

Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

## BAKALÁŘSKÁ PRÁCE



Lenka Kreibichová

Chemická laboratoř

Katedra softwarového inženýrství

Vedoucí bakalářské práce: RNDr. Filip Zavoral, Ph.D.

Studijní program: Informatika, programování

2006

Děkuji panu RNDr. Filipu Zavoralovi, Ph.D. za odborné vedení mé práce, za rady a za čas, který mi během vypracování této práce věnoval.

Prohlašuji, že jsem svou bakalářskou práci napsala samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 11.8.2006

Lenka Kreibichová

# Obsah

<b>1</b>	<b>Úvod</b>	<b>6</b>
<b>2</b>	<b>Výchozí teorie</b>	<b>8</b>
2.1	Základní pojmy . . . . .	8
2.2	Změny teploty . . . . .	9
2.3	Změny skupenství . . . . .	9
2.4	Rozpustnost . . . . .	10
2.5	Chemické reakce . . . . .	10
2.5.1	Vliv koncentrace . . . . .	11
2.5.2	Vliv teploty . . . . .	11
2.5.3	Další vlivy . . . . .	11
<b>3</b>	<b>Implementace</b>	<b>12</b>
3.1	Požadavky na program . . . . .	12
3.1.1	Grafické rozhraní . . . . .	12
3.1.2	Implementace chemické a fyzikální teorie . . . . .	12
3.1.3	Řízení pokusů . . . . .	13
3.2	Problémy a jejich řešení - grafické prostředí . . . . .	13
3.2.1	Základní rozdělení aplikace . . . . .	13
3.2.2	Vykreslování . . . . .	13
3.2.3	Posuvníky . . . . .	15
3.2.4	Podokna . . . . .	16
3.2.5	Operace Zpět/Vpřed . . . . .	16
3.2.6	Kopírování a mazání objektů . . . . .	17
3.3	Problémy a jejich řešení - výpočetní model . . . . .	17
3.3.1	Základní datové struktury . . . . .	17
3.3.2	Datové typy . . . . .	18
3.3.3	Změny teploty . . . . .	19
3.3.4	Vyhledávání reakcí . . . . .	20
3.3.5	Výpočet rychlosti reakcí . . . . .	20
3.3.6	Editace skriptu . . . . .	20
3.3.7	Komunikace programu a skriptu . . . . .	21

<b>4</b>	<b>Uživatelská dokumentace</b>	<b>22</b>
4.1	Instalace programu . . . . .	22
4.2	Spuštění programu . . . . .	22
4.3	Ovládání chemické laboratoře . . . . .	23
4.3.1	Objekty v laboratoři . . . . .	23
4.3.2	Vkládání chemických látek do baňky . . . . .	24
4.4	Editace řídicích scénářů . . . . .	25
4.4.1	Řídicí krok . . . . .	26
4.4.2	Spuštění skriptu . . . . .	28
4.5	Editace chemické databáze . . . . .	28
4.5.1	Databáze chemických látek . . . . .	28
4.5.2	Databáze chemických reakcí . . . . .	31
4.6	Podrobný popis pracovního prostředí . . . . .	32
4.6.1	Menu a toolbar . . . . .	32
4.6.2	Podokna . . . . .	34
4.6.3	Ovládání . . . . .	34
4.6.4	Nastavení . . . . .	34
<b>5</b>	<b>Srovnání s jinými programy</b>	<b>35</b>
5.1	ChemLab 2.3 . . . . .	35
<b>6</b>	<b>Závěr</b>	<b>37</b>
6.1	Další budoucnost programu . . . . .	37
	<b>Literatura</b>	<b>38</b>
<b>A</b>	<b>XML Schema souboru látek</b>	<b>39</b>
<b>B</b>	<b>XML Schema souboru reakcí</b>	<b>40</b>
<b>C</b>	<b>XML Schema uložené laboratoře</b>	<b>41</b>
<b>D</b>	<b>XML Schema řídicího skriptu</b>	<b>42</b>

Název práce: Chemická laboratoř  
Autor: Lenka Kreibichová  
Katedra: Katedra softwarového inženýrství  
Vedoucí bakalářské práce: RNDr. Filip Zavoral, Ph.D.  
e-mail vedoucího: Filip.Zavoral@mff.cuni.cz

Abstrakt: Předmětem této práce je implementace simulátoru chemické laboratoře. Výsledný program je určen pro uživatele, kteří si chtějí vyzkoušet, jak vypadá práce v chemické laboratoři, aniž by byli nuceni do ní vstoupit. V prostředí laboratoře bude je vytvářet si baňky a zkumavky různého tvaru, přidávat do nich chemické látky, ohřívat je nebo chladit, mísit obsahy nádob a sledovat jejich reakce.

Významnou vlastností je možnost definovat posloupnosti řídicích kroků, které postupně korigují činnost uživatele v laboratoři.

Klíčová slova: chemie, laboratoř, reakce

Title: Název bakalářské práce v angličtině  
Author: Lenka Kreibichová  
Department: Department of Software Engineering  
Supervisor: RNDr. Filip Zavoral, Ph.D.  
Supervisor's e-mail address: Filip.Zavoral@mff.cuni.cz

Abstract: The subject of this work is to implement the simulation of chemistry laboratory. The result is specified for user, who would like to test, how it works in a chemistry laboratory without entering the real room. In the virtual laboratory you can use test tubes and flasks of various shape, insert chemical matters, warm and old them, mix the contents of the flasks and watch the reactions.

The important merit is the ability to set the series of instructions, that interactive help user to go through the chemical experiment.

Keywords: chemistry, laboratory, reaction

# Kapitola 1

## Úvod

V dnešní době patří chemie vedle informatiky k nejvíce využívaným a rozvíjeným vědním oborům a je nedílnou součástí většiny průmyslových odvětví. Ať se podíváme do libovolné výrobní firmy, všude je důležité vyvíjet parametry výrobků tak, aby výsledné produkty měly co nejlepší výsledné vlastnosti, odpovídaly předpisům či dokonce nebyly zdraví závadné — potravinářská chemie se zabývá vhodnými i nežádoucími změnami, kterým potraviny podléhají v průběhu jejich výroby, uchovávání a zpracování, a to jejich podpořením i potlačením, chemie životního prostředí zase ochranou a úpravami vody či zpracováním odpadů, chemie materiálů se stará o dosažení přesně vymezených parametrů stavebních materiálů, hnojiv či kosmetických produktů, farmakologická chemie se soustřeďuje na syntézu nových kvalitnějších léčiv.

Tento obor je tedy natolik důležitý, že se základy jeho výuky se začíná již na druhém stupni základní školy. Věda je vyučována nejen jako teoretické učivo v posluchárnách, žák si nejlépe zapamatuje nové poznatky metodou praktického poznávání s možností sám si chemické vlastnosti vyzkoušet, nejlépe rovnou v chemické laboratoři. Základní školy bohužel často nemají prostředky, prostředí ani možnosti ke zřízení těchto laboratoří, děti plnící základní školní docházku také ještě někdy nejsou schopné dodržovat přesně předepsané předpisy a bezpečnostní opatření, pokus se často nepovede a na jeho opětovné založení již nezbyvá dostatek času či není možné plýtvat chemickými surovinami.

Proto by bylo vhodné umožnit žákům či lidem, kteří by si chtěli vyzkoušet, jak práce v chemické laboratoři může vypadat, aby si mohli spustit počítač, zapnout vhodný program a odpřednášené učivo vyzkoušet sami bez nutnosti nakupovat drahé chemické aparáty či bez strachu z výbuchu při nepovedeném pokusu.

Nevýhodou je, že výzkum v oboru chemie pravděpodobně nikdy nebude v takové fázi, kdy by bylo možné obecně popsat všechny vlastnosti chemických látek a jejich vzájemné interakce, natož pak vyčíslit tyto vztahy pomocí vzorců a rovnic. Již tak je v této době chemická nauka ovšem velmi rozsáhlá.

V průběhu práce vznikl program CHEMICA, který simuluje základy chemických dějů a fyzikálních vlastností, které mohou zajímat laika, nemajícího odbornou prů-

pravu ve složitých chemických disciplínách, a které jsou již natolik probádané, že je lze s určitou nadsázkou bez ohledu na působení složitějších vlastností obecně popsat. Simulace programu vychází z teoretických základů ve druhé kapitole.

Významnou funkcí programu, využitelnou zejména pro práci ve škole, je možnost vytvářet tzv. *skripty* — spustitelné soubory obsahující definici posloupnosti kroků, vedoucí studenta ke zdárnému dokončení pokusu pomocí poznámek, které se studentovi v průběhu práce zobrazují a vyčkávají, až student žádanou akci vykoná.

Třetí kapitola obsahuje uživatelskou dokumentaci, která slouží pro orientaci při užívání programu i pro přehled všech jeho funkcí. V další kapitole jsou představeny problémy, které vznikly při implementaci a jak byly řešeny. Poslední kapitoly srovnávají program CHEMICA s konkurenčním programem a nastiňují další jeho možný vývoj. Nedílnou přílohou této práce je přiložené CD, které obsahuje:

- instalační soubor programu Chemica
- elektronickou verzi tohoto textu
- všechny zdrojové soubory programu
- specifikaci

V podadresáři *skripty* hlavního adresáře programu jsou vzorové příklady, které je možné si vyzkoušet.

# Kapitola 2

## Výchozí teorie

### 2.1 Základní pojmy

Látky jsou formy hmoty, které se skládají z částic (atomů, molekul či složitějších útvarů). Vlastnosti jednotlivých látek jsou závislé na struktuře a vzájemném postavení jednotlivých částic.

Pro výpočty či pro získání informací o vlastnostech látky jsou důležité tyto fyzikální veličiny:

1. **Relativní molekulová hmotnost** — skutečné hmotnosti jednotlivých molekul a atomů jsou velmi malá čísla, se kterými by bylo obtížné počítat. Proto se zavádí pojem *atomová hmotnostní jednotka*, definovaná jako  $\frac{1}{12}$  hmotnosti atomu uhlíku. Relativní molekulová hmotnost látky  $M_r$  se počítá jako součet relativních atomových hmotností všech atomů, které tvoří tuto molekulu. Pro výpočet skutečné hmotnosti molekuly platí vztah [4]

$$m = M_r \cdot m_u$$

kde  $m_u$  je atomová hmotnostní jednotka.

2. Pro chemické výpočty se více hodí veličina **Látkové množství**, která charakterizuje množství částic v látce. Jednotkou je jeden *mol*, obsahující  $6,023 \cdot 10^{23}$  částic[4].
3. **Látková koncentrace** se užívá při výpočtu rychlosti chemické reakce. Vyjadřuje látkové množství rozpuštěné látky v  $1 \text{ dm}^3$  roztoku. Látková koncentrace  $c(A)$  látky  $A$  je v daném roztoku rovna podílu látkového množství této látky  $n(A)$  a objemu roztoku  $V$  [4]:

$$c(A) = \frac{n(A)}{V}$$



## 2.2 Změny teploty

O teplotě platí tento poznatek:

Jestliže mají dvě různá tělesa  $A$ ,  $B$  různé teploty  $T_A$ ,  $T_B$  a jsou v tepelném kontaktu, jejich teploty se vyrovnávají, až dosáhnou teploty  $T$ .

Z toho vyplývá, že teplotu lze přiřadit pouze takové termodynamické soustavě, ve které jsou všechny části v tepelné rovnováze. V baňce dochází k poměrně rychlému promíšení jejích částí, proto jí můžeme jednotnou teplotu přiřadit (vyzařování tepla do okolí zde neuvažujeme, baňku lze pro účely simulace považovat izolovanou soustavu).

Při tepelné výměně dvou těles  $A_A$ ,  $A_B$  lze energetickou bilanci při tepelné výměně vyjádřit pomocí *Kalorimetrické rovnice*[5]:

$$m_A c_A (T_A - T) = m_B c_B (T - T_B) \quad (2.1)$$

kde  $m_i$  je hmotnost tělesa,  $c_i$  měrná tepelná kapacita tělesa,  $T_i$  je původní teplota tělesa a  $T$  je výsledná teplota soustavy.

Obsah baňky můžeme vnímat jako soustavu jednotlivých látek v ní obsažených, proto pro *míšené obsahy* dvou baňek  $B_1$ ,  $B_2$  využijeme 2.1

$$\left( \sum_{i \in B_1} m_i c_i \right) (T_{B1} - T) = \left( \sum_{i \in B_2} m_i c_i \right) (T - T_{B2})$$

kde  $i$  jsou jednotlivé látky v baňce,  $m_i$  jejich hmotnosti a  $c_i$  jejich měrné tepelné kapacity.

Při ohřívání či chlazení baňky dochází k dodání kladného (ohřívání) či záporného (chlazení) tepla soustavě a dochází ke změně teploty soustavy. Změnu teploty lze vyjádřit úpravou 2.1 takto:

$$Q = \left( \sum_{i \in B} m_i c_i \right) (T - T_B) \quad (2.2)$$

kde  $Q$  je teplo soustavě (baňce) dodané.

## 2.3 Změny skupenství

Při zahřívání krystalického pevného tělesa dosáhneme určité teploty, nazývané *bod tání*, kdy nastává spontánně v celém objemu tělesa přechod do kapalného stavu — tento jev se nazývá *tání*. Opačný jev — *tuhnutí* — nastává za stejné teploty. Pro změnu stavu celého tělesa musíme tělesu dodat energii nazývanou *skupenské teplo tání*. Velikost tohoto tepla je závislá na hmotnosti tělesa. Platí [5]

$$L = lm$$

kde  $L$  je skupenské teplo tání,  $l$  je měrné skupenské teplo tání a  $m$  je hmotnost tělesa. Po dobu, kdy těleso přijímá teplo tání, nedochází k růstu teploty.

Stejný případ nastává při změně skupenství mezi kapalným a plynným, ke které dochází při *teplotě varu*.

V případě, že bod tání je roven bodu varu, těleso mění svůj stav přímo mezi pevným a plynným.

## 2.4 Rozpustnost

Roztok je směs dvou a více látek, které jsou navzájem dokonale promíšeny, ale přitom spolu nereagují[8]. V roztoku obvykle rozlišujeme rozpouštědlo a rozpuštěnou látku, přičemž rozpouštědlem nazýváme většinou látku, které je víc. Nejznámějším rozpouštědlem je voda.

Rozpustnost látky je vlastnost, která vyjadřuje, jaké maximální množství látky je možno rozpustit ve vodě (rozpuštědle); pokud se již další množství látky nerozpouští, jedná se o *nasycený roztok*.

Informace o rozpustnosti látek ve vodě za normálních podmínek můžeme získat z chemických tabulek.

## 2.5 Chemické reakce

Chemická reakce je děj, při němž za vhodných vnějších podmínek dochází k přeměně některých látek (reaktantů) na látky jiné (produkty). Jeho podstatou jsou ve skutečnosti srážky molekul reaktantů, po nichž následuje zánik některých vazeb a vytvoření vazeb nových.

Pro objasnění mechanismu přeměny reaktantů na produkty byly postupně vyvinuty dvě teorie [2]:

- **Srážková teorie** předpokládá, že částice při srážce musí mít určitou orientaci a energii, kterou využije k rozštěpení existujících vazeb a poté jsou vytvořeny vazby nové.
- **Teorie aktivovaného komplexu** překonala srážkovou teorii, neboť přesněji popisuje energetické změny při reakci. Vychází z představy, že reagující částice při srážce vytvoří nejdříve tzv. *aktivovaný komplex*. Ten je nestálý a rychle se rozpadá na produkty reakce. Rozdíl mezi energií aktivovaného komplexu a energií výchozích látek označujeme jako *aktivační energii*  $E_A$ .

*Reakční rychlost* definujeme jako přírůstek látkového množství produktů či jako úbytek látkového množství reaktantů. Chemické reakce probíhají různou rychlostí, neboť jsou ovlivňovány vnějšími i vnitřními faktory, které můžeme shrnout takto:

- povahou reagujících látek
- koncentrací reagujících látek
- teplotou
- tlakem
- přítomností katalyzátorů

Tato oblast ještě není zcela probádána, existuje proto mnoho teorií, které popisují vztahy mezi rychlostí reakce a podmínkami, ve kterých probíhá. Většina těchto poznatků je získána pouze *experimentálně* a je použitelná často pouze pro určitou reakci, jejímž průběhem se vědci zabývají často velmi dlouhou dobu. Přesto se pokusíme zobecnit alespoň hlavní faktory u jednodušších reakcí:

### 2.5.1 Vliv koncentrace

Pokud probíhá chemická reakce za konstantní teploty, je její rychlost přímo úměrná součinu koncentrací reaktantů. Závislost rychlosti reakce na koncentraci reaktantů vyjadřuje *kinetická rovnice*[6], např. pro reakci  $A + B \rightarrow \text{produkty}$  má tvar:

$$v = k \cdot [A]^\alpha \cdot [B]^\beta \quad (2.3)$$

kde  $v$  je rychlost reakce,  $[A]$ ,  $[B]$  molární koncentrace látek (počet molů látky na 1000 ml celé soustavy) a  $k$  je rychlostní konstanta reakce, závislá na teplotě.  $\alpha$ ,  $\beta$  určují řád reakce, v jednodušších případech jsou totožné se stechiometrickými koeficienty příslušných látek v chemické reakci (zde 1,1).

### 2.5.2 Vliv teploty

Závislost rychlosti reakce na teplotě vyjadřuje *Arrheinova rovnice*[6]

$$k = A \cdot e^{\frac{-E_A}{RT}} \quad (2.4)$$

kde  $A$  je frekvenční faktor,  $E_A$  je aktivační energie,  $R$  je tzv. univerzální plynová konstanta ( $8.314 \text{ J} \cdot \text{K}^{-1} \cdot \text{mol}^{-1}$ ) a  $T$  je teplota zadaná v kelvinech.

Hodnoty  $E_A$  a  $A$  jsou závislé na reakčním mechanismu jednotlivých reakcí, vyjadřují mimo jiné i pravděpodobnost, že jednotlivé částice budou vhodně natočeny, tedy že jejich srážky budou účinné. Tyto hodnoty jsou v praxi zjišťovány experimentálně.

### 2.5.3 Další vlivy

Přítomnost katalyzátorů ovlivňuje faktory určující reakční rychlost - například zmenšuje *aktivační energii*  $A$ . Tento a podobné jevy ovšem není možné dostatečně vyčíslit, proto jej zde zanedbáme.

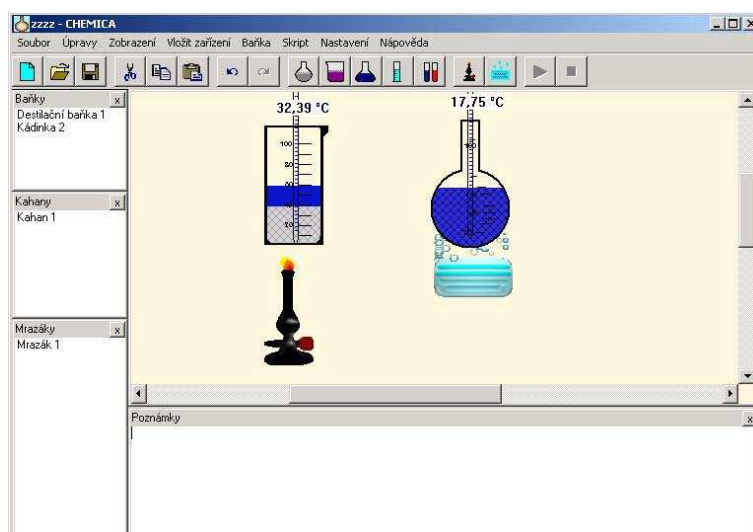
# Kapitola 3

## Implementace

### 3.1 Požadavky na program

#### 3.1.1 Grafické rozhraní

Program je určen k výukovým účelům a měl by tedy obsahovat příjemné a názorné uživatelské rozhraní, ovládání by mělo být dostatečně jednoduché pro běžného uživatele. Fyzické objekty obsažené v laboratoři by tedy měly co nejvěrněji zobrazovat reálný objekt. Ovládání programu by mělo být možné prostřednictvím "klikání".



Obrázek 3.1: Hlavní okno programu Chemica

#### 3.1.2 Implementace chemické a fyzikální teorie

Není logicky možné implementovat všechny reálné vlastnosti, realita je určitým způsobem zjednodušena (např. není brán v potaz tlak, rozpínavost látek v důsledku poměrné absence potřebných vzorců či dat). Fyzikální a chemické chování, které lze

popsat pomocí vzorců, rovnic či algoritmů bude takto (algoritmicky) v programu implementováno. Vlastnosti, chování a veličiny, které není možné obecně popsat, budou zavedeny do programu pomocí databáze. Cílem této práce není shromažďovat chemické údaje o všech chemických látkách či reakcích, je tedy pouze na uživateli, jaká data do programu vloží. Určitá sada dat je dodávána spolu s programem.

### 3.1.3 Řízení pokusů

Důležitou vlastností programu je možnost korigovat práci studenta v průběhu pokusu, což se děje pomocí řídicích skriptů. V první fázi je nutné zadat posloupnost kroků, které budou uživateli postupně zobrazeny. Pro editaci těchto kroků by mělo být vytvořeno dostatečně názorné rozhraní s jednoduchým ovládním.

## 3.2 Problémy a jejich řešení - grafické prostředí

Pro příjemnou práci v programu je nutné vyvinout dostatečně přitažlivé grafické uživatelské rozhraní, jehož implementace se stává jednou z nejdůležitějších částí práce.

### 3.2.1 Základní rozdělení aplikace

V jednom okamžiku může být v rámci jedné aplikace otevřeno pouze jedno prostředí laboratoře.

Většinu plochy hlavního okna zabírá pracovní plocha laboratoře. Pod hlavním menu aplikace je nástrojová lišta usnadňující volání nejpoužívanějších příkazů z menu. Mimo okna pracovní plochy jsou zde další podokna, pomáhající udržovat orientaci mezi jednotlivými objekty v laboratoři.

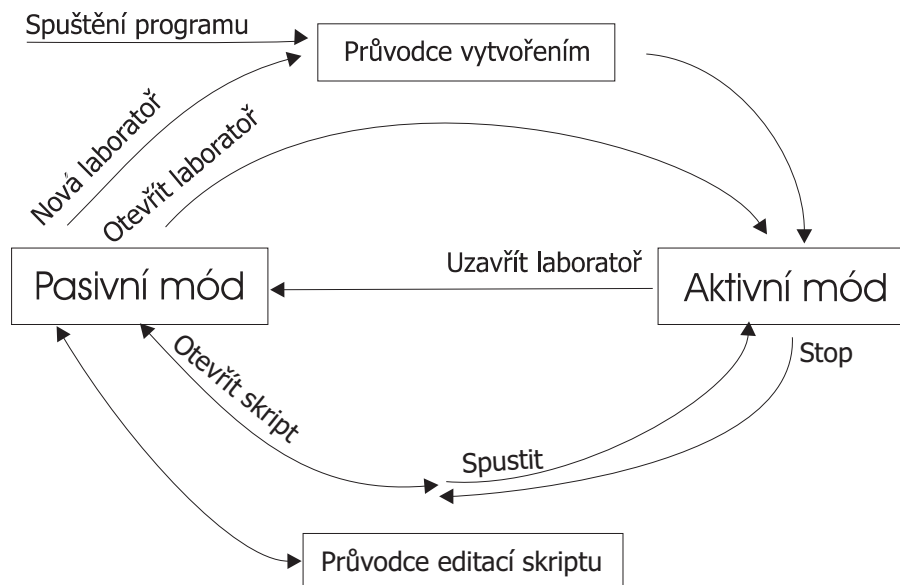
Program prochází za svého chodu mezi stavy *aktivní* a *pasivní*. Pokud je aplikace otevřena a je v aktivním stavu, je možné vytvářet objekty a v pravidelných intervalech dochází ke kontrolnímu výpočetnímu cyklu, který upravuje obsahy jednotlivých baněk v závislosti na jejich obsahu a vzájemném postavení. V pasivním stavu k tomuto nedochází.

Po spuštění se program nachází ve stavu pasivním. Odpovídá tomu skutečnost, že nemáme načtené databáze s informacemi a nemůžeme tedy pracovat v prázdné laboratoři. Do aktivního módu je možné přejít tak, že vytvoříme pomocí průvodce novou laboratoř (pokud je určena defaultní databáze informací, je nová laboratoř vytvořena automaticky), otevřeme uloženou laboratoř či otevřeme a spustíme uložený skript.

Při ukončení aktuální laboratoře je program převeden znovu do stavu pasivního.

### 3.2.2 Vykreslování

Pracovní plochu laboratoře je nutné překreslit ze dvou důvodů - pokud došlo k nějaké akci ze strany uživatele nebo překrytí okna aplikace jiným oknem, či po změně obsahu



Obrázek 3.2: Přechny mezi módy programu

baněk způsobených výpočtem při kontrolním cyklu.

Vzhledem k tomu, že vykreslení baněk a jejich obsahů není jen pouhým vykreslováním grafických primitiv, ale je zde třeba určitého výpočtu, není vhodné vykreslovat vždy celou plochu okna - například při tažení objektu myši by bylo nutné ve velmi krátkých intervalech přepočítávat barvu, výšku a další vlastnosti obsahu a vykreslovat bitmapu obrázku skla všech baněk v laboratoři, což by představovalo příliš vysoké nároky na systém.

Z tohoto důvodu jsou provedeny dvě úpravy základního způsobu vykreslování:

1. v každém okamžiku, kdy je třeba překreslit určitou oblast laboratoře, je překreslován pouze nejmenší obdélník, který tuto oblast obsahuje
2. každý objekt v laboratoři (baňka, kahan, mrazák) si udržuje svou vlastní malou bitmapu a její kontext zařízení. V případě, že má být změněn vzhled samotného objektu — došlo k změně obsahu baňky v důsledku chemické reakce, došlo ke změně skupenství, do baňky byla přidána další látka, došlo ke změně vykreslení plaménku kahanu atd. — objekt vypočte informace nutné ke svému zobrazení a vykreslí svůj obrázek pouze do pomocné bitmapy.

V případě, že později dojde ke zneplatnění určité oblasti okna či ke změně pozice objektu, objekt je požádán o vykreslení. Poté pouze překreslí obrázek ze své bitmapy do hlavní bitmapy na určené místo.

K omezení blikání při vykreslování je použita metoda tzv. dvojího bufferingu[9] — nejdříve jsou obrázky všech objektů vykresleny do pomocné bitmapy, která je v paměti, a až nakonec je celý obsah bitmapy v paměti zkopírován do bitmapy v okně.

Pokud baňka obsahuje nějaké látky, je třeba je graficky znázornit. Pomocí informací v databázi vypočítáme objem obsahu baňky. Každá baňka má zadán svůj maximální objem a své rozměry (poloměr nejširšího místa průřezu, výška). Z těchto údajů vypočteme, jakou výšku má hladina obsahu, v závislosti na tvaru baňky (koule, válec, kužel). Barvu obsahu získáme váženým průměrem barev jednotlivých látek v závislosti na jejich jednotlivých objemech.

V okamžiku, kdy uživatel ohřívá baňku a dospěje do bodu změny skupenství některé z látek v baňce, přestane dočasně narůstat teplota v baňce, jak jsme si popsali v kapitole 2. Při testování programu bylo zjištěno, že neinformovaný uživatel nechápe, proč se teplota "zastavila", i když z dialogového okna popisujícího obsah baňky je patrné, že dochází k přeměně skupenství. Bylo proto nutné tuto situaci nějak graficky znázornit. Při vykreslování obsahu baňky tedy není nevykreslujeme celý obsah baňky najednou, ale pomyslně rozdělíme její obsah na tři části, jež vykreslíme zvlášť: ke "dnu" baňky vykreslíme objem látek, které se nalézají v pevném skupenství, nad ně vykreslíme látky v kapalném skupenství a látky rozpuštěné ve vodě a nejvýše vykreslíme všechny plyny. Vertikální pořadí skupenství jsme zvolili z obecné představy, že pevné látky bývají obecně nejtěžší, proto klesly ke dnu, plynné látky bývají nejlehčí a kapaliny je tudíž nadnáší. Tato představa samozřejmě nevystihuje přesně všechny látky v přírodě, ovšem pro vytvoření představy o situaci uvnitř baňky je vhodná.

Každému ze tří skupenství vypočítáme jeho barvu. Kapalnou část vykreslíme způsobem, jaký jsme popsali v původním algoritmu. Pevnou část vykreslíme se vzorkováním, jehož barva je tmavší než barva pevných látek (zvýrazníme tak např. změnu skupenství v případě, že v baňce je jen jeden druh látky - obě části by byly vykresleny stejnou barvou). Plynnou část odlišíme tak, že vykreslíme pouze náhodně zvolených 50 procent vykreslovaných pixelů - znázorníme tak "mihotání" jednotlivých částic.

Pokud je baňka nakloněna, předpokládáme, že obsah se promíchal a obsah vykreslujeme pouze základním způsobem.

Při přelévání obsahu jedné baňky do druhé je vidět změnu výšky sloupce obsahu u obou baňek. Pro lepší grafické znázornění přetékání vykreslujeme proud "kapiček" stékající dolů mezi horní a spodní baňkou.

Při vykreslování hořáku simulujeme třepotání plaménku tak, že jsme jej rozdělili na tři části, z nichž každá může být zobrazena čtyřmi způsoby nezávisle na sobě — celkem tak při náhodném výběru vykreslované varianty získáváme dvanáct různých zobrazení plaménku.

Při vykreslování "mrazáku" jsme využili fantazie. Na jeho povrchu tedy náhodně vytváříme "bublínky", které postupně stoupají vzhůru a můžeme si představit, že při dotyku s jiným tělesem ho ochlazují.

### 3.2.3 Posuvníky

Plocha, ve které může uživatel umísťovat své objekty, není programem nijak omezena, má pouze své dočasné omezení, které je možné libovolně zvětšovat. Výhled na všechny objekty je omezen pouze výřezem okna, kde je pracovní plocha zobrazena. Pro pohyb po ploše jsou slouženy dva posuvníky. Při spuštění nové laboratoře má pra-

covní plocha zadána určitou počáteční velikost. Tu je možné zvětšit tak, že některý z objektů umístíme na okraj dosavadní plochy, ta automaticky upraví svoje okraje.

Při vytváření posuvníků bylo nutné dohlédnout na to, aby velikost jejich jezdců stále odpovídala poměru velikosti výřezu okna a velikosti plochy a aby nebylo možné vyjet pomocí posuvníku mimo aktuálně omezenou plochu či některý z objektů laboratoře nebyl nedostupný.

Při vykreslování laboratoře není nutné udržovat si v paměti velkou bitmapu, do které by se vykreslovala celá plocha. Vykreslujeme pouze oblast určenou velikostí výřezu okna a pozice obou posuvníků.

### 3.2.4 Podokna

Uživatelská plocha hlavního okna aplikace je rozdělena na několik podoken. Uživatel si může nastavit rozměry jednotlivých oken či některá z nich skrýt, pro zachování "pořádku" a příjemnějšího vzhledu ale změna pozice jednoho podokna ovlivní pozici ostatních.

Při změně rozměru jednoho z oken je volána funkce, která poopraví rozměry ostatních, v závislosti na tom, zda jsou viditelná. Při změně velikosti celého okna mají podokna vlevo a nahoře "přednost" před ostatními okny — jejich rozměry jsou ponechány a upravena je velikost oken více napravo a výše.

### 3.2.5 Operace Zpět/Vpřed

Vzhledem k tomu, že obsah baňky může vlivem chemické reakce podléhat určitým změnám, nebylo by vhodné operace Zpět a Vpřed implementovat, v současnosti je to ovšem u grafických aplikací samozřejmost, je zde možnost vracet alespoň editační akce, které neberou ohled na obsah baňky:

1. vložení objektu
2. posun objektu
3. smazání objektu
4. vyprázdnění baňky
5. změření teploty
6. překlopení baňky

Implementace provádění operací Zpět a Vpřed byla vytvořena takováto: V programu jsou dva zásobníky, zastupující paměť pro operace Zpět a Vpřed. Pokud dojde k některé z těchto editačních akcí, je vytvořena instance třídy *pamet\_akce*, která obsahuje informaci o typu akce, objektu, kterého se akce týkala, a dalších podrobností. Tato instance je umístěna na vrchol zásobníku Zpět. Při stisknutí tlačítka *Zpět* je ze zásobníku vyjmuta posledně vložená instance, podle jejích informací jsou provedeny příslušné změny a instance je vložena na vrchol zásobníku Vpřed. Při stisknutí tlačítka *Vpřed* je obdobně vrchol zásobníku Vpřed přemístěn na zásobník Zpět.



Program rozezná, zda se jedná o vrácení či opakování akce, a podle toho se zachová — není třeba upravovat informaci o akci podle toho, do jakého zásobníku má být vložena.

Obsah zásobníku Zpět není nijak omezen, v zásobníku Vpřed je vždy maximálně tolik položek, kolik jich bylo přemístěno ze zásobníku Zpět. V okamžiku, kdy dojde ze strany uživatele k nějaké akci a do zásobníku Zpět je vložena nová informace, zásobník Vpřed je vyprázdňen.

### 3.2.6 Kopírování a mazání objektů

Při kopírování označených objektů jsou kopie jejich instancí vloženy do speciálního kontejneru.

Při mazání objektu není z důvodu kompatibility akce Zpět jeho objekt smazán z paměti programu — pouze je uveden do stavu, kdy není vykreslován na plochu a neprobíhají v něm žádné změny. Všechny instance objektů, které byly do laboratoře vloženy od chvíle vytvoření či otevření této laboratoře, jsou udržovány v paměti programu a jsou vymazány až při ukončení aktuální laboratoře.

## 3.3 Problémy a jejich řešení - výpočetní model

Program je v podstatě real-time aplikací simulující prostředí laboratoře. V krátkém intervalu je opakovaně upravován její obsah v závislosti na implementovaných pravidlech. Program byl napsán v jazyce C++ s využitím Win32API funkcí.

### 3.3.1 Základní datové struktury

Logickým zástupcem programu je instance třídy *ProgramEngine*, která obsahuje všechny položky a metody nutné pro chod programu. Jejími nejvýznamnějšími třídami jsou kontejnery *m\_datalatka* obsahující databázi informací o chemických látkách, *m\_datareakce* obsahující databázi informací o chemických reakcích a *m\_objekt*, která obsahuje seznam všech fyzických objektů (baněk, kahanů, mrazáků) v laboratoři.

*M\_datalatka* je asociativní pole obsahující instance třídy *datalatka*, logicky zastupující jednotlivé chemické látky, jejímiž položkami jsou všechny vlastnosti chemické látky popsané v 4.5.1, klíče tvoří jejich hodnoty CAS-RN. *M\_datareakce* je asociativní pole s možností duplicitních klíčů, obsahem jsou zde instance třídy *datareakce* zastupující jednotlivé informace o reakci (především seznamy CAS-RN reaktantů a produktů a údaje pro výpočet rychlosti). Klíče zde tvoří speciálně utvořené hodnoty třísloužkového typu CAS-RN, vypočítané jako součty jednotlivých složek CAS-RN reaktantů reakce. Aby bylo možné rychle vyhledat všechny reakce, kde známe reaktanty, ale neznáme jejich jednotlivé počty, započítáváme hodnotu CAS-RN pouze jednou u každého reaktantu, nezávisle na jeho stechiometrickém koeficientu (např. reakce  $A+B \longrightarrow \dots$  bude tedy zařazena se stejným klíčem jako reakce  $2A+B \longrightarrow \dots$ ).

Všechny fyzické objekty jsou v programu sdruženy v seznamu *m\_objekt*. Ten obsahuje ukazatele na instance třídy *objekt*, jíž jsou všechna tělesa v laboratoři potomky.

Takto bylo zvoleno z hlediska jednoduššího sériového přístupu ke všem objektům v laboratoři a možnosti použít virtuální funkce odlišné podle typu objektu, především pro vykreslení objektu. Od třídy *objekt* jsou odvozeny třídy *bankakulata* zastupující v laboratoři baňku, *horak* zastupující kahan a třída *mrazak*, která zastupuje mrazák. Třída *objekt* nese informaci o pozici a rozměru fyzického objektu, jeho jméno a viditelnost a metody pro manipulaci s těmito členskými proměnnými.

Třída *horak* a *mrazak* obsahují navíc proměnné a metody potřebné k jejich vykreslení. Tyto dvě třídy jsou si velmi podobné, vzhledem k jinému způsobu vykreslování a s tím spojenému rozdílu mezi pomocnými daty bylo nutné vytvořit třídy samostatné.

Třída *bankakulata* je odvozena od třídy *objekt* a od třídy *bankaVelka*, která obsahuje data a metody určené k chemickému a fyzikálnímu výpočtu. Tyto vlastnosti jsou přiděleny třídě *bankakulata* prostřednictvím dědění z logického důvodu proto, že chemické a fyzikální výpočty jsou implementovány bez závislosti na tom, o jaký typ baňky se jedná či kde a jakým způsobem bude vykreslena.

Třída *bankaVelka* obsahuje údaj o teplotě a seznam látek, které obsahuje. Nej důležitějšími jsou metody pro vkládání látky do baňky, pro změnu po dodání tepla baňce a na výpočty ohledně chemických reakcí.

Třída *bankakulata* přidává další informace, které popisují baňku jako fyzický objekt, například údaje o objemu baňky a jejím druhu viz 4.1 a implementuje metody používané především při vykreslení baňky a při manipulaci s baňkou jako s fyzickým objektem.

Při vložení nového objektu do laboratoře je ukazatel na něj připojen do seznamu *m\_objekt*; při vymazání objektu uživatelem je pouze nastavena hodnota viditelnosti objektu na *FALSE*; v tom případě nedochází k vykreslování objektu, ani k žádným jeho vnitřním výpočtům.

### 3.3.2 Datové typy

V programu byla vytvořena třída *cislolong* jako speciální číselný typ převážně ze dvou důvodů:

1. u některých údajů v laboratoři lze předpokládat, že může nastat situace, kdy maximální hodnota či délka standardních číselných typů nebude dostačující (např. hmotnost některé látky v baňce bude tak vysoká, že v použitém standardním typu nezbude místo na zadání desetinných cifer až po přesnost, kterou požadujeme) — bylo by tedy vhodné vytvořit typ s pružnou délkou záznamu
2. mnoho chemických či fyzikálních dat zadávaných do databáze je udáváno ve formátu  $x10^y$ , jehož syntaxi je vhodné v dialogích podpořit. Tento údaj by mohl být uvnitř programu překládán na typ *double*, ovšem vzhledem k (i když nepatrné) nepřesnosti při jeho manipulaci může již při zpětném výpisu této hodnoty dojít k zobrazení této chyby uživateli

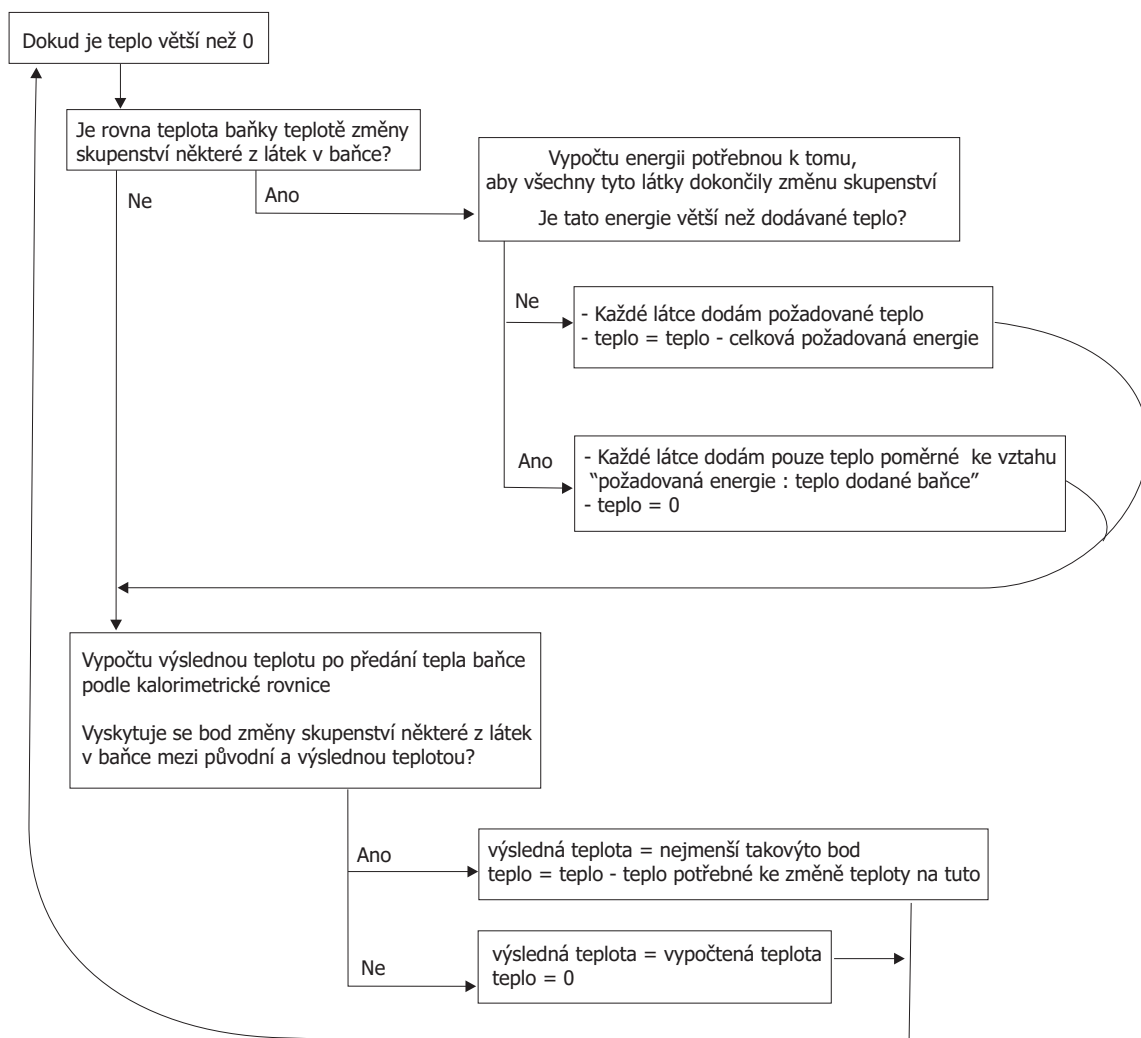
Z tohoto důvodu byla zavedena třída *cislolong* a její metody jako řídký seznam dvojic tvaru  $bas10^{exp}$ , která je využita ve velké části číselných údajů v laboratoři.

Třída *cislolong* implementuje aritmetické operace počítáním s položkami v dekadickém tvaru — poupravěním této třídy na binární formát by bylo možné urychlit výpočty.

### 3.3.3 Změny teploty

Každá látka v baňce má zadáno, jaké je její množství v gramech. Abychom si mohli pamatovat, jaká část látky je již např. v kapalném stavu při tání látky, udržujeme si v paměti také hmotnost látky, která již podlešla změně skupenství.

Při ohřívání či ochlazování baňky je volána metoda třídy *bankaVelka Ohrev (Ochlazeni)*, která počítá změnu teploty baňky. Jako parametr získává velikost tepla, které je baňce předáno v závislosti na výkonu kahanu a jeho vzdálenosti od baňky. Při změně teploty vycházíme ze vzorce 2.2. Musíme dát ale pozor na změny skupenství látek, proto počítáme podle následujícího algoritmu:



Obrázek 3.3: Algoritmus změny teploty

Pokud vkládáme do baňky látky o jiné teplotě, vložíme látky do baňky a poté odebereme celému obsahu teplo rovné teplu, které by bylo třeba dodat přidávané látce, aby byla její výsledná teplota rovna původní teplotě baňky.

### 3.3.4 Vyhledávání reakcí

Při vytváření seznamu všech reakcí, které v baňce probíhají, postupujeme tak, že pro  $n$  rovno jedné až po počet látek v baňce vybíráme všechny  $n$ -tice těchto látek, vypočítáme speciální CAS-RN hodnotu rovnou součtu CAS-RN vybraných látek a vyhledáváme reakce mající tyto reaktanty v asociativním poli *m\_datareakce* na klíči rovném vypočtenému kódu.

### 3.3.5 Výpočet rychlosti reakcí

Podle 2.5 vypočítáme změny koncentrace u jednotlivých reakcí, které probíhají v baňce. Pokud by ale jedna látka byla reaktantem více reakcí současně, mohlo by dojít ke kolizi - reakce by "požadovaly" celkem více látky, než je v baňce k dispozici. Proto postupně procházíme všechny probíhající reakce a u každého jejich reaktanta si poznamenejeme, kolik látky tato reakce odebere. Poté procházíme jednotlivé látky v baňce a porovnáváme součet požadovaného množství a skutečné množství látky. Pokud je požadováno příliš mnoho látky, poznamenejeme si k ní poměr těchto údajů, jinak jí učíme poměr rovný 1. Poté procházíme znovu postupně všechny reakce a vypočteme číslo rovné minimu poměrů u jednotlivých reaktantů. Pokud je toto číslo menší než 1, vynásobíme změnu koncentrace pro tuto reakci tímto číslem — reakci bylo nutné omezit z důvodu konkurence. Nyní pro každou reakci odebereme odpovídající vypočtené množství reaktantů a přidáme odpovídající množství produktů.

### 3.3.6 Editace skriptu

Řídící skript v laboratoři zastupuje instance třídy *skript*. Ta obsahuje seznam jednotlivých řídicích kroků — instancí třídy *skript\_akce* a metody pro obsluhu skriptu.

Každá instance třídy *skript\_akce* obsahuje mimo jiné údaj výčtového typu udávající typ této akce, ukazatel *obj* na třídu *objekt* a ukazatel *objRef* na třídu *skript\_akce*. Je třeba umožnit, aby řídicí krok odkazoval na objekt vytvořený v některém předchozím kroku. (Např. při zadávání kroku *Vymazat objekt*). Rozdělíme tedy chování při editaci kroku podle jeho typu:

1. Pokud se jedná o vložení některého objektu do laboratoře, tento objekt v laboratoři při editaci skriptu také vytvoříme (to je možné, neboť program se nachází v pasivním módu a na plochu není nic vykreslováno). Ukazatel *obj* umístíme na nově vytvořený objekt.
2. V ostatních případech při potřebě odkazovat na některý z objektů laboratoře (např. který objekt má být smazán) umístíme ukazatel *objRef* na instanci typu *skript\_akce*, která objekt vytvořila.

Při vytvoření nového objektu nastavujeme jeho viditelnost na hodnotu *TRUE*. Pokud uživatel přechází mezi jednotlivými kroky skriptu, při přechodu přes tento krok směrem doleva (proti času) je viditelnost objektu nastavena na hodnotu *FALSE* (což má vliv na vykreslení jmen objektů v seznamu objektů na dialogu), při přechodu doprava opět na hodnotu *TRUE* - tedy tak, jak bude tento objekt v určité chvíli viditelný při spuštění skriptu. Pokud se jedná o krok typu *Smazat objekt*, je logicky viditelnost nastavována obráceně.

Pokud tedy uživatel v seznamu objektů vyznačuje již vložený objekt, index této položky představuje zároveň pořadí instance *skript\_akce* v posloupnosti, kde ukazatel *obj* ukazuje na v tuto chvíli viditelnou instanci třídy *objekt*.

Skript ukládáme do souboru XML. Referenci na objekt (mimo kroků vytváření objektů, kde referenci neukládáme) uložíme tak, že zapíšeme pořadí tohoto objektu v okamžiku aktivace této akce.

Při následném načítání souboru do paměti nejdříve načítáme pouze čísla určující pořadí vytvořených objektů v laboratoři, poté simulujeme průchod posloupností kroků jako v případě předchozí editace skriptu a ukazatele *objRef* správně nastavíme.

Při mazání kroku uživatelem je nutné nejdříve zkontrolovat, zda mazaný objekt není v některém z následujících kroků referován.

### 3.3.7 Komunikace programu a skriptu

Při spuštění řídicího skriptu lze řídicí kroky rozdělit na dvě skupiny podle toho, zda je jejich splnění podmíněno nějakou akcí uživatele (například vložení kahanu), či stavem referovaného objektu (například ohřev baňky na určitou teplotu). Běh programu je poupraven tak, aby po každém činu uživatele, který mohl teoreticky zapříčinit splnění kroku potenciálně spuštěného skriptu, byla volána jednotná funkce *skriptRefer*, ve které je zadáno pomocí výčtového typu o jaký čin šlo a kterého objektu se čin týkal. Pokud je skript spuštěn, sám se poté postará o zjištění, zda byl aktuální krok splněn.

Při každém přepočítání obsahu laboratoře ve funkci *Cyklus* je zavolána funkce skriptu, ve které řídicí skript zjistí, zda je splněn krok, jehož typ patří do druhé skupiny akcí (tedy kde záleží na stavu referovaného objektu).

# Kapitola 4

## Uživatelská dokumentace

### 4.1 Instalace programu

Program CHEMICA je napsán pro operační systém Windows. Instalaci zajišťuje instalační soubor CHEMICAsetup-1.0.exe, který se postará o vše potřebné. Uživatel si vybere místo instalace, je možné vytvořit položku v nabídce Start či ikonu na ploše.

### 4.2 Spuštění programu

Program se spouští souborem CHEMICA.exe, který se nachází v hlavním adresáři. Po spuštění programu se program pokusí vytvořit novou laboratoř, v rámci čehož je nutné zadat databáze chemických informací. Program je koncipován tak, že je možné nejen přidávat, editovat a mazat položky databáze, ale je možné také udržovat si různé verze těchto databází, rozdělené do jednotlivých souborů například tak, že každý soubor obsahuje informace o určité skupině chemických látek, a při vytváření laboratoře tyto soubory libovolně slévat. Protože tento postup (tedy zadávat tyto údaje znovu a znovu při každém vytvoření nové laboratoře) by byl zbytečně zdoluhavý, je zde možnost určit si, která databáze bude defaultně načtena.

Pokud tedy novou laboratoř vytváříme výběrem položky menu *Soubor / Vytvořit novou laboratoř. . .*, zobrazí se průvodce vytvořením nové laboratoře, ve kterém nejen určíme, které databáze budou použity, ale máme zde také možnost databáze editovat či určit, která databáze bude vybrána jako defaultní.

Pokud novou laboratoř vytváříme výběrem položky menu *Soubor / Nový . . .*, je otevřena automaticky defaultní laboratoř, která je (při jejím určení viz předchozí případ) uložena v souboru *\_default.xml*, *\_default.xmlr*. Pokud je některý z těchto souborů smazán, zobrazí se průvodce vytvořením jako v předchozím případě.

Uloženou laboratoř otevřeme výběrem z menu *Soubor* položka *Otevřít laboratoř*.

V podadresáři *Ukladani* můžeme nalézt některé uložené pokusy, v podadresáři *Skripty* spustitelné scénáře. Jejich načtení je možné buď výběrem z nabídky menu, či přímo poklepnutím na určený soubor ve file manageru.

## 4.3 Ovládání chemické laboratoře

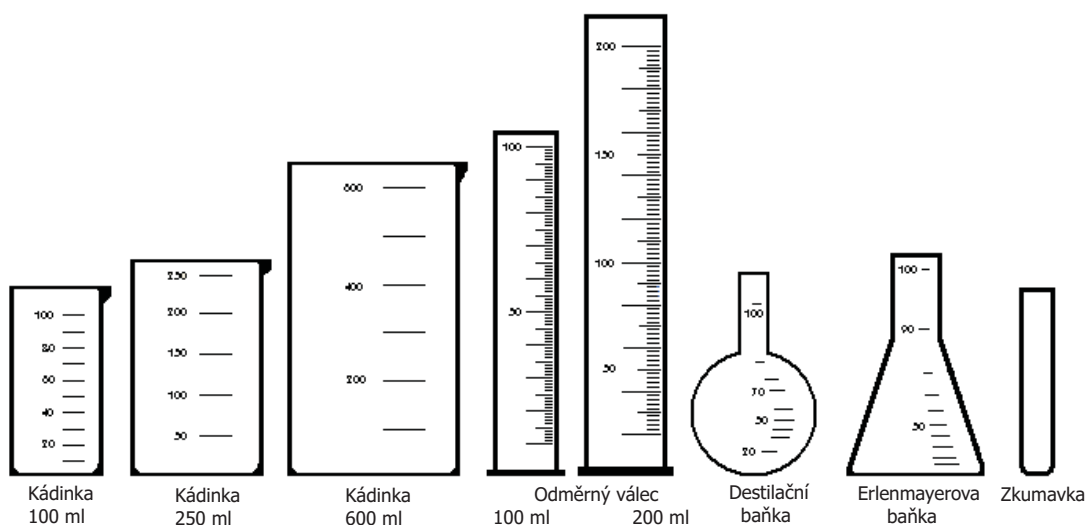
### 4.3.1 Objekty v laboratoři

Do pracovní plochy okna, zobrazující chemickou laboratoř, můžeme přidávat různé druhy objektů:

- **baňky** slouží jako nádoby k uchovávání chemických látek
- **kahany, mrazáky** mohou měnit teplotu v baňce

Vkládat objekty do laboratoře můžeme pomocí položky menu *Vložit zařízení* či kliknutí pravým tlačítkem na pracovní plochu a výběrem z kontextového menu. V prvním případě je nový objekt vložen doprostřed okna, ve druhém je vložen na místo kliku myši.

V levé části hlavního okna je vyobrazen seznam objektů, které laboratoř v tuto chvíli obsahuje. Pro snazší orientaci je každému objektu přiřazeno jméno. Při vytvoření je každému objektu přiřazeno defaultní jméno, odvozené od druhu tohoto objektu, tedy např. "Destilační baňka" či "Kahan". Duplicitě defaultních jmen vkládaných objektů je zamezeno přidáním "pořadového" čísla tohoto typu objektu — např. "Destilační baňka 2". Změnit jméno objektu je možné kliknutím pravým tlačítkem myši na jeho vyobrazení na pracovní ploše a výběrem z kontextového menu položky *Vlastnosti*. Uživatel má právo zadávat duplicitní jména objektů.



Obrázek 4.1: Souhrn různých typů baněk, které je možno vložit do laboratoře

### Označování objektů

Pozici objektů v pracovní ploše laboratoře můžeme libovolně měnit. Posouvat můžeme ovšem pouze označené objekty. Jeden objekt označíme kliknutím levým tlačítkem na

jeho vyobrazení na ploše laboratoře. Kolem objektu se vykreslí přerušovaný obdélník. Pokud chceme označit více objektů, podržíme klávesu *Shift* a postupujeme stejně jako v předchozím případě. Jeden z označených objektů můžeme odznačit tak, že podržíme klávesu *Shift* a klikneme na něj. Objekty odznačíme tak, že klikneme na pozadí laboratoře (mimo všechny objekty).

Pokud chceme označit více objektů ležících v sousední oblasti, přemístíme kurzor myši mimo označovanou oblast. Stiskneme levé tlačítko myši a táhneme přes celou oblast. Vykreslí se přerušovaný obdélník, jehož jeden roh je v místě stisknutí tlačítka a pozice protilehlého rohu je shodná s aktuální pozicí kurzoru. Přetáhneme myš tak, aby obdélník obsahoval všechny označované objekty. Po uvolnění levého tlačítka myši budou označeny všechny objekty, které ležely celé uvnitř obdélníka.

Přidat k těmto objektům další oblast můžeme opakováním předchozího úkonu při podržení tlačítka *Shift*. Všechny objekty je možné označit pomocí kombinace kláves *Ctrl-A*.

### 4.3.2 Vkládání chemických látek do baňky

Do baňky je možné vkládat látky tak, že ji označíme levým tlačítkem myši a vybereme položku menu *Baňka / Vložit chemikálie*, nebo při kliknutí pravým tlačítkem výběrem položky *Vložit chemikálie* z menu kontextového.

V seznamu látek v zobrazeném dialogovém okně vybereme přidávanou látku a zadáme požadované množství. Pokud se jedná o látku, která se bude po přidání do laboratoře (v závislosti na standardní teplotě laboratoře) nacházet v pevném skupenství, můžeme si vybrat, zda zadaná hodnota určuje hmotnost v gramech či miligramech; pokud se jedná o látku kapalnou či plynnou, zadáváme objem látky v mililitrech.

V okamžiku přidání do některé baňky v laboratoři bude mít vkládaná látka teplotu rovnou základní teplotě laboratoře (standardně 20°C). Pokud baňka již dříve nebyla prázdná, dojde okamžitě k vyrovnání teplot mezi přidávanou látkou a soustavou stávajících látek podle pravidel nastíněných v kapitole 2.

#### Obsah baněk

Informaci o aktuálním obsahu baňky získáme jejím označením a výběrem z menu *Baňka* položka *Vložit chemikálie*. V horní části dialogového okna je vypsán výčet všech látek v baňce a jejich hmotností. U každé položky seznamu je také uvedeno, v jakém skupenství se látka právě nachází. Pokud právě dochází ke změně skupenství látky, každé skupenství přechodu je vyobrazeno na samostatné řádce.

Pro průběžné zobrazení teploty baňky slouží teploměr. U jedné baňky ho zobrazíme výběrem položky *Měřit teplotu* v kontextovém menu baňky; výběrem v hlavním menu položky *Baňka / Měřit teplotu* zobrazíme teploměr u všech označených baněk.

Opětovným výběrem teploměr skryjeme.



## Změny teploty

Všechny látky jsou do baněk přidávány o stejné teplotě, která je rovna standardní teplotě laboratoře. Zvýšit či snížit teplotu v baňce lze pomocí kahanu či mrazáku tak, že jej umístíme pod baňku.

Kahan předává baňce své teplo a postupně zvyšuje její teplotu, mrazák teplo odebírá. Předpokládejme, že jsme pod baňku umístili kahan. Teplo, které je předáváno, závisí na vzdálenosti kahanu (mrazáku) od baňky. Pokud je hrdlo kahanu těsně pod dnem baňky, je každou vteřinu předáváno teplo rovné teplu, které vydá kahan (výkon kahanu i mrazáku je 100 wattů, tedy 100 joule za sekundu). Velikost předávaného tepla klesá nepřímo úměrně se vzdáleností hrdla kahanu ode dna baňky. Vzdálenost, při které již nedochází k předání tepla, odpovídá přibližně 2,5 násobku výšky plamenu.

”Mrazák” si lze představit jako plošinku, nad kterou je při ochlazování nutné baňku umístit. Platí zde stejné pokyny jako v případě ohřívání teploty kahanem.

## Přelévání

Mísit obsahy jednotlivých nádob můžeme tak, že obsah jedné baňky přelijeme do druhé.

Baňku, jejíž obsah chceme přelít, označíme a nakloníme pomocí výběru položky *Převrátit baňku*. Obsah nakloněné baňky přetéká do baňky, která se nachází pod touto a nakloněná není, rychlostí 1/10 objemu baňky za sekundu. Přesun hmoty znázorňují kapky, které stékají směrem dolů mezi oběma baňkami.

Do původní polohy vrátíme baňku opětovným výběrem *Převrátit baňku*.

## 4.4 Editace řídicích scénářů

Významnou vlastností programu je možnost definovat posloupnosti řídicích kroků, které postupně předepisují uživateli jeho další činnost. Skripty jsou ukládány ve formě XML souborů, syntaxi můžeme nalézt v Příloze D.

Pro usnadnění tvorby těchto scénářů byl vytvořen průvodce vytvořením a editací řídicího skriptu, kterého spustíme volbou položky menu *Skript / Vytvořit nový*. Výhodou použití průvodce není jen urychlení editace skriptu, ale i okamžitá kontrola korektnosti kroků (např. není možné měřit teplotu mrazáku).

V prvním kroku je možné zadat autora a název skriptu, v pravé části dialogu je oblast pro poznámky, které se budou zobrazovat po celou dobu pokusu. Zde je možné popsat téma, kterého se pokus týká, a shrnout jednotlivé kroky, které bude student později provádět. Do editačního pole ve spodní části dialogu zadáme zprávu, která bude zobrazena *studentovi*, až dokončí celý pokus.

V dalším kroku je zobrazen seznam chemických látek a reakcí, které budou při pokusu k dispozici. Po stisknutí jednoho ze dvou tlačítek *Upravit* se zobrazí dialog editující odpovídající databázi 4.3.

Nyní již můžeme zadat posloupnost kroků, které je třeba během pokusu postupně provést. Každý krok je editován samostatně na vlastním listu:

V levém horním rohu lze vybrat typ kroku, který *učitel* edituje. Po levé straně je zobrazen seznam baněk, kahanů a mrazáků, které již byly na základě požadavku skriptu vloženy do laboratoře před tímto krokem. Ve spodní části obrazovky je oblast, do které zapíšeme, jaký komentář bude zobrazen, pokud při pokusu dospěje *student* k tomuto kroku. Text v komentáři je z hlediska následné funkčnosti zcela nezávislý na tom, jaké vlastnosti tohoto kroku zadáme, měl by ovšem slovně tento krok popisovat, či alespoň svým obsahem vést *studenta* k dedukci, jaký čin povede ke splnění tohoto kroku, aby *student* zbytečně netápal.

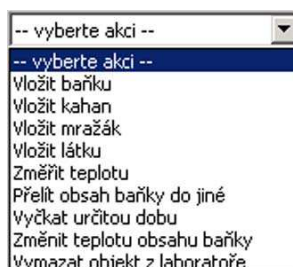
V pravém dolním rohu je skupina tlačítek korigující průchod mezi jednotlivými kroky při editaci:

- tlačítka **Předchozí krok** a **Další krok** posunují na vedlejší kroky
- tlačítko **Vložit krok** vloží nový krok *před* krok, na kterém právě stojíme
- tlačítko **Zrušit krok** vymaže z posloupnosti krok, na kterém právě stojíme, a posune nás na krok následující
- na konci posloupnosti nemusíme další kroky přidávat — při stisknutí tlačítka *Další krok* je automaticky vložen nový prázdný krok

Zbýlý obsah dialogového okna bude využit pro zobrazení dodatečných informací.

#### 4.4.1 Řídící krok

Prvním úkolem při zadávání nového kroku skriptu je určit, o jaký typ kroku se bude jednat:



Obrázek 4.2: Různé typy řídicích kroků

##### Vložit baňku, Vložit kahan, Vložit mrazák

Pokud se jedná o baňku, vybereme, který typ baňky vložíme. Další informace zde nejsou potřeba. Po přechodu na další krok bude do seznamu objektů v levé části okna přidáno symbolické jméno tohoto objektu. Při vytváření a editaci řídicích skriptů není možné měnit symbolická jména objektů.

## Vložit látku

V tomto kroku požadujeme explicitní přidání látky do baňky pomocí dialogu. V seznamu objektů vybereme baňku, do které má být látka přidána. V seznamu chemických látek vybereme, která látka má být přidána do této baňky. Požadované množství přidané látky lze vyjádřit dvěma způsoby:

- určení pomocí rozmezí: tento krok bude uznán, pokud se hmotnost přidávané látky bude vyskytovat mezi zadaným minimem a maximem
- přesné určení: krok bude uznán za splněný, pokud odchylka přidané hmotnosti nebude od zadané hodnoty větší než zadaná tolerance. Pokud je tolerance rovna nule, požadujeme zadat přesné množství

Pokud *student* vloží do baňky vícekrát menší množství látky, ale jejich součet vyhovuje požadovanému rozsahu, krok bude splněn. Pokud *student* vloží větší množství než bylo požadováno, má možnost *Vyprázdnit baňku* — součet bude poté počítán znovu od nuly.

## Změřit teplotu

V seznamu objektů vybereme baňku, u které má být zobrazen teploměr. Pokud již bude v průběhu pokusu u této baňky teploměr zobrazen v době, kdy přejdeme na tento krok, podmínka postupu bude automaticky splněna

## Přelít obsah baňky do jiné

V tomto kroku jsou zobrazeny dva seznamy objektů v laboratoři. Označíme dvě baňky v nich tak, že bude požadováno přelití obsahu baňky na levé straně do baňky na pravé straně.

## Vyčkat určitou dobu

Při plnění tohoto kroku není *student* nucen vykonat žádnou činnost. V dialogu vyplníme čas v sekundách, za jakou dobu bude tato akce automaticky splněna. Tento krok můžeme využít při čekání, než dostatečně zreaguje obsah nějaké baňky, či k zobrazení informativního textu bez nutnosti splnění nějakého kroku.

Pokud chceme zobrazit čas, který zbývá do konce odpočítávání, napíšeme do doprovodné poznámky řetězec "*xxx*", tedy např. "Čekejte ještě *xxx* sekund.". Na místo řetězce bude průběžně vypisován zbývající čas ve vteřinách.

## Změnit teplotu obsahu baňky

V seznamu objektů vybereme baňku, u které má dojít ke změně teploty. Požadovanou změnu teploty baňky můžeme opět vyjádřit více způsoby:

- relativně: zadaná hodnota značí, o kolik stupňů Celsia nejméně je žádáno zvýšit či snížit původní teplotu.

- pomocí rozmezí: krok bude uznán za splněný v okamžiku, kdy teplota v baňce bude ležet uvnitř vymezeného intervalu [*minimum*, *maximum*]
- přesné určení: krok bude uznán za splněný v okamžiku, kdy odchylka výsledné teploty nebude od zadané hodnoty větší než zadaná tolerance.

### Vymazat objekt z laboratoře

V seznamu objektů vybereme objekt, který má být z laboratoře odstraněn. Při editaci následujících kroků skriptu již tento objekt nebude zobrazen v seznamu přidávaných objektů.

Při editaci skriptu je možné volně přecházet mezi jednotlivými kroky, v okamžiku opuštění kroku bude tento uložen. Pokud opustíme krok, u kterého není zadán typ požadované akce, krok bude automaticky smazán.

## 4.4.2 Spuštění skriptu

Spustitelný skript načteme výběrem položky menu *Skript / Otevřít . . .*. Ve spodní části obrazovky je zobrazen průvodní popis, který budeme moci pročítat po celou dobu pokusu. Skript spustíme příkazem *Skript / Spust.* Laboratoř je možné ovládat stejným způsobem jako při základní práci v laboratoři. V pravé části hlavního okna programu je zobrazeno okénko, obsahující poznámku ke kroku, který je v tomto okamžiku nutné vykonat. Při splnění podmínky je zobrazena poznámka ke kroku následujícímu.

Skript je možné libovolně spouštět (*Spust.*) či ukončovat (*Stop*) bez nutnosti jeho opětovného načtení. Po ukončení a následném znovuspuštění probíhá posloupnost řídicích kroků znovu od začátku. Při pokusu řízeném skriptem není možné provádět akci *Zpět* a *Vpřed*.

## 4.5 Editace chemické databáze

Databáze chemických údajů můžeme editovat pomocí průvodce vytvořením chemické laboratoře, kterého spustíme výběrem položky menu *Soubor / Vytvořit novou laboratoř*. V prvním kroku je třeba načíst či vytvořit databázi chemických látek, až ve druhém kroku databázi chemických reakcí. Toto pořadí nelze zaměnit, neboť do databáze chemických reakcí je možné vložit pouze reakce, jejichž všechny reaktanty i produkty jsou obsaženy v dané databázi chemických látek (při načítání ze souboru jsou ostatní reakce ignorovány).

### 4.5.1 Databáze chemických látek

Nejdříve je třeba vyplnit seznam látek. Tlačítkem *Vložit novou látku* zobrazíme formulář, ve kterém zadáme vlastnosti této látky.

Pro potřeby laboratoře lze zadat následující informace:

Obrázek 4.3: Ukázka vyplněného formuláře s informacemi o látce

- **vzorec** — pod pojmem chemický vzorec se může skrývat řada různých reprezentací stavby molekul látky, v závislosti na tom, v jaké míře různá odvětví chemie požadují, aby struktura chemického vzorce odpovídala struktuře molekuly látky. V programu nemá význam, jaký typ vzorce bude použit, pro přehlednost doporučuji použít jednotně *molekulární vzorec*[7] - zapisuje se jako řada symbolů prvků obsažených v dané sloučenině, případně doplněných číselnými indexy, umístěvanými vpravo za symbolem určitého prvku, kde symbol každého z těchto prvků se vyskytne v molekulárním vzorci právě jednou, seřazený většinou podle vzrůstajícího protonového čísla
- **název** — pro některé látky může být užíváno více chemických názvů, výběr je opět na zadání uživatele, nemá vliv na chod laboratoře
- **CAS-RN** *Chemical Abstracts Service Registry Number* — jedná se o unikátní číslo, přidělené každé světově registrované látce. Tímto číslem je indexována oficiální *Chemical Abstracts* databáze a tento program rozlišuje látky podle něj. Jedná se o číslo ve formátu Y-XX-X, kde Y zastupuje 2–6 číslic a X zastupuje jednu číslici, např. tedy 7732-18-5 pro vodu. Toto číslo musí být v celé databázi jedinečné, při vícenásobném zadání stejného čísla budete dotázáni, zda má být původní látka přepsána či zadáte jiné informace
- **relativní molekulová hmotnost** - bezrozměrná veličina, jejíž jednotkou je 1/12 hmotnosti atomu uhlíku
- **hustota** — jednotka kg / m<sup>3</sup>
- **barva** — vyberte barvu látky pomocí tlačítka *Upravit barvu* — zobrazí se okénko, ve kterém je možné pomocí tří posuvníků vybrat žádanou kombinaci barev červené, zelené a modré

- **bod tání** — teplota, při které pevná krystalická látka přechází do kapalného skupenství. Je rovna *teplotě tuhnutí*, při které látka přechází ze stavu kapalného do stavu pevného. U amorfních látek je třeba zadat alespoň přibližnou hodnotu. Zaškrtnutím volby *Není znám* určujeme, že tato hodnota nebyla zjištěna a změna na kapalné (či pevné) skupenství nebyla v praxi dosažena ani za extrémních teplot. Bod tání zadáváme v jednotce  $^{\circ}\text{C}$
- **bod varu** — Obdobně jako bod tání, jedná se o změnu skupenství látky mezi kapalným a plynným. V případě, že látka sublimuje, zadáme stejnou hodnotu jako 4.5.1

V případě, že není určen bod tání ani bod varu, vybereme skupenství, ve kterém se bude tato látka v laboratoři vyskytovat. Pokud jsme zadali alespoň jeden z bodu tání či bodu varu, látka je ze označena jako *stabilní*

- **měrné skupenské teplo tání** — množství tepla, které je třeba dodat (odebrat) 1 kg látky, aby došlo ke změně skupenství z pevného na kapalné (z kapalného na pevné). Zadáváme v jednotce  $\frac{\text{J}}{\text{kgK}}$
- **měrné skupenské teplo varu** — obdobně jako *měrné skupenské teplo tání*
- **rozpuštěnost ve vodě** — některé látky se při výskytu ve vodě částečně rozpustí a vzniká roztok, tedy molekuly rozpuštěné látky jsou rozptýleny po celém objemu vody. Udáváme v  $\frac{\text{mg}}{\text{dm}^3}$  (maximální počet gramů látky rozpuštěné ve 100 ml vody). Nula označuje látku ve vodě nerozpustnou; pokud není množství rozpuštěné látky omezeno, zadáme záporné číslo.
- **střední tepelná kapacita** — veličina měrná tepelná kapacita udává, kolik tepla je třeba přidat látce, aby zvýšila svou teplotu o 1 Kelvin. Tato veličina mění svoji hodnotu spolu se změnou teploty látky, její závislost na teplotě ovšem nelze zatím obecně popsat. Pro technické výpočty se tudíž udává tzv. střední tepelná kapacita, která je pro látku každou určena jednoznačně. Zadáme v jednotce  $\frac{\text{J}}{\text{kgK}}$
- poslední důležitou informací je to, zda bude látka **dostupná v laboratoři jako surovina**, tedy zda bude možné ji do některé z baněk explicitně vložit (v praxi by této možnosti odpovídala skutečnost, zda si můžeme balení této látky koupit). V opačném případě je výskyt této látky v baňce možný pouze ve formě produktu některé chemické reakce

Velmi často se může stát, že neznáme všechny požadované údaje. V tom případě nemusíme nutně upustit od vložení této látky do databáze — program použije defaultní hodnoty na úkor pravdivostní hodnoty experimentu. Významnost zadání jednotlivých vlastností je popsána v následující tabulce:

Vlastnost	Defaultní hodnota
vzorec	povinná
název	povinná
CAS-RN	povinná unikátní
rel. mol. hmotnost	povinná
hustota	povinná
barva	černá
bod tání	není nutná
bod varu	není nutná
m. s. teplo tání	0
m. s. teplo varu	0
rozpuštnost ve vodě	0
stř. tep. kap.	při hodnotě 0 je použita jako defaultní hodnota střední tepelná kapacita vody $4180 \frac{J}{kgK}$
dostupnost látky jako suroviny	volitelná

Po potvrzení je vložení položky do databáze vyznačeno přidáním vzorce látky do abecedně uspořádaného seznamu látek databáze. Opětovně můžeme vyvolat formulář po označení vzorce látky v seznamu vzorců všech látek v databázi laboratoře příkazem *Upravit látku*, příkazem *Smazat látku* označené látky vymažeme.

Pro pozdější využití již zadané kolekce látek máme možnost uložit celou databázi ve formě XML souboru.

## 4.5.2 Databáze chemických reakcí

V dalším kroku je možné určit, jaké reakce budou v laboratoři moci probíhat. Stiskem tlačítka *Vložit novou reakci* zobrazíme dialog, ve kterém můžeme editovat údaje o vkládané reakci.

Obrázek 4.4: Ukázka vyplněného formuláře s informacemi o reakci

V pravém podokně vidíme seznam vzorců látek, které jsme vložili do databáze v předchozím kroku. V levé části je vykreslen seznam reaktantů a seznam produktů reakce. Látku přidáme mezi reaktanty (produkty) tak, že ji označíme v seznamu látek a klikneme na tlačítko *Vložit reaktant (Vložit produkt)*. Látka se zobrazí v odpovídajícím okně levé části obrazovky. Číslo v závorce udává stechiometrický koeficient této látky v chemické rovnici této reakce. Zvýšit jej můžeme opětovným stisknutím tlačítka *Vložit reaktant*; reaktant či produkt označený v odpovídajícím okně v levé části dialogu smažeme stisknutím tlačítka *Smazat reaktant (Vložit produkt)*.

V dolní části obrazovky jsou políčka určená k zadání aktivační energie a frekvenčního faktoru, ze kterých bude počítána rychlost reakce.

Po potvrzení informací je chemická rovnice nově přidané reakce zobrazena v seznamu reakcí. Editovat reakci můžeme jejím označením a stisknutím tlačítka *Upravit reakci* či dvojklikem na její rovnici. Seznam použitých reakcí a jejich údajů můžeme archivovat uložením do XML souboru pomocí tlačítka *Uložit soubor*.

Načíst již uložené informace můžeme pomocí tlačítka *Načíst soubor*. Při načítání reakcí ze souboru jsou vloženy pouze ty reakce, jejichž reaktanty i produkty jsou obsaženy v databázi látek.

## 4.6 Podrobný popis pracovního prostředí

### 4.6.1 Menu a toolbar

Nabídka menu v horní části hlavního okna obsahuje většinu funkcí programu. Nej-používanější příkazy mají kvůli rychlé a pohodlnější dostupnosti své zastoupení i v nástrojové liště, která se nachází pod menu. Většina příkazů má také vlastní kombinaci řídicích kláves. Příkazy z menu lze rozdělit do několika částí:

**Soubor** - obsahuje hlavní příkazy pro práci s programem a se samostatnou laboratoří

- **Nová laboratoř ...** — vytvoří novou laboratoř obsahující defaultní databáze informací. Tento příkaz je možné vyvolat také stisknutím kombinace kláves *Ctrl-N*
- **Nová laboratoř ...** — spustí průvodce pro vytvoření nové laboratoře (popsáno v 4.5).
- **Otevřít ...** — zobrazí dialog pro otevření uložené laboratoře. Tento příkaz je možné vyvolat také stisknutím kombinace kláves *Ctrl-O*
- **Uložit** — uloží laboratoř. Pokud aktuální laboratoř ještě nebyla uložena, zobrazí dialog po výběr názvu souboru
- **Uložit jako ...** — Zobrazí dialog pro výběr názvu souboru a uloží aktuální laboratoř
- **Konec** — ukončí program (*Alt-F4*)

**Úpravy** - obsahuje příkazy pro editační akce v programu



- **Zpět** — vrátí zpět poslední editační akci
- **Vpřed** — Zopakuje poslední editační akci
- **Vyjmout** — přesune označené objekty do schránky programu
- **Vložit** — zkopíruje označené objekty do schránky programu
- **Smazat** — vymaže označené objekty
- **Vybrat vše** — označí všechny objekty v laboratoři

**Zobrazení** - obsahuje dvě podsložky Zobrazit a Okna, lišící se svým posláním:

- **Zobrazit / Všechny látky** — zobrazí databázi všech chemických látek, jejichž údaje jsou zadány v databázi laboratoře. Poklepáním na vzorec látky je zobrazeno okno se zadanými vlastnostmi této látky.
- **Zobrazit / Všechny reakce** — zobrazí chemické rovnice všech reakcí, které byly zadány do databáze
- **Zobrazit / Reakce surovin** — zobrazí pouze ty reakce, jejichž reaktanty tvoří pouze látky, které je možné vložit do baňky pomocí příkazu *Vložit látku*
- **Okna** — odškrtnutím či zaškrtnutím některé z položek dojde ke změně viditelnosti příslušného podokna (zobrazeno / nezobrazeno)

**Vložit zařízení** — dojde k vložení nového objektu, jehož typ odpovídá názvu příslušné položky.

**Baňka** — obsahuje příkazy pro manipulaci s baňkou

- **Vložit chemikálie** — zobrazí dialogové okno pro výběr látky a množství, v jakém má být vložena do baňky. Vkládá chemikálie pouze do baňky, která byla označena jako poslední
- **Měřit teplotu** — všem označeným baňkám změní viditelnost jejich teploměru (mezi zobrazen / nezobrazen)
- **Překlopit baňku** — všechny označené baňky, které nejsou nakloněny, nakloní, Všechny označené baňky, které nakloněny byly, vrátí do normální polohy

**Skript** — obsahuje příkazy pro manipulaci se řídicími skripty

- **Vytvořit nový** — zobrazí průvodce pro vytvoření nového skriptu
- **Upravit ...** — zobrazí průvodce pro editaci řídicího skriptu
- **Otevřít ...** — načte spustitelný skript ze souboru
- **Spusť** — spustí skript
- **Stop** — zastaví skript

**Nastavení** — otevře dialog pro nastavení programu

## 4.6.2 Podokna

Hlavní okno aplikace je rozděleno na několik podoken:

- Pracovní plocha laboratoře. Zde je možné umísťovat baňky, kahaný a mrazáky
- Podokna zobrazující seznam baněk, kahanů a mrazáků.
- Podokno *Poznámky* umožňuje uživateli zapsat poznámky ke svému pokusu. Pokud je spuštěn řídicí skript, v tomto okně jsou zobrazeny poznámky zadané při tvorbě skriptu do okna není možné zapisovat
- Podokno *Skript* je zobrazeno pouze při práci řízené spuštěným skriptem. Obsahuje poznámky k aktuálnímu kroku.

Při kliknutí na jméno objektu v seznamu baněk, kahanů či mrazáků dojde k označení tohoto objektu, celá plocha laboratoře je posunuta tak, aby se objekt dostal do středu obrazovky.

## 4.6.3 Ovládání

Program vyžaduje ovládání pomocí myši, mnoho příkazů v menu je ovšem přístupných i pomocí klávesové zkratky. Pomocí klávesnice lze měnit i pozici objektů na ploše — pomocí šipek je možné přesouvat označené objekty.

## 4.6.4 Nastavení

Výběrem položky menu *nastavení* zobrazíme dialogové okno, kde můžeme měnit některé vlastnosti laboratoře. Na záložce *Vzhled* můžeme nastavit barvu pozadí a vykreslovanou barvu skla baněk. Na záložce *Výpočty* můžeme nastavit teplotu, jakou budou mít chemické látky při vkládání do baňky pomocí dialogu, minimální a maximální teplotu laboratoře. Hodnota *Zaokrouhlovat* nám udává, s jakou přesností bude prováděn výpočet v laboratoři.

# Kapitola 5

## Srovnání s jinými programy

Existuje mnoho programů, které se zabývají chemickou tematikou, od vykreslování strukturovaných chemických vzorečků po složité odborné aplikace, zabývající se výpočtem specifických vlastností jednotlivých chemikálií.

Projekty zaměřené na simulaci laboratoře s účelem výuky základů chemie bohužel nejsou zřejmě atraktivním tématem, což je zapříčiněno pravděpodobně tím, že chemická pravidla není dost dobře možné "naprogramovat" a většina údajů tedy musí být zadána pomocí databáze.

Jediný podobný program je komerční program ChemLab.

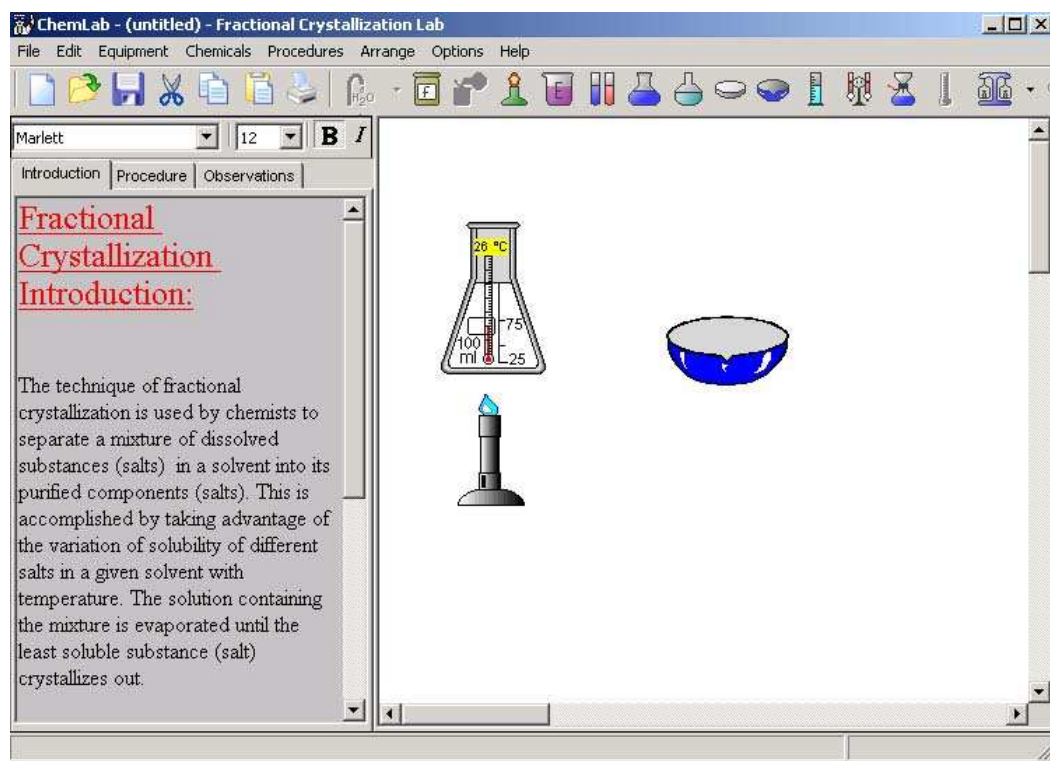
### 5.1 ChemLab 2.3

Program ChemLab je vyvíjen již dlouhou dobu, testovali jsme jeho demoverzi. Při otevření nabízí program na výběr z několika pokusů, každý s jiným zaměřením. Do laboratoře je možné vkládat různé typy baněk nebo kahany. Vkládání látek do baňky je implementováno pomocí podobného dialogu jako u programu Chemica, je zde navíc možnost permanentně měřit hmotnost baňky. Při umístění nad hořákem je zvyšována teplota v baňce, teplotu baněk je možné snižovat pouze vložení látky o nižší teplotě. V nástrojové liště je navíc položka pro vkládání vody do baňky, nemusí být tedy často hledána v seznamu ostatních chemikálií. Vodu je možné přidávat o teplotě +20 či -20 stupňů Celsia. Část obrazovky zabírá plocha, kde je popsán pokus a umožněno zapisovat vlastní poznámky.

#### Srovnání s programem Chemica

Na rozdíl od programu Chemica je aplikace ChemLab vyvíjena skupinou odborníků již několik let, což se podepsalo na větší propracovanosti této aplikace. Má implementovanu větší škálu chemických vlastností, například je možné měřit pH pomocí indikátorů nebo možnost zamíchat obsah baňky tyčinkou a urychlit tak rozpouštění látky ve vodě. Je zde více druhů nádob, některé z nich mají speciální funkce — v programu Chemica se jednotlivé baňky liší pouze tvarem a objemem. Je zde lépe propracována podpora poznámek, uživatel si může vybrat i druh a velikost písma.

Oproti tomu program Chemica má některé funkce vylepšeny. Například podporuje funkci Undo / Redo, která u programu ChemLab chybí. Umožňuje také snižovat teplotu v baňce pomocí mrazáku, v programu ChemLab je možné ochlazovat pouze přidáním ledu o teplotě  $-20\text{ }^{\circ}\text{C}$ . Dále zde není možnost změny nastavení — teplota přidávaných látek je pevně dána ( $+20\text{ }^{\circ}\text{C}$ ), není možné změnit barvu pozadí ani obrysů baněk. Všechny tyto hodnoty lze v programu Chemica nastavit. Nejvýznamější výhodou programu Chemica je možnost vytváření řídicích scénářů — v programu ChemLab není nijak možné interaktivně zasahovat do činnosti uživatele.



Obrázek 5.1: Program Chemlab

# Kapitola 6

## Závěr

Cílem této bakalářské práce bylo vytvořit program, který by simuloval chování obsahů baněk v chemické laboratoři a umožňoval tak uživateli práci v ní vyzkoušet z domova či ze školy. V rámci práce vznikl program Chemica, který toto v určité míře umožňuje.

Snahou bylo vytvořit co nejjednodušší a nejnázornější uživatelské rozhraní, aby program mohli využívat žáci ve škole i běžní uživatelé ovládající základní práci s počítačem.

Program získal důležitou vlastnost — možnost utvářet skripty, které po spuštění vedou uživatele krok za krokem po celou dobu pokusu.

V programu jsou implementovány některé základní chemické a fyzikální jevy, rozmanitost těchto věd dává programu nekonečné množství možností rozšíření.

### 6.1 Další budoucnost programu

V budoucnu by program mohl být rozšířen o mnoho dalších funkcí a vlastností:

- Implementace dalších chemických jevů a vlastností
- Dokonalejší vykreslení objektů
- Možnost uložení grafického vykreslení pracovního prostředí do obrázku, vytisknutí
- Implementace podmíněného vyhodnocování řídicích skriptů
- Zavedení více příkazů řídicích skriptů

# Literatura

- [1] Ing. Vratislav Šrámek: Obecná a anorganická chemie, Nakladatelství Olomouc, 2000
- [2] Mgr. Marika benešová, Mgr. Hana Satrapová: Odmaturuj z chemie, DIDAKTIS, 2002
- [3] Jiří Vacík: Fyzikální chemie, SNTL, 1986
- [4] prof. RNDr. Jiří Vacík, DrSc. a kolektiv, Přehled středoškolské chemie, SPN, 1995
- [5] E. Svoboda a kol.: Přehled středoškolské fyziky, Prometheus, 1996
- [6] Dr. Heinrich Remy: Anorganická chemie, SNTL, 1971
- [7] [www.wikipedia.org](http://www.wikipedia.org): Chemický vzorec,  
[http://cs.wikipedia.org/wiki/Chemick%C3%BD\\_vzorec](http://cs.wikipedia.org/wiki/Chemick%C3%BD_vzorec)
- [8] Roztoky - základní poznatky,  
[http://xantina.hyperlink.cz/roztoky/zakl\\_poznatky.html](http://xantina.hyperlink.cz/roztoky/zakl_poznatky.html)
- [9] Richard J. Simon, Michael Gouker, Brian C. Barnes: Win32 API - průvodce vývojáře , UNIS Publishing, 1997

# Příloha A

## XML Schema souboru látek

```
<?xml version="1.0" encoding="windows-1250"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="cis">
    <xs:sequence>
      <xs:element name="bas" type="xs:integer" />
      <xs:element name="exp" type="xs:integer" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="datalatka">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="latka" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <!-- vlastnosti latky -->
            <xs:sequence>
              <xs:element name="CAS-RN" type="xs:string"/>
              <xs:element name="nazev" type="xs:string"/>
              <xs:element name="vzorec" type="xs:string"/>
              <xs:element name="r_mol_hm" type="xs:cis"/>
              <xs:element name="hustota" type="xs:cis"/>
              <xs:element name="bod_tani" type="xs:integer"/>
              <xs:element name="bod_varu" type="xs:integer"/>
              <xs:element name="barva" >
                <xs:complexType>
                  <!-- barva RGB -->
                  <xs:sequence>
                    <xs:element name="R" type="xs:integer"/>
                    <xs:element name="G" type="xs:integer"/>
                    <xs:element name="B" type="xs:integer"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="rozpustnost" type="xs:cis"/>
              <xs:element name="stk" type="xs:cis"/>
              <xs:element name="mstt" type="xs:cis"/>
              <xs:element name="mstv" type="xs:cis"/>
              <xs:element name="stav" type="xs:integer"/>
              <xs:element name="surovina" type="xs:integer"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

# Příloha B

## XML Schema souboru reakcí

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="cis">
    <xs:sequence>
      <xs:element name="bas" type="xs:integer" />
      <xs:element name="exp" type="xs:integer" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="datareakce">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="reakce" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="reaktant" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <!-- casrn latky a pocet molekul -->
                  <xs:sequence>
                    <xs:element name="casrn" type="xs:string"/>
                    <xs:element name="i" type="xs:integer"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="produkt" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <!-- casrn latky a pocet molekul -->
                  <xs:sequence>
                    <xs:element name="casrn" type="xs:string"/>
                    <xs:element name="i" type="xs:integer"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="akt_ener" type="xs:cis"/>
              <xs:element name="frek_fakt" type="xs:cis"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



# Příloha C

## XML Schema uložené laboratoře

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="cis">
    <xs:sequence>
      <xs:element name="bas" type="xs:integer" />
      <xs:element name="exp" type="xs:integer" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="chemsave">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="zaokrouhleni" type="xs:string"/>
        <xs:element name="teplota_labu" type="xs:string"/>
        <xs:element name="min_teplota" type="xs:string"/>
        <xs:element name="max_teplota" type="xs:string"/>
        <xs:element name="Backcolor" type="xs:string"/>
        <xs:element name="Bankacolor" type="xs:string"/>
        <xs:element name="poznamka" type="xs:string"/>
        <xs:element name="laborator_datalatka " type="xs:string"/>
        <xs:element name="laborator_datareakce " type="xs:string"/>
        <xs:element name="objekt" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="jmeno" type="xs:string"/>
              <xs:element name="typ0" type="xs:integer"/>
              <xs:element name="posX" type="xs:integer"/>
              <xs:element name="posY" type="xs:integer"/>
              <xs:element name="naklon" type="xs:boolean"/>
              <xs:element name="teplota" >
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="sign" type="xs:boolean"/>
                    <xs:element name="casrn" type="xs:string"/>
                    <xs:element name="i" type="xs:integer"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:element name="merenateplota" type="xs:boolean"/>
            <xs:element name="akt_ener" type="xs:cis"/>
            <xs:element name="frek_fakt" type="xs:cis"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

# Příloha D

## XML Schema řídicího skriptu

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="cis">
    <xs:sequence>
      <xs:element name="bas" type="xs:integer" />
      <xs:element name="exp" type="xs:integer" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="chem\_skript">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="chem\_nazev" type="xs:string"/>
        <xs:element name="teplota\_labu" type="xs:string"/>
        <xs:element name="chem\_pozn" type="xs:string"/>
        <xs:element name="chem\_autor" type="xs:string"/>
        <xs:element name="chem\_end" type="xs:string"/>
        <xs:element name="akce" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType><xs:all>
            <xs:element name="typ" > <!-- typ akce -->
              <xs:restriction base="xs:string">
                <xs:enumeration value="S\_INSERT\_ZK"/> <!-- vlozit banku-->
                <xs:enumeration value="S\_INSERT\_HOR"/> <!-- vlozit kahan-->
                <xs:enumeration value="S\_INSERT\_MRAZ"/> <!-- vlozit mrazak-->
                <xs:enumeration value="S\_INSERT\_IN"/> <!-- vlozit latku-->
                <xs:enumeration value="S\_ZMERIT"/> <!-- zmerit teplotu-->
                <xs:enumeration value="S\_PRELIT"/> <!-- prelit neco-->
                <xs:enumeration value="S\_CEKAT"/> <!-- cekat cas-->
                <xs:enumeration value="S\_TEPLOTA"/> <!-- zmena teploty-->
                <xs:enumeration value="S\_DELETE"/> <!-- smazat objekt-->
              </xs:restriction>
            <xs:element name="poznamka" type="xs:string"/>
            <xs:element name="typBanky" type="xs:integer"/>
            <xs:element name="objektRefer" type="xs:integer"/>
            <!-- poradi vlozeného objektu, na který odkazuje -->
            <xs:element name="objektRefer2" type="xs:integer"/>
            <!-- poradi vlozeného objektu, na který odkazuje -->
            <xs:element name="omezeni" type="xs:integer"/>
            <!-- způsob zadání číselného omezení -->
            <xs:element name="time" type="xs:integer"/>
            <xs:element name="valMin" type="xs:decimal"/> <!-- omezení minima -->
            <xs:element name="valMax" type="xs:decimal"/> <!-- omezení maxima -->
            <xs:element name="valEqual" type="xs:decimal"/>
            <xs:element name="valTol" type="xs:decimal"/>
          </xs:all> </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```