertString(doc.getLength(), splitLine[1]+"\n\n", null);
                                                                        while((line = br.readLine())
!= null){
                                                                                try{

        if(line.equals(";;")){
                                                                                                break;
                                                                                        }

        doc.insertString(doc.getLength(), line+"\n", null);
                                                                                }

        catch(BadLocationException exc){

exc.printStackTrace();
                                                                                }
                                                                        }
                                                                }
                                                                catch(BadLocationException exc){
                                                                    exc.printStackTrace();
                                                                }
                                                                }
                                                        }
                                                }
                                                br.close();
                                        }
                                        catch(IOException exc){
                                                exc.printStackTrace();
                                        }
                                }
                        }
                });
                list.addKeyListener(new KeyAdapter(){
                        public void keyReleased(KeyEvent ke){
                            if(ke.getKeyCode() == KeyEvent.VK_ENTER){
                                if(list.getLeadSelectionIndex() != selectedItemIndex){
                                        String temp = list.getSelectedValue().toString();
                                        name = new StringBuffer();
                                                for(int i=0; i<temp.length(); i++){
```

```
                                                        if(temp.charAt(i) == '-' && temp.charAt(++i)
== ' '){
                                                                for(int j=i+1; j<temp.length(); j++){
                                                                        if(temp.charAt(j) == '<')
                                                                                break;
                                                                        name.append(temp.charAt(j));
                                                                }
                                                        }
                                                }
                                                File inFile;
                                                if(signedIn)
                                                        inFile = new
File("./local/"+userNameField.getText()+"/compoundsList.txt");
                                                else
                                                        inFile = new
File("./local/default/compoundsList.txt");
                                                BufferedReader br = null;
                                                try{

        if(!TableOfElements.searchCheckBox.isSelected())
                                                                for(int i=0;
i<TableOfElements.buttons.length; i++)

        if(TableOfElements.buttons[i].getBackground() == Color.ORANGE)

        TableOfElements.buttonColor(i);

                                                        br = new BufferedReader(new
FileReader(inFile));
                                                        String line = null;
                                                        while((line = br.readLine()) != null){
                                                                if(line.length()>2 &&
line.substring(0, 2).equals("..")){
                                                                        String[] splitLine =
line.split("::");

        if(splitLine[1].equalsIgnoreCase(name.toString())){
                                                                                Document doc =
PropertyList.pane.getDocument();
                                                                                try{
                                                                                        doc.remove(0,
doc.getLength());

        doc.insertString(doc.getLength(), splitLine[1]+"\n", null);
                                                                                while((line =
br.readLine()) != null){
                                                                                        try{

        if(line.equals(";;")){

        break;
                                                                                                }


        doc.insertString(doc.getLength(), line+"\n", null);
                                                                                        }


        catch(BadLocationException exc){

exc.printStackTrace();
                                                                                        }
                                                                                }
                                                                        }
                                                                        catch(BadLocationException
exc){
                                                                            exc.printStackTrace();
                                                                        }
                                                                        }
                                                                }
                                                        }
                                                        br.close();
                                                }
```

```
                                        catch(IOException exc){
                                                exc.printStackTrace();
                                        }

                                ke.consume();
                            }
                            selectedItemIndex = list.getLeadSelectionIndex();
                        }
                        else
                            selectedItemIndex = -1;
                    }
                });

                JPanel scrollPanel = new JPanel();
                scrollPanel.setBackground(Color.BLACK);
                listScroller.setPreferredSize(new Dimension(250, 395));
                scrollPanel.add(listScroller);
                masterPanel.add(scrollPanel);

                //button panel
                buttonPanel.setLayout(null);
                buttonPanel.setPreferredSize(new Dimension(258, 310));
                buttonPanel.setBackground(Color.BLACK);
                String[] formulaButtonNames = {"Add", "Edit", "Remove", "Import", "Export",
"Update", "Sign Up", "Sign In", "Sign Out"};
                for(int i=0; i<formulaButtons.length; i++){
                        formulaButtons[i] = new JButton(formulaButtonNames[i]);
                        formulaButtons[i].setMargin(new Insets(0, 0, 0, 0));
                        formulaButtons[i].setFocusable(false);
                        if(i!=5 && i!=6 && i!=7)
                                formulaButtons[i].setEnabled(signedIn);
                        if(i < 3)
                                formulaButtons[i].setBounds(5+62*i, 0, 60, 22);
                        else if(i < 6)
                                formulaButtons[i].setBounds(5+62*(i-3), 24, 60, 22);
                        else
                                formulaButtons[i].setBounds(86+57*(i-6), 290, 55, 18);
                        //register panel
                        regSignPanel.setLayout(null);
                        regSignPanel.setBounds(5, 48, 250, 240);
                        regSignPanel.setBackground(Color.BLACK);

                        JLabel userNameLabel = new JLabel("User Name");
                        userNameLabel.setForeground(Color.GRAY);
                        userNameLabel.setFont(new Font("Arial", Font.PLAIN, 12));
                        userNameLabel.setBounds(0, 20, 80, 20);
                        regSignPanel.add(userNameLabel);

                        userNameField.setBounds(80, 20, 150, 20);
                        regSignPanel.add(userNameField);

                        JLabel userPassLabel = new JLabel("Password");
                        userPassLabel.setForeground(Color.GRAY);
                        userPassLabel.setFont(new Font("Arial", Font.PLAIN, 12));
                        userPassLabel.setBounds(0, 42, 80, 20);
                        regSignPanel.add(userPassLabel);

                        userPassField.setBounds(80, 42, 150, 20);
                        regSignPanel.add(userPassField);

                        warning.setForeground(Color.RED);
                        warning.setBounds(20, 120, 200, 60);
                        regSignPanel.add(warning);

                        JButton confirm = new JButton("Confirm");
                        confirm.setMargin(new Insets(0, 0, 0, 0));
                        confirm.setBounds(175, 80, 55, 18);
                        confirm.setFocusable(false);
                        confirm.addMouseListener(new MouseAdapter(){
                                public void mouseClicked(MouseEvent evt){
                                        boolean userExists = false;
```

```
                                        char[] passArray = userPassField.getPassword();
                                        StringBuffer pass = new StringBuffer();
                                        warning.setText("<html><div style=padding-left:50>*User
already exists*</div></html>");
                                        warning.setVisible(false);

                                        for(int i=0; i<passArray.length; i++)
                                                pass.append(passArray[i]);
                                        if(userNameField.getText().length() < 5 || pass.length() <
5){
                                                if(formulaButtons[6].getForeground() == Color.RED ||
formulaButtons[7].getForeground() == Color.RED)
                                                        warning.setText("<html><div style=text-
align:center>*User Name and Password must contain atleast 5 characters*</div></html>");
                                                warning.setVisible(true);
                                                userExists = true;
                                        }
                                        else{
                                                try{
                                                        Scanner sc = new Scanner(new
File("./server/accounts/registeredUsers.txt"));
                                                        while(sc.hasNextLine()){
                                                                String temp = sc.nextLine();
                                                                String[] accountProperties =
temp.split("::");
                                                                if(formulaButtons[6].getForeground()
== Color.RED)

        if(accountProperties[0].compareTo(userNameField.getText()) == 0)
                                                                        userExists = true;
                                                                if(formulaButtons[7].getForeground()
== Color.RED){
                                                                        Scanner scTemp = new
Scanner(new File("./server/accounts/registeredUsers.txt"));
                                                                        boolean tempUserExists =
false;
                                                                        while(scTemp.hasNextLine()){
                                                                                String tempUser =
scTemp.nextLine();
                                                                                String[] userName =
tempUser.split("::");

        if(userName[0].compareTo(userNameField.getText()) == 0){
                                                                                        tempUserExists
= true;
                                                                                }
                                                                        }

        if((accountProperties[0].compareTo(userNameField.getText()) == 0) &&
(accountProperties[1].compareTo(pass.toString()) == 0)){
                                                                                signedIn = true;
                                                                                userExists = false;
                                                                                privileges =
Integer.parseInt(accountProperties[2]);
                                                                                model.clear();

        TableOfElements.tempSearchFormulaList.clear();
                                                                                new
File("./local/"+userNameField.getText()).mkdir();
                                                                                FileChannel source =
null;
                                                                        FileChannel destination =
null;
                                                                        try{
                                                                                source = new
FileInputStream("./server/userFiles/"+userNameField.getText()+"/compoundsList.txt").getChannel();
                                                                        }
                                                                        catch(IOException e){
                                                                                source = new
FileInputStream("./local/default/compoundsList.txt").getChannel();
                                                                        }
```

```
                                                                destination = new
FileOutputStream("./local/"+userNameField.getText()+"/compoundsList.txt").getChannel();
                                                                if(destination != null &&
source != null){
                                                                        try{

        destination.transferFrom(source, 0, source.size());
                                                                                }
                                                                        catch(IOException e){

        e.printStackTrace();
                                                                                }
                                                                }
                                                                        try{
                                                                                source.close();

        destination.close();
                                                                        }
                                                                        catch(IOException e) {

        e.printStackTrace();
                                                                        }

        searchComboBox.setSelectedIndex(0);
                                                                        model.clear();

        formulaButtons[7].setForeground(Color.BLACK);

        formulaButtons[0].setEnabled(signedIn);formulaButtons[1].setEnabled(signedIn);formulaButto
ns[2].setEnabled(signedIn);

        formulaButtons[3].setEnabled(signedIn);formulaButtons[4].setEnabled(signedIn);formulaButto
ns[5].setEnabled(signedIn);

        formulaButtons[6].setEnabled(!signedIn);formulaButtons[7].setEnabled(!signedIn);formulaBut
tons[8].setEnabled(signedIn);
                                                                }
                                                                else
if((accountProperties[0].compareTo(userNameField.getText()) == 0) &&
(accountProperties[1].compareTo(pass.toString()) != 0)){

        warning.setText("<html><div style=padding-left:50>*Wrong Password*</div></html>");
                                                                        userExists = true;
                                                                }
                                                                else if(!tempUserExists){

        warning.setText("<html><div style=padding-left:50>*Wrong Username*</div></html>");
                                                                        userExists = true;
                                                                }
                                                        }
                                                }
                                                sc.close();
                                                if(userExists)
                                                        warning.setVisible(true);
                                                else if(!userExists &&
formulaButtons[6].getForeground() == Color.RED){
                                                        try{

        formulaButtons[6].setForeground(Color.BLACK);
                                                                new
File("./server/userFiles/"+userNameField.getText()).mkdir();
                                                                BufferedWriter writer = new
BufferedWriter(new FileWriter("./server/accounts/registeredUsers.txt", true));

        writer.write(userNameField.getText()+"::"+pass+"::0");
                                                                writer.newLine();
                                                                writer.close();
                                                        }
                                                        catch(IOException exc){
                                                                exc.printStackTrace();
                                                        }
```

```
                                        }
                                }
                                catch(FileNotFoundException exc){
                                        exc.printStackTrace();
                                }
                        }
                        if(userExists){
                                userNameField.setText("");
                                userPassField.setText("");
                        }
                        else{
                                formulaButtons[5].setForeground(Color.BLACK);
                                regSignPanel.setVisible(false);
                                warning.setVisible(false);
                        }
                }
        });
        regSignPanel.add(confirm);
        regSignPanel.setVisible(false);
        buttonPanel.add(regSignPanel);

        //various button functions
        final int j = i;
        formulaButtons[i].addMouseListener(new MouseAdapter(){
                public void mouseClicked(MouseEvent evt){
                        if(j == 0){//Add
                                if(signedIn){
                                        final JDialog addDialog = new
JDialog(mainWindow.window, true);
                                        addDialog.setLayout(null);
                                        addDialog.setBackground(Color.LIGHT_GRAY);

                                        JLabel formulaEquationLabel = new
JLabel("Formula Equation:");
                                        formulaEquationLabel.setBounds(15, 30, 200,
20);
                                        addDialog.add(formulaEquationLabel);
                                        final JTextField formulaEquationField = new
JTextField();
                                        formulaEquationField.setBounds(130, 30, 535,
20);
                                        addDialog.add(formulaEquationField);
                                        JLabel inputExample = new
JLabel("<html><div>Input Example for Be(NO<sub>3</sub>)<sub>2</sub>3H<sub>2</sub>O is:
Be(NO_3)_2+3H_2O</div></html>");
                                        inputExample.setBounds(150, 50, 400, 20);
                                        inputExample.setFont(new Font("Arial",
Font.PLAIN, 10));
                                        addDialog.add(inputExample);

                                        JLabel formulaNameLabel = new JLabel("Formula
Name:");
                                        formulaNameLabel.setBounds(15, 80, 200, 20);
                                        addDialog.add(formulaNameLabel);
                                        final JTextField formulaNameField = new
JTextField();
                                        formulaNameField.setBounds(130, 80, 535, 20);
                                        addDialog.add(formulaNameField);

                                        JLabel formulaDescriptionLabel = new
JLabel("Formula Description:");
                                        formulaDescriptionLabel.setBounds(15, 120,
200, 20);
                                        addDialog.add(formulaDescriptionLabel);
                                        final JTextPane addTextPane = new
JTextPane();
                                        JScrollPane addScrollPane = new
JScrollPane(addTextPane);
                                        Font myFont = new Font("Courier New",
Font.PLAIN, 14);
                                        addTextPane.setFont(myFont);
```

```
                                                addScrollPane.setBounds(15, 140, 650, 300);
                                                addDialog.add(addScrollPane);

                                                JButton yes = new JButton(), no = new
JButton();

                                                yes.setBounds(543, 450, 60,20);
                                                yes.setMargin(new Insets(0, 0, 0, 0));
                                                yes.setText("Accept");
                                                yes.setFocusable(false);
                                                yes.addActionListener(new ActionListener(){
                                                        public void
actionPerformed(ActionEvent evt){
                                                                String enteredName =
formulaNameField.getText();
                                                                boolean compoundFound = false;
                                                                error.setBounds(250, 450, 250,
20);

        error.setForeground(Color.RED);
                                                                error.setText("");
                                                                error.setVisible(false);
                                                                addDialog.add(error);

                                                                File inFile = new
File("./local/"+userNameField.getText()+"/compoundsList.txt");
                                                                BufferedReader br = null;
                                                                PrintWriter pw = null;
                                                                try {
                                                                        br = new
BufferedReader(new FileReader(inFile));
                                                                        pw = new
PrintWriter(new FileWriter(inFile, true));
                                                                        String line = null;
                                                                        while((line =
br.readLine()) != null){

        if(line.length()>2 && line.substring(0, 2).equals("..")){

        String[] splitLine = line.split("::");

        if(enteredName != null && splitLine[1].equalsIgnoreCase(enteredName)){

        compoundFound = true;

        error.setText("*Compound already exists*");

        break;
                                                                                        }
                                                                                }
                                                                }
                                                                br.close();
                                                                if(!compoundFound &&
formulaNameField.getText().length()>0 && formulaEquationField.getText().length()>1){
                                                                        StringBuffer
fixedName = new StringBuffer(formulaNameField.getText());

        if(fixedName.charAt(0) >= 'a' && fixedName.charAt(0) <= 'z'){

        fixedName.replace(0, 1, Character.toString((char)(fixedName.charAt(0) - 32)));
                                                                        }
                                                                        pw.println();

        pw.println(".."+formulaEquationField.getText()+"::"+fixedName.toString());

        pw.println(addTextPane.getText());
                                                                        pw.println();
                                                                        pw.print(";;");

        addDialog.dispose();
                                                                }
                                                                else{
```

```
        if(formulaNameField.getText().length()<1 || formulaEquationField.getText().length()<1)

        error.setText("*Enter formula name and equation*");

        error.setVisible(true);
                                                                        }
                                                                        pw.close();
                                                                }
                                                                catch(IOException e){
                                                                        e.printStackTrace();
                                                                }
                                                        }
                                                });
                                                addDialog.add(yes);

                                                no.setBounds(605, 450, 60, 20);
                                                no.setMargin(new Insets(0, 0, 0, 0));
                                                no.setText("Cancel");
                                                no.setFocusable(false);
                                                no.addActionListener(new ActionListener(){
                                                        public void
actionPerformed(ActionEvent evt){
                                                                addDialog.dispose();
                                                        }
                                                });
                                                addDialog.add(no);

                                                addDialog.setResizable(false);

        addDialog.setBounds(mainWindow.window.getX()+200, mainWindow.window.getY()+120, 675, 500);

        addDialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
                                                addDialog.setVisible(true);
                                        }
                                }
                                if(j == 1){//Edit
                                        if(signedIn)
                                                if(list.getLeadSelectionIndex() != -1 &&
selectedItemIndex != -1){
                                                        final JDialog editDialog = new
JDialog(mainWindow.window, true);
                                                        editDialog.setLayout(null);

        editDialog.setBackground(Color.LIGHT_GRAY);

                                                        JLabel formulaEquationLabel = new
JLabel("Formula Equation:");
                                                        formulaEquationLabel.setBounds(15,
30, 200, 20);
                                                        editDialog.add(formulaEquationLabel);
                                                        final JTextField formulaEquationField
= new JTextField();
                                                        formulaEquationField.setBounds(130,
30, 535, 20);
                                                        editDialog.add(formulaEquationField);
                                                        JLabel inputExample = new
JLabel("<html><div>Input Example for Be(NO<sub>3</sub>)<sub>2</sub>3H<sub>2</sub>O is:
Be(NO_3)_2+3H_2O</div></html>");
                                                        inputExample.setBounds(150, 50, 400,
20);
                                                        inputExample.setFont(new
Font("Arial", Font.PLAIN, 10));
                                                        editDialog.add(inputExample);

                                                        JLabel formulaNameLabel = new
JLabel("Formula Name:");
                                                        formulaNameLabel.setBounds(15, 80,
200, 20);
                                                        editDialog.add(formulaNameLabel);
```

```
new JTextField();
535, 20);

JLabel("Formula Description:");
120, 200, 20);
        editDialog.add(formulaDescriptionLabel);
JTextPane();
JScrollPane(editTextPane);
Font.PLAIN, 14);

650, 300);

File("./local/"+userNameField.getText()+"/compoundsList.txt");
File(inFile.getAbsolutePath() + ".tmp");

FileReader(inFile));
FileWriter(tempFile, true));

!= null){
line.substring(0, 2).equals("..")){
splitLine = line.split("::");
        if(splitLine[1].equalsIgnoreCase(name.toString())){
        formulaNameField.setText(splitLine[1]);
        formulaEquationField.setText(splitLine[0].substring(2));
        Document doc = editTextPane.getDocument();
        br.readLine();
br.readLine()) != null){
        if(line.equals(";;")){
        endOfCompound = true;
        break;
        }
        doc.insertString(doc.getLength(), line+"\n", null);
        catch(BadLocationException exc){
exc.printStackTrace();
```

```
final JTextField formulaNameField =
formulaNameField.setBounds(130, 80,
editDialog.add(formulaNameField);
JLabel formulaDescriptionLabel = new
formulaDescriptionLabel.setBounds(15,

final JTextPane editTextPane = new
JScrollPane editScrollPane = new
Font myFont = new Font("Courier New",
editTextPane.setFont(myFont);
editScrollPane.setBounds(15, 140,
editDialog.add(editScrollPane);

final File inFile = new
);
final File tempFile = new

BufferedReader br = null;
try{
        br = new BufferedReader(new
        pw = new PrintWriter(new
        String line = null;
        boolean endOfCompound = false;
        while((line = br.readLine())
                if(line.length()>2 &&
                        String[]
                                while((line =
                                        try{
                                }
                        }
                }
```

```
                                                                try{
        if(doc.getLength()>1)
        doc.remove(doc.getLength()-1, 1);
                                                                }
        catch(BadLocationException exc){
exc.printStackTrace();
                                                                }
                                                                }
                                                        }
                                                        if(line != null &&
!endOfCompound)
        pw.println(line);
                                                                endOfCompound = false;
                                                        }
                                                        br.close();
                                                }
                                                catch(IOException exc){
                                                        exc.printStackTrace();
                                                }

                                                JButton yes = new JButton(), no = new
JButton();

                                                yes.setBounds(543, 450, 60,20);
                                                yes.setMargin(new Insets(0, 0, 0,
0));

                                                yes.setText("Accept");
                                                yes.setFocusable(false);
                                                yes.addActionListener(new
ActionListener(){
                                                        public void
actionPerformed(ActionEvent evt){
                                                                error.setBounds(250,
450, 250, 20);
        error.setForeground(Color.RED);
                                                                error.setText("");
        error.setVisible(false);
                                                                editDialog.add(error);

        if(formulaNameField.getText().length()>0 && formulaEquationField.getText().length()>1){
                                                                        StringBuffer
fixedName = new StringBuffer(formulaNameField.getText());
        if(fixedName.charAt(0) >= 'a' && fixedName.charAt(0) <= 'z'){
        fixedName.replace(0, 1, Character.toString((char)(fixedName.charAt(0) - 32)));
                                                                        }
        pw.println(".."+formulaEquationField.getText()+"::"+fixedName.toString());
        pw.println(editTextPane.getText());
                                                                pw.println();
                                                                pw.print(";;");
                                                                pw.flush();
        editDialog.dispose();
                                                        }
                                                        else{
        if(formulaNameField.getText().length()<1 || formulaEquationField.getText().length()<1)
        error.setText("*Enter formula name and equation*");
        error.setVisible(true);
```

```
                                                                }
                                                                pw.close();
                                                                inFile.delete();

        tempFile.renameTo(inFile);
                                                        }
                                                });
                                                editDialog.add(yes);

                                                no.setBounds(605, 450, 60, 20);
                                                no.setMargin(new Insets(0, 0, 0, 0));
                                                no.setText("Cancel");
                                                no.setFocusable(false);
                                                no.addActionListener(new
ActionListener(){
                                                        public void
actionPerformed(ActionEvent evt){
                                                                pw.close();
                                                                tempFile.delete();
                                                                editDialog.dispose();
                                                        }
                                                });
                                                editDialog.add(no);

                                                editDialog.setResizable(false);

        editDialog.setBounds(mainWindow.window.getX()+200, mainWindow.window.getY()+120, 675,
500);

        editDialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
                                                editDialog.setVisible(true);
                                        }
                                }
                                if(j == 2){//Remove
                                        if(signedIn){
                                                if(list.getLeadSelectionIndex() != -1 &&
selectedItemIndex != -1){
                                                final JDialog removeDialog = new
JDialog(mainWindow.window, true);
                                                JPanel textRemovePanel = new
JPanel();

        textRemovePanel.setBackground(Color.LIGHT_GRAY);
                                                JLabel removeText = new
JLabel("<html><div style=text-align:center>Do you really want to remove <b
style=color:#C35617>"+name+"</b> from your database?</div></html>");
                                                removeText.setPreferredSize(new
Dimension(280, 25));
                                                textRemovePanel.add(removeText);

                                                JPanel simple2ButtonPanel = new
JPanel();
                                                simple2ButtonPanel.setLayout(null);

        simple2ButtonPanel.setPreferredSize(new Dimension(300, 35));

        simple2ButtonPanel.setBackground(Color.LIGHT_GRAY);
                                                JButton yes = new JButton(), no = new
JButton();
                                                yes.setBounds(208, 16, 40,18);
                                                yes.setMargin(new Insets(0, 0, 0,
0));
                                                yes.setText("Yes");
                                                yes.setFocusable(false);
                                                yes.addActionListener(new
ActionListener(){
                                                        public void
actionPerformed(ActionEvent evt){
                                                                File inFile = new
File("./local/"+userNameField.getText()+"/compoundsList.txt");
```

```
File(inFile.getAbsolutePath() + ".tmp");
null;


BufferedReader(new FileReader(inFile));
PrintWriter(new FileWriter(tempFile));
null;
br.readLine()) != null){
        if(line.length()>2 && line.substring(0, 2).equals("..")){
        String[] splitLine = line.split("::");
        if(splitLine[1].equals(name.toString()))
        while((line = br.readLine()) != ";;"){
                if(line.equals(";;")){
                        line = br.readLine();
                        break;
                }
        }
!= null)
        pw.println(line);



        e.printStackTrace();


        tempFile.renameTo(inFile);
        removeDialog.dispose();
        model.remove(list.getSelectedIndex());
        TableOfElements.tempSearchFormulaList.clear();
                                                                    }
                                                                });

                                                                no.setBounds(250, 16, 40,18);
                                                                no.setMargin(new Insets(0, 0, 0, 0));
                                                                no.setText("No");
                                                                no.setFocusable(false);
                                                                no.addActionListener(new
ActionListener(){
actionPerformed(ActionEvent evt){
        removeDialog.dispose();
                                                                        public void


                                                                    }
                                                                });
                                                                simple2ButtonPanel.add(yes);
                                                                simple2ButtonPanel.add(no);
```

```
File tempFile = new
BufferedReader br =
PrintWriter pw = null;
try {
        br = new
        pw = new
        String line =
        while((line =
                        }
                        if(line
        }
        pw.close();
        br.close();
}
catch(IOException e){

}
inFile.delete();
```

```
        textRemovePanel.add(simple2ButtonPanel, BorderLayout.LINE_END);
                                                        removeDialog.add(textRemovePanel);

                                                        removeDialog.setResizable(false);

        removeDialog.setBounds(mainWindow.window.getX()+700, mainWindow.window.getY()+520, 300,
100);

        removeDialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
                                                        removeDialog.setVisible(true);
                                                }
                                        }
                                }
                                if(j == 3){//Import
                                        if(signedIn){
                                                final JDialog importDialog = new
JDialog(mainWindow.window, true);

                                                JPanel simple2ButtonPanel = new JPanel();
                                                simple2ButtonPanel.setLayout(null);
                                                simple2ButtonPanel.setPreferredSize(new
Dimension(300, 35));

        simple2ButtonPanel.setBackground(Color.LIGHT_GRAY);

                                                final JRadioButton importDefault = new
JRadioButton("<html>Import from <b style=color:#C35617>default</b> database</html>");
                                                importDefault.setBounds(10, 5, 190, 18);

        importDefault.setBackground(Color.LIGHT_GRAY);
                                                importDefault.setFocusable(false);
                                                simple2ButtonPanel.add(importDefault);

                                                final JRadioButton importUser = new
JRadioButton("<html>Import from <b style=color:#C35617>your</b> database</html>");
                                                importUser.setBounds(10, 23, 180, 18);
                                                importUser.setBackground(Color.LIGHT_GRAY);
                                                importUser.setFocusable(false);
                                                simple2ButtonPanel.add(importUser);

                                                ButtonGroup importOption = new ButtonGroup();
                                                importOption.add(importDefault);
                                                importOption.add(importUser);

                                                JButton yes = new JButton(), no = new
JButton();
                                                yes.setBounds(208, 51, 40,18);
                                                yes.setMargin(new Insets(0, 0, 0, 0));
                                                yes.setText("Yes");
                                                yes.setFocusable(false);
                                                yes.addActionListener(new ActionListener(){
                                                        public void
actionPerformed(ActionEvent evt){
                                                                File inFile = null;
                                                                if(importUser.isSelected())
                                                                        inFile = new
File("./server/userFiles/"+userNameField.getText()+"/compoundsList.txt");
                                                                else
if(importDefault.isSelected())
                                                                        inFile = new
File("./server/userFiles/default/compoundsList.txt");

                                                                if(importUser.isSelected() ||
importDefault.isSelected()){
                                                                        File tempFile = new
File(new File("local/"+userNameField.getText()+"/compoundsList.txt").getAbsoluteFile()+".tmp");
                                                                        BufferedReader br =
null;
                                                                        PrintWriter pw = null;
                                                                        try {
```

```
                                                                br = new
BufferedReader(new FileReader(inFile));
                                                                pw = new
PrintWriter(new FileWriter(tempFile));
                                                                String line =
null;
                                                                while((line =
br.readLine()) != null)

        pw.println(line);


                                                                pw.close();
                                                                br.close();
                                                        }
                                                        catch(IOException e){

        e.printStackTrace();

                                                        }
                                                        File localFile = new
File("local/"+userNameField.getText()+"/compoundsList.txt");
                                                        localFile.delete();

        tempFile.renameTo(localFile);


                                                        model.clear();

        TableOfElements.tempSearchFormulaList.clear();

        importDialog.dispose();
                                                                }
                                                        }
                                                });

                                                no.setBounds(250, 51, 40,18);
                                                no.setMargin(new Insets(0, 0, 0, 0));
                                                no.setText("No");
                                                no.setFocusable(false);
                                                no.addActionListener(new ActionListener(){
                                                        public void
actionPerformed(ActionEvent evt){
                                                                importDialog.dispose();
                                                        }
                                                });

                                                simple2ButtonPanel.add(yes);
                                                simple2ButtonPanel.add(no);
                                                importDialog.add(simple2ButtonPanel);

                                                importDialog.setResizable(false);

        importDialog.setBounds(mainWindow.window.getX()+700, mainWindow.window.getY()+520, 300,
100);

        importDialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
                                                importDialog.setVisible(true);
                                        }
                                }
                                if(j == 4){//Export
                                        if(signedIn){
                                                final JDialog exportDialog = new
JDialog(mainWindow.window, true);
                                                JPanel textExportPanel = new JPanel();

        textExportPanel.setBackground(Color.LIGHT_GRAY);
                                                JLabel exportText = new JLabel("<html><div
style=text-align:center>Export current formula list to your database?</div></html>");
                                                exportText.setPreferredSize(new
Dimension(260, 25));
                                                textExportPanel.add(exportText);

                                                JPanel simple2ButtonPanel = new JPanel();
                                                simple2ButtonPanel.setLayout(null);
```

```
                                                        simple2ButtonPanel.setPreferredSize(new
Dimension(300, 35));

        simple2ButtonPanel.setBackground(Color.LIGHT_GRAY);

                                                        final JCheckBox defaultExport = new
JCheckBox("And to the default server database");
                                                        if(privileges == 1){

        defaultExport.setBackground(Color.LIGHT_GRAY);
                                                                defaultExport.setFont(new
Font("Arial", Font.PLAIN, 10));
                                                                defaultExport.setFocusable(false);
                                                                defaultExport.setMargin(new Insets(0,
0, 0, 0));
                                                                defaultExport.setBounds(18, 16, 185,
20);

        simple2ButtonPanel.add(defaultExport);
                                                        }

                                                        JButton yes = new JButton(), no = new
JButton();
                                                        yes.setBounds(208, 16, 40,18);
                                                        yes.setMargin(new Insets(0, 0, 0, 0));
                                                        yes.setText("Yes");
                                                        yes.setFocusable(false);
                                                        yes.addActionListener(new ActionListener(){
                                                                public void
actionPerformed(ActionEvent evt){
                                                                        File inFile, tempFile,
serverFile;

        if(defaultExport.isSelected()){
                                                                                inFile = new
File("./local/"+userNameField.getText()+"/compoundsList.txt");
                                                                                tempFile = new
File(new File("server/userFiles/default/compoundsList.txt").getAbsoluteFile()+".tmp");
                                                                                BufferedReader br =
null;
                                                                                PrintWriter pw = null;
                                                                                try {
                                                                                        br = new
BufferedReader(new FileReader(inFile));
                                                                                        pw = new
PrintWriter(new FileWriter(tempFile));
                                                                                        String line =
null;
                                                                                        while((line =
br.readLine()) != null)

        pw.println(line);
                                                                                        pw.close();
                                                                                        br.close();
                                                                                }
                                                                                catch(IOException e){

        e.printStackTrace();
                                                                                }
                                                                                serverFile = new
File("server/userFiles/default/compoundsList.txt");
                                                                                serverFile.delete();

        tempFile.renameTo(serverFile);
                                                                        }

                                                                        inFile = new
File("./local/"+userNameField.getText()+"/compoundsList.txt");
                                                                        tempFile = new File(new
File("server/userFiles/"+userNameField.getText()+"/compoundsList.txt").getAbsolutePath()+".tmp");
                                                                        BufferedReader br = null;
```

```
                                                                PrintWriter pw = null;
                                                                try {
                                                                        br = new
BufferedReader(new FileReader(inFile));
                                                                        pw = new
PrintWriter(new FileWriter(tempFile));
                                                                        String line = null;
                                                                        while((line =
br.readLine()) != null)

        pw.println(line);
                                                                        pw.close();
                                                                        br.close();
                                                                }
                                                                catch(IOException e){
                                                                        e.printStackTrace();
                                                                }
                                                                serverFile = new
File("server/userFiles/"+userNameField.getText()+"/compoundsList.txt");
                                                                serverFile.delete();
                                                                tempFile.renameTo(serverFile);
                                                                exportDialog.dispose();
                                                        }
                                                });

                                                no.setBounds(250, 16, 40,18);
                                                no.setMargin(new Insets(0, 0, 0, 0));
                                                no.setText("No");
                                                no.setFocusable(false);
                                                no.addActionListener(new ActionListener(){
                                                        public void
actionPerformed(ActionEvent evt){
                                                                exportDialog.dispose();
                                                        }
                                                });

                                                simple2ButtonPanel.add(yes);
                                                simple2ButtonPanel.add(no);
                                                textExportPanel.add(simple2ButtonPanel,
BorderLayout.LINE_END);
                                                exportDialog.add(textExportPanel);

                                                exportDialog.setResizable(false);

        exportDialog.setBounds(mainWindow.window.getX()+700, mainWindow.window.getY()+520, 300,
100);

        exportDialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
                                                        exportDialog.setVisible(true);
                                                }
                                        }
                                        if(j == 5){//Update
                                                final JDialog updateDialog = new
JDialog(mainWindow.window, true);
                                                JPanel textUpdatePanel = new JPanel();
                                                textUpdatePanel.setBackground(Color.LIGHT_GRAY);
                                                JLabel updateText = new JLabel("<html><div
style=text-align:center>Update default formula list?</div></html>");
                                                updateText.setPreferredSize(new Dimension(160, 25));
                                                textUpdatePanel.add(updateText);

                                                JPanel simple2ButtonPanel = new JPanel();
                                                simple2ButtonPanel.setLayout(null);
                                                simple2ButtonPanel.setPreferredSize(new
Dimension(300, 35));
                                                simple2ButtonPanel.setBackground(Color.LIGHT_GRAY);

                                                JButton yes = new JButton(), no = new JButton();
                                                yes.setBounds(208, 16, 40,18);
                                                yes.setMargin(new Insets(0, 0, 0, 0));
                                                yes.setText("Yes");
```

```
                                        yes.setFocusable(false);
                                        yes.addActionListener(new ActionListener(){
                                                public void actionPerformed(ActionEvent evt){
                                                        File inFile = new
File("./server/userFiles/default/compoundsList.txt");
                                                        File tempFile = new File(new
File("local/default/compoundsList.txt").getAbsolutePath()+".tmp");
                                                        BufferedReader br = null;
                                                        PrintWriter pw = null;
                                                        try {
                                                                br = new BufferedReader(new
FileReader(inFile));
                                                                pw = new PrintWriter(new
FileWriter(tempFile));
                                                                String line = null;
                                                                while((line = br.readLine())
!= null)
                                                                        pw.println(line);
                                                                pw.close();
                                                                br.close();
                                                        }
                                                        catch(IOException e){
                                                                e.printStackTrace();
                                                        }
                                                        File localFile = new
File("local/default/compoundsList.txt");
                                                        localFile.delete();
                                                        tempFile.renameTo(localFile);

                                                        model.clear();

        TableOfElements.tempSearchFormulaList.clear();
                                                        updateDialog.dispose();
                                                }
                                        });

                                        no.setBounds(250, 16, 40,18);
                                        no.setMargin(new Insets(0, 0, 0, 0));
                                        no.setText("No");
                                        no.setFocusable(false);
                                        no.addActionListener(new ActionListener(){
                                                public void actionPerformed(ActionEvent evt){
                                                        updateDialog.dispose();
                                                }
                                        });

                                        simple2ButtonPanel.add(yes);
                                        simple2ButtonPanel.add(no);
                                        textUpdatePanel.add(simple2ButtonPanel,
BorderLayout.LINE_END);
                                        updateDialog.add(textUpdatePanel);

                                        updateDialog.setResizable(false);
                                        updateDialog.setBounds(mainWindow.window.getX()+700,
mainWindow.window.getY()+520, 300, 100);

        updateDialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
                                        updateDialog.setVisible(true);
                                }
                                if(j == 6){//Sign up
                                        userNameField.setText("");
                                        userPassField.setText("");

        TableOfElements.warningNoElementPanel.setVisible(false);
                                        if(formulaButtons[6].getForeground() == Color.RED){
                                                warning.setVisible(false);
                                                regSignPanel.setVisible(false);
                                                formulaButtons[6].setForeground(Color.BLACK);
                                        }
                                        else{
```

```
                                                        if(formulaButtons[7].getForeground() ==
Color.RED){
                                                                warning.setVisible(false);
                                                                regSignPanel.setVisible(false);

        formulaButtons[7].setForeground(Color.BLACK);
                                                        }
                                                        regSignPanel.setVisible(true);
                                                        formulaButtons[j].setForeground(Color.RED);
                                                }
                                        }
                                        if(j == 7){//Sign in
                                                userNameField.setText("");
                                                userPassField.setText("");
                                                PropertyList.pane.setText("");

        TableOfElements.warningNoElementPanel.setVisible(false);
                                                if(formulaButtons[7].getForeground() == Color.RED){
                                                        warning.setVisible(false);
                                                        regSignPanel.setVisible(false);
                                                        formulaButtons[7].setForeground(Color.BLACK);
                                                }
                                                else{
                                                        if(formulaButtons[6].getForeground() ==
Color.RED){
                                                                warning.setVisible(false);
                                                                regSignPanel.setVisible(false);

        formulaButtons[6].setForeground(Color.BLACK);
                                                        }
                                                        regSignPanel.setVisible(true);
                                                        formulaButtons[j].setForeground(Color.RED);
                                                }
                                        }
                                        if(j == 8){//Sign out
                                                signedIn=false;
                                                privileges = 0;
                                                model.clear();
                                                PropertyList.pane.setText("");
                                                TableOfElements.tempSearchFormulaList.clear();

        formulaButtons[0].setEnabled(signedIn);formulaButtons[1].setEnabled(signedIn);formulaButto
ns[2].setEnabled(signedIn);

        formulaButtons[3].setEnabled(signedIn);formulaButtons[4].setEnabled(signedIn);formulaButto
ns[5].setEnabled(!signedIn);

        formulaButtons[6].setEnabled(!signedIn);formulaButtons[7].setEnabled(!signedIn);formulaBut
tons[8].setEnabled(signedIn);
                                                File directory = new
File("./local/"+userNameField.getText());
                                                String[] fileInDirectory = directory.list();
                                           for(int i=0; i<fileInDirectory.length; i++){
                                                File currentFile = new
File("./local/"+userNameField.getText()+"/"+fileInDirectory[i]);
                                                        currentFile.delete();
                                                }
                                         directory.delete();
                                        }
                                }
                        });
                        buttonPanel.add(formulaButtons[i]);
                }
                masterPanel.add(buttonPanel);

                return masterPanel;
        }
}
```

Razred PropertyList

```
import java.awt.*;
import javax.swing.*;

public class PropertyList {
	protected static JTextPane pane = new JTextPane();
	protected static JScrollPane scrollPane = new JScrollPane(pane);

	public JPanel propPanel(){
		pane.setEditable(false);
		JPanel masterPanel = new JPanel();
		masterPanel.setBackground(Color.BLACK);
		Font myFont = new Font("Arial", Font.PLAIN, 14);
		pane.setForeground(Color.ORANGE);
		pane.setBackground(new Color(50, 50, 50));
		pane.setFont(myFont);
		scrollPane.setPreferredSize(new Dimension(750, 300));
		masterPanel.add(scrollPane);

		return masterPanel;
	}
}
```