

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

BAKALÁŘSKÁ PRÁCE

Program pro podporu výuky číslicové techniky



2010

František Mrtvý

Anotace

Program pro podporu výuky číslicové techniky je software, který má pomoci žákům Vyšší odborné školy a Střední průmyslové školy elektrotechnické Božetěchova 3, Olomouc zvládnout a procvičit látku předmětu Číslicová technika. Konkrétně jsou to kapitoly o převodech mezi číselnými soustavami a počítání s čísly v dvojkové soustavě.

Chtěl bych poděkovat za vedení této práce zejména panu Mgr. Jiřímu Zápalo-
lovi, Ph.D. Dále patří můj dík paní Ing. Zuzaně Veselé a Ing. Liboru Kyselému,
učitelům VOŠ a SPŠ Elektrotechnické Božetěchova, za pomoc při sestavování
konkrétního zadání práce a v neposlední řadě mé rodině, zejména manželce Lucii
Mrtvé.

Obsah

1. Úvod	7
1.1. Téma převodů čísel mezi různými soustavami	7
1.1.1. Příklady aplikací zabývajících se převody mezi číselnými soustavami	7
1.2. Téma počítání v binární soustavě	8
1.2.1. Příklady aplikací zabývajících se počítáním v binární soustavě	8
1.3. Shrnutí	10
2. Uživatelská část	11
2.1. Pracovní plocha	11
2.2. Hlavní menu	12
2.3. Procvičování v CIT	13
2.3.1. Panel průvodce pro převod D do B/H	14
2.3.2. Panel průvodce pro převod B/H do D	15
2.3.3. Panel průvodce pro převod B/H do H/B	17
2.3.4. Panel průvodce sčítáním v dvojkové soustavě	18
2.3.5. Panel průvodce vyjádřením záporného čísla v dvojkové soustavě	19
2.3.6. Panel průvodce odčítáním čísel v binární soustavě	19
2.4. Testování v CIT	20
2.4.1. Nastavení testu	20
2.4.2. Test	21
3. Administrátorská část	22
3.1. Nabídka Administrátor	22
3.1.1. Správci	22
3.1.2. Upravit rejstřík	23
3.2. Správa testů	24
3.2.1. Nový test	24
3.2.2. Správa souborů	25
3.2.3. Zadání testu	27
3.2.4. Vyhodnotit test	28
3.2.5. Historie	29
3.3. Přidávání kapitol	29
4. Programátorská část	31
4.1. Třída FormCIT	31
4.1.1. Nástroje pro administraci	31
4.1.2. Nástroje pro cvičení a testování	32
4.2. Třída Form_sprava	34
4.2.1. Vytváření a editování testů	34

4.2.2. Spouštění a vyhodnocování testů	35
4.3. Třída Form_test	35
4.4. Třída From_historie	35
4.5. Soubory typu *.ctf (CIT test file)	36
5. Závěr	37
6. Conclusions	38
Reference	39

Seznam obrázků

1.	Aplikace JHmatika	8
2.	Aplikace emu8086	9
3.	Aplikace Binary Adder 1	9
4.	Základní okno aplikace CIT	11
5.	Vyhledávání výrazů v textech	12
6.	Panel pro cvičení	13
7.	Panel pro převod z desítkové soustavy do dvojkové	14
8.	Tlačítko pro přidání elementu dělení a tlačítko pro odebrání elementu dělení	15
9.	Panel pro převod z dvojkové soustavy do desítkové	16
10.	Tlačítko pro přidání elementu součinu a tlačítko pro odebrání elementu součinu	16
11.	Panel pro převod z dvojkové soustavy do šestnáctkové	17
12.	Tlačítka Přidat panel součtu a Odebrat panel součtu	18
13.	Panel součtu	18
14.	Panel záporného čísla	19
15.	Tlačítka pro přidání a odebrání panelu záporného čísla	19
16.	Průvodce pro odčítání čísel v dvojkové soustavě	20
17.	Formulář pro nastavení testu	21
18.	Formulář testu	22
19.	Formulář Upravit admina	23
20.	Formulář Upravit rejstřík	23
21.	Formulář pro správu testů s vytvořeným novým testem	25
22.	Panel pro vytváření a editování úloh převodů mezi soustavami	26
23.	Panel pro vytváření a editování úloh počítání v binární soustavě	26
24.	Soubory příslušné k aplikaci	27
25.	Panel vyhodnoceného testu – celkový přehled	28
26.	Panel vyhodnoceného testu – prohlížení testů	29
27.	Formulář Historie	30

1. Úvod

Tématem, kterým se tato aplikace zabývá, jsou základy předmětu Číslicová technika. K vytvoření aplikace CIT 1.0 (dále jen CIT) vedla zkušenost potřebnosti názorného vysvětlení dané látky. V době, ve které žijeme, je již výpočetní technika na takové úrovni, že se její aplikace v pedagogické činnosti stává čím dál častější.

Témata zpracovaná v této práci jsou vhodná pro takovéto zpracování zejména kvůli tomu, že podstatou jejich pochopení a řešení příslušných úloh je naučit se algoritmus, postup jejich řešení, který se pak aplikuje na všechny ostatní úlohy.

Základním cílem této práce je tedy snaha nabídnout studentovi prostředek k názornému zafixování těchto postupů v daném tématu.

1.1. Téma převodů čísel mezi různými soustavami

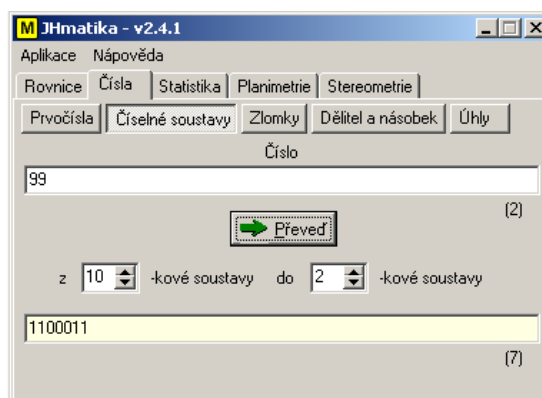
Z historie víme, že lidé se vždy snažili zachytit informaci pomocí čísel. Dělali to více či méně praktickým způsobem. Ale který způsob je ten nepraktičtější? Odpověď jistě není jediná, protože je úzce spjata s účelem použití daného zápisu.

V číslicové technice, tudíž i v této práci, se budeme bavit o číselných soustavách o základech 2, 10 a 16, které jsou velmi častým vyjádřením informací vůbec v celé výpočetní technice. Číselné soustavě, která má svůj základ v čísle 2 a která používá k vyjadřování znaků 1 a 0, říkáme dvojková (binární). Podobně definujeme též soustavu desítkovou (dekadickou) o základu 10, je to nám známá soustava se znaky 0—9 a šestnáctkovou (hexadecimální) o základu 16 a s prvky 0—15 [1].

1.1.1. Příklady aplikací zabývajících se převody mezi číselnými soustavami

Aplikace Kalkulačka v systému Windows Kalkulačka obsahuje spoustu nástrojů pro práci s čísly. Jedním z nich je i nástroj pro převádění mezi číselnými soustavami. Je realizován panelem RadioButtonů přiřazených k jednotlivým soustavám. Při výběru každého z nich je číslo v textovém poli převedeno do příslušné soustavy. Kalkulačka je, nutno říct, praktický nástroj, ale není to výukový software, slouží spíše ke kontrole vypočítaného.

Aplikace JHmatika [6] Autor této mnohostranné matematické aplikace je Jiří Hajšl. Aplikace je schopná převádět čísla z jedné soustavy do druhé, ale také řešit kvadratické rovnice, kubické rovnice, soustavy rovnic, počítat statistické hodnoty a hodnoty planimetrie a stereometrie. Převodník soustav je v této aplikaci schopen převádět soustavy o základu 2–36.



Obrázek 1. Aplikace JHmatika

Aplikace emu8086 [7] Anglická aplikace, která je narozdíl od již zmíněných výukový software. Je zaměřená na výuku pokročilejších kapitol číslicové techniky jako je programování mikroprocesoru v assembleru. Dílčí funkcí této aplikace je i převodník mezi soustavami, který dokáže i namodelovat postup, ovšem pro začátečníka je obtížné se v programu zorientovat.

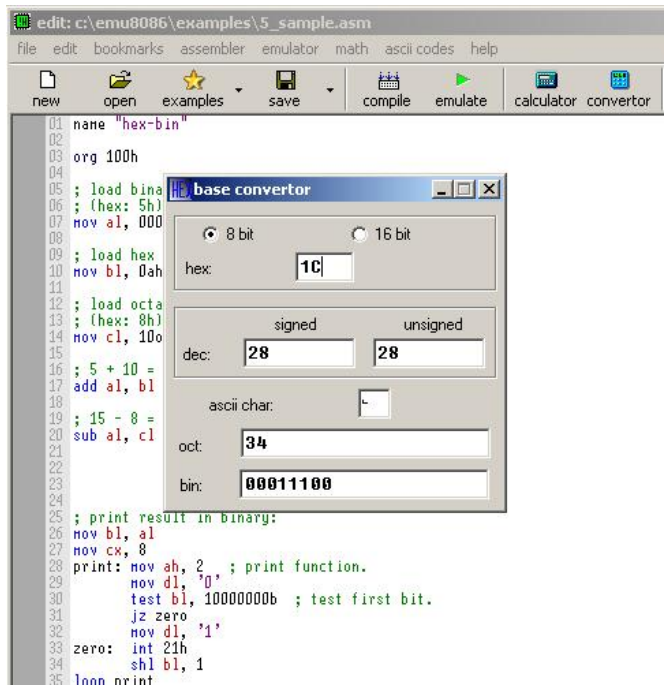
1.2. Téma počítání v binární soustavě

Na kapitolu 1.1. navazuje další kapitola o počítání v binární soustavě, která je užitečná k prohloubení představy o práci jednoduchého počítače. Po prostudování této kapitoly by měl žák umět vysvětlit princip sčítání a odčítání, na kterém pracuje jednoduchá aritmeticko-logická jednotka.

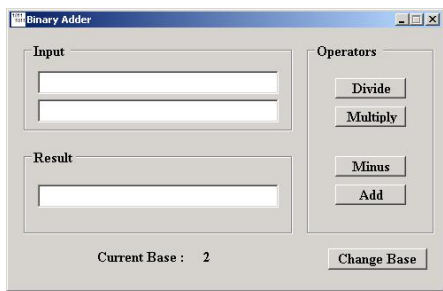
1.2.1. Příklady aplikací zabývajících se počítáním v binární soustavě

Aplikace Kalkulačka v systému Windows Velmi praktický pomocník k výuce. Stejně jako u převodů soustav však kalkulačka vrací pouze výsledky, které ale nic nevyprávějí o postupu výpočtu. Poslouží jako kontrola. Kalkulačka umí pracovat i se znaménkovými vlastnostmi binárních čísel.

Aplikace Binary Adder 1 [8] Jednoduchá aplikace v anglickém jazyce, která umožňuje sčítat, odčítat, ale i násobit a dělit v různých soustavách včetně binární. Nevýhodou je ovšem opět nedostatečná prezentace postupů.



Obrázek 2. Aplikace emu8086



Obrázek 3. Aplikace Binary Adder 1

1.3. Shrnutí

Reakcí na předcházející průzkum je podoba aplikace CIT, která je určena především studentům „začátečnickům“ v předmětu Číslicová technika a jejich učitelům.

Jejím cílem je:

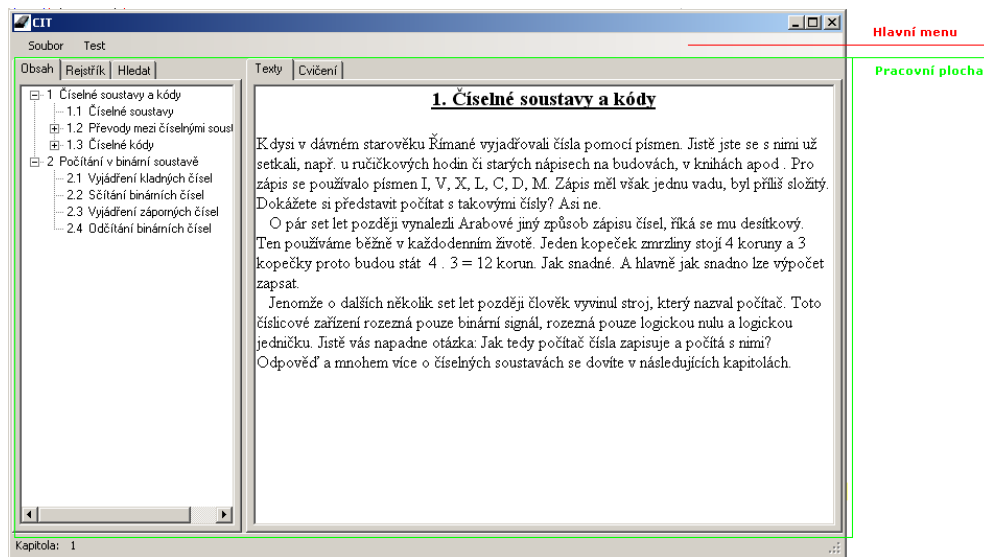
- Nabídnout studentovi přístup ke studijním materiálům vypracovaným jeho učitelem.
- Nabídnout studentovi přístup k velkému počtu řešených úloh.
- Krok po kroku provést studenta postupem řešení zadaných úloh.
- Nabídnout učiteli možnost zadat test studentům a rychle jej vyhodnotit.

Ve spolupráci s učiteli VOŠ a SPŠ Elektrotechnické v Olomouci, pro jejichž výuku je aplikace především určena, byla vypracována textová část programu, která obsahuje materiály, podle nichž se probírají daná témata na této škole.

Program sám generuje zadání úloh, které jsou procvičovány s možností jak náhledu do již zmíněných materiálů, tak i s nápovědou vztahující se ke konkrétní úloze.

V aplikaci jsou pak možnosti cvičení na třech úrovních obtížnosti, dle kterých se program více či méně snaží pomoci studentovi v jednotlivých krocích řešení.

Testovat příslušná témata je možno na úlohách zadaných samotným učitelem. Testy studenti odevzdávají na společné místo (možnost použít i místo v síti), ze kterého potom aplikace testy vyhodnocuje.



Obrázek 4. Základní okno aplikace CIT

2. Uživatelská část

V úvodu této kapitoly si řekneme něco o instalaci programu CIT 1.0:

- Program CIT 1.0 je vytvořen pro operační systém Microsoft Windows XP SP3.
- Pro správné fungování je třeba nainstalovat zároveň knihovny Microsoft .NET Framework 3.5 a novější.
- Instalace programu probíhá standartním způsobem. Je vytvořen instalátor *CIT1.0setup.exe* v programu Inno Setup Compiler 5.3.10 [5].

Nyní už k aplikaci samotné. Pro přehlednost rozdělíme okno aplikace na dvě části (Obr. 4.), které následně popíšeme:

- **Pracovní plocha**
- **Hlavní menu**

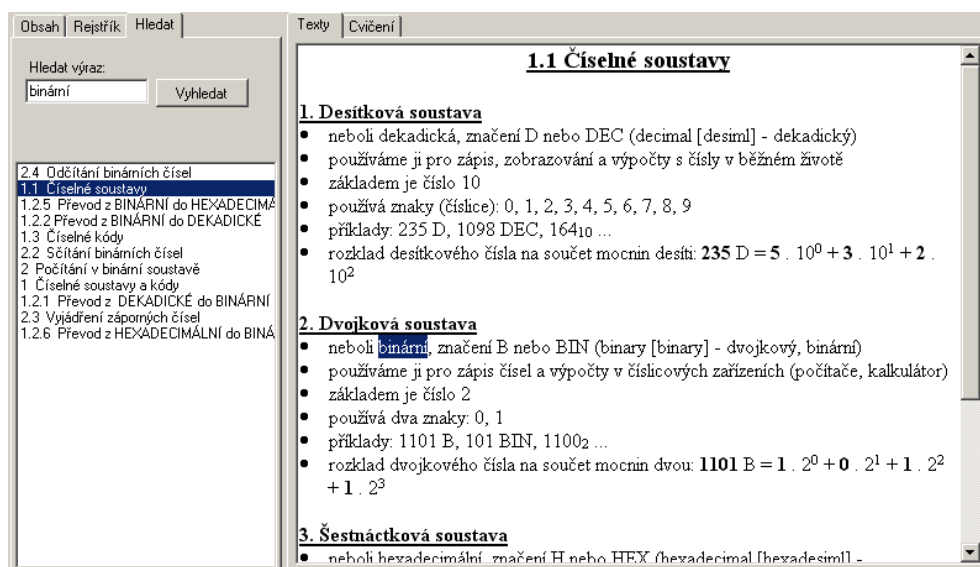
2.1. Pracovní plocha

Pracovní plocha je rozdělena do dvou částí:

- **Orientační panel** – se záložkami: Obsah, Rejstřík, Hledat
- **Pracovní panel** – se záložkami: Texty, Cvičení

Jednotlivé záložky orientačního panelu jsou propojeny s panelem pracovním. Tyto záložky slouží k výběru kapitol, na základě kterého se potom načítají do pracovního panelu příslušné texty ke kapitolám a cvičení. Záložky obsahují:

- **Obsah** – stromovou strukturu kapitol a jejich podkapitol.
- **Rejstřík** – stromovou strukturu učitelem vytvořených klíčových pojmů a k nim příslušné odkazy na kapitoly.
- **Hledat** – Textové pole pro zapsání hledaného výrazu a seznam nalezených kapitol. Po kliknutí na tlačítko hledat program prohledá všechny texty a vypíše do seznamu nalezených kapitol odkazy na ty, ve kterých výraz nalezl. Seznam nalezených kapitol je utřízen podle výskytu hledaného výrazu. Viz Obr. 5..

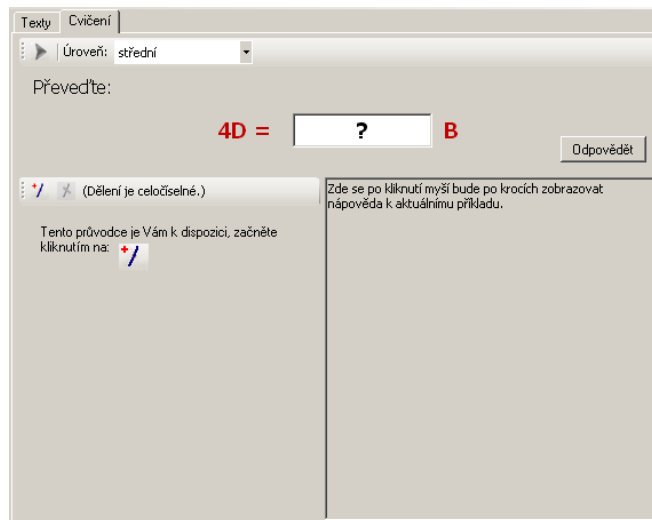


Obrázek 5. Vyhledávání výrazů v textech

Karty Texty a Cvičení pak zobrazují texty a cvičení příslušné k vybrané kapitole.

2.2. Hlavní menu

V této kapitole si krátce popíšeme položky Hlavního menu, kde se pod nabídkami s názvy Soubor a Test skrývají tyto položky:



Obrázek 6. Panel pro cvičení

- **Přihlásit...** – Pro přihlášení učitele, jako správce aplikace.
- **Odhlásit** – Odhlášení správce.
- **Správce testů...** – Tato možnost je otevřena pouze správcí.
- **Exit** – Pro zavření aplikace.
- **Nastavení...** – Zde je možné nastavit test viz. 2.4.1..
- **Spustit test** – Spuštění testu viz. 2.4.2..

2.3. Procvičování v CIT

Panely pro cvičení se liší podle zadání, ale účelem všech je co možná nejefektivněji vést studenta procesem řešení úlohy ke správné odpovědi. Některé prvky mají všechny panely společné. Krátce si o nich něco řekneme.

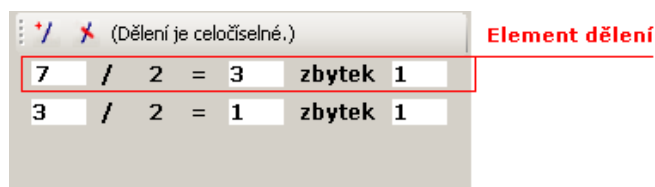
- **Panel nástrojů** – Tento prvek obsahuje **tlačítko pro generování nového příkladu** a **menu pro změnu úrovně obtížnosti**. Se změnou obtížnosti se automaticky generuje nový příklad.
- **Zadání** – Zde se vypíše přehledně zadání úlohy, která byla programem CIT vygenerována.
- **Textové pole pro odpověď** – Jeho počáteční hodnota je „?“ . Slouží k vepsání odpovědi, která bude aplikací kontrolována.
- **Tlačítko odpovědět** – Po jeho stisknutí se objeví obrázek představující buď usměvavého nebo zamračeného smajlíka podle správnosti odpovědi.

- **Panel průvodce** – Slouží studentovi jako kostra pro výpočet. Do něho si zapisuje mezivýsledky. Konečný výsledek je však třeba zapsat do *Textového pole pro odpověď*. Jednotlivé průvodce si ukážeme v následujících kapitolách.
- **Panel pro nápovědu** – Kliknutím na něj se v tomto panelu zobrazí následující krok řešení úlohy.

2.3.1. Panel průvodce pro převod D do B/H

Průvodce vytvořený pro simulaci algoritmu převodu čísla z desítkové soustavy do dvojkové resp. šestnáctkové.

Algoritmus převodu D do B/H Zadanou hodnotu v desítkové soustavě celočíselně dělíme základem cílové soustavy tj. dvěma resp. šestnácti a zapíšeme si zbytek. Vzniklý podíl pak opět dělíme dvěma resp. šestnácti. Tento postup opakujeme do chvíle, než základem dělíme nulu. Seřadíme-li nakonec zbytky od posledního dělení k prvním, dostaneme hodnotu v soustavě, na kterou převádíme. Tomuto postupu říkáme metoda postupného dělení dvěma (resp. šestnácti) [1].



Obrázek 7. Panel pro převod z desítkové soustavy do dvojkové

- **Element dělení** – Je to komponenta, ve které uživatel nastavuje nejvýše tři textové pole (dělenec, podíl po celočíselném dělení a zbytek). Dělitel, základ soustavy do které převádíme, je podle zadání doplněn do elementu automaticky.
- **Tlačítko Přidat element dělení** – Přidává v průvodci další element dělení.
- **Tlačítko Odebrat element dělení** – Odebírá poslední element dělení.



Obrázek 8. Tlačítko pro přidání elementu dělení a tlačítko pro odebrání elementu dělení

Chování průvodce v závislosti na úrovni obtížnosti

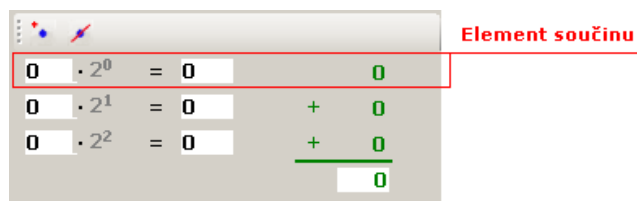
- **Lehká** – Program kontroluje během výpočtu každé zadané číslo v průvodci. Pokud najde chybu, označí číslo červeně. Pokud je v elementu dělení chyba, program zablokuje přidání dalšího elementu. Dále automaticky dosazuje hodnotu předchozího podílu do dalšího elementu dělení (v prvním elementu dosadí zadané číslo).
- **Střední** – Program nekontroluje čísla během výpočtu, ale označí chyby po stisknutí tlačítka *Odpověďt*. Dále automaticky dosazuje hodnotu předchozího podílu do dalšího elementu dělení (v prvním elementu dosadí zadané číslo). Při změně hodnoty podílu — který není poslední — mění program automaticky i hodnotu dělence následujícího elementu.
- **Těžká** – Program nekontroluje výpočet vůbec a neprovádí žádné automatické dosazení.

2.3.2. Panel průvodce pro převod B/H do D

Průvodce vytvořený pro simulaci algoritmu převodu čísla z dvojkové resp. šestnáctkové soustavy do desítkové.

Algoritmus převodu B/H do D Převáděné binární číslo si vyjádříme jako součet mocnin 2 resp. 16, přičemž respektujeme váhy cifer, výrazy vyčíslíme a sečteme [1].

- **Element součinu** – Je to komponenta, ve které student nastavuje jeden z činitelů (druhý je mocnina čísla 2 resp. 16) a zapisuje takto vzniklý součin. Součet těchto součinů je hodnota v desítkové soustavě.
- **Tlačítko Přidat element součinu** – Přidává v průvodci další element součinu.
- **Tlačítko Odebrat element součinu** – Odebírá poslední element součinu.



Obrázek 9. Panel pro převod z dvojkové soustavy do desítkové

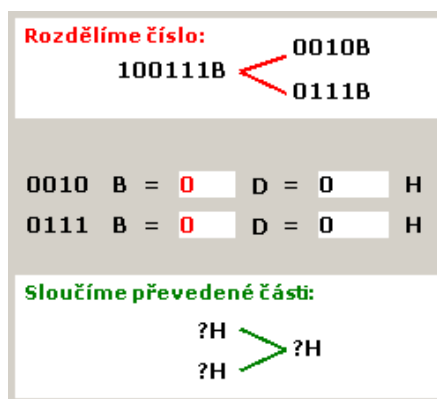
- **Textové pole pro součet pod sebou** – Textové pole pod posledním elementem součinu, do kterého se vpisují cifry z prava doleva. To kvůli možnosti sčítat součiny v zápisu pod sebou. Pro mazání cifer se používá klávesa *Delete*.



Obrázek 10. Tlačítko pro přidání elementu součinu a tlačítko pro odebrání elementu součinu

Chování průvodce v závislosti na úrovni obtížnosti

- **Lehká** – Program automaticky vypíše minimální potřebný počet elementů součinu, do kterých dosadí činitele dle váhy cifer. Dále kontroluje během výpočtu každé zadané číslo v průvodci. Pokud najde chybu, označí číslo červeně.
- **Střední** – Program automaticky vypíše minimální potřebný počet elementů součinu, do kterých dosadí činitele dle váhy cifer. Nekontroluje čísla během výpočtu, ale označí chyby po stisknutí tlačítka Odpovědět.
- **Těžká** – Jsou aktivována tlačítka pro přidávání a odebrání elementů, která si student přidává sám. V průvodci se neprovádí žádná kontrola správnosti výpočtu.



Obrázek 11. Panel pro převod z dvojkové soustavy do šestnáctkové

2.3.3. Panel průvodce pro převod B/H do H/B

Průvodce vytvořený pro simulaci algoritmu převodu čísla z dvojkové resp. šestnáctkové soustavy do šestnáctkové resp. dvojkové.

Algoritmus převodu B/H do H/B Obecně se dá použít algoritmus převodu přes desítkovou soustavu, ale kvůli zadání uvádíme tento algoritmus upravený.

Dvojkové resp. šestnáctkové číslo rozdělíme na čtveřice resp. jednotlivé znaky a každou vyjádříme přes soustavu desítkovou jedním znakem šestnáctkové soustavy resp. čtyřmi znaky dvojkové soustavy. Takto vyjádřené znaky spojíme podle váhy původních cifer [1].

Je-li vygenerované číslo v intervalu 15–255, zobrazí se v průvodci panely, které názorně zobrazují rozdělení čísla a jejich sloučení ¹ (Obr. 11.) dle výše uvedeného algoritmu. Tyto panely jsou jen pomocné, nemají v programu žádnou jinou funkci než znázornění. Důležité jsou řádky mezi těmito panely, kde student vyplňuje:

- **První převodový řádek** – 1. část rozloženého čísla, její převod do desítkové soustavy 2.3.2. a následně převod desítkového čísla do soustavy cílové 2.3.1..
- **Druhý převodový řádek** – je-li zobrazen – 2. část rozloženého čísla, její převod do desítkové soustavy 2.3.2. a následně převod desítkového čísla do soustavy cílové 2.3.1..

Chování průvodce v závislosti na úrovni obtížnosti

- **Lehká** – V případě potřeby program zobrazuje rozdělení a sloučení dle výše uvedeného algoritmu. Do převodových řádků vyplňuje automaticky jednotlivé části převáděného čísla. Po každé úpravě zapsaného čísla kontroluje jeho správnost, v případě chyby označí číslo červeně.

¹v intervalu 0–15 rozdělení není potřeba

- **Střední** – V případě potřeby program zobrazuje rozdělení a sloučení dle výše uvedeného algoritmu. Do převodových řádků vyplňuje automaticky jednotlivé části převáděného čísla. Nekontroluje čísla během výpočtu, ale označí chyby po stisknutí tlačítka Odpovědět.
- **Těžká** – Program nezobrazuje rozdělení ani sloučení částí zadaného čísla a nekontroluje správnost vstupů.

2.3.4. Panel průvodce sčítáním v dvojkové soustavě

Pravidla pro sčítání a odčítání binárních čísel viz. [1] Tento panel se skládá z menších panelů součtu, které si student přidává sám kliknutím na tlačítko *Přidat panel součtu* nebo odebírám kliknutím na tlačítko *Odebrat panel součtu* (tyto panely mohou být maximálně 4).



Obrázek 12. Tlačítka Přidat panel součtu a Odebrat panel součtu

Panel součtu Kliknutím na libovolnou cifru se neguje její hodnota. Tímto způsobem tak můžeme pohodlně zapisovat výpočty a mezivýpočty do těchto předchystaných panelů.



Obrázek 13. Panel součtu

Chování průvodce v závislosti na úrovni obtížnosti

- **Lehká** – Program vygeneruje právě dvě binární čísla, vytvoří jeden panel součtu a jako sčítance mu vyplní vygenerovaná čísla. Na studentovi je čísla jen sečíst.
- **Střední** – Program vygeneruje tři binární čísla, vytvoří dva panely součtu, které však nedoplňuje.

- **Těžká** – Program vygeneruje čtyři binární čísla. Student si sám přidává panely součtu.

2.3.5. Panel průvodce vyjádřením záporného čísla v dvojkové soustavě

Na tomto panelu je jeden menší panel pro vyjádření záporného čísla v dvojkové soustavě (dále jen „záporného čísla“). Tento panel je vytvořen tak, aby se z jeho pomocí dalo vyjádřit záporné číslo v dvojkovém doplňku [1].

Panel záporného čísla Kliknutím na libovolnou cifru se neguje její hodnota. Zadané číslo je označeno na tomto panelu znakem „A“. Znakem „ \bar{A} “ je označeno číslo, jehož každá cifra je negací příslušné cifry čísla „A“.

A	0	1	1	0	0	0	0	1
\bar{A}	0	0	0	0	0	0	0	0
+	0	0	0	0	0	0	0	1
<hr/>								
	0	0	0	0	0	0	0	0

Obrázek 14. Panel záporného čísla

Pro tento typ cvičení není možné volit úroveň obtížnosti.

2.3.6. Panel průvodce odčítáním čísel v binární soustavě

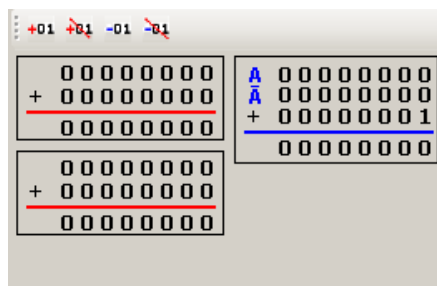
Tento panel je doplněním panelu součtu o možnost přidávat a ubírat i panel pro vyjádření záporného čísla v binární soustavě pomocí tlačítek na Obr. 15. (tyto panely mohou být maximálně 3). Odčítání pak probíhá tak, že sečteme první sčítanec s číslem vyjádřeným v dvojkovém doplňku [1].



Obrázek 15. Tlačítka pro přidání a odebrání panelu záporného čísla

Chování průvodce v závislosti na úrovni obtížnosti

- **Lehká** – Program vygeneruje dvě binární čísla, zobrazí jeden panel součtu a jeden panel záporného čísla. První vygenerované číslo vloží jako první sčítanec panelu součtu a druhé vloží do panelu záporného čísla. Číslo, které



Obrázek 16. Průvodce pro odčítání čísel v dvojkové soustavě

je výsledkem v panelu záporného čísla se automaticky promítá do druhého sčítance panelu součtu.

- **Střední** – Program vygeneruje tři čísla a vytvoří dva panely součtu a dva panely záporného čísla. Tyto panely spolu nejsou nijak propojeny.
- **Těžká** – Program vygeneruje čtyři binární čísla. Student si sám přidává panely součtu a záporného čísla.

2.4. Testování v CIT

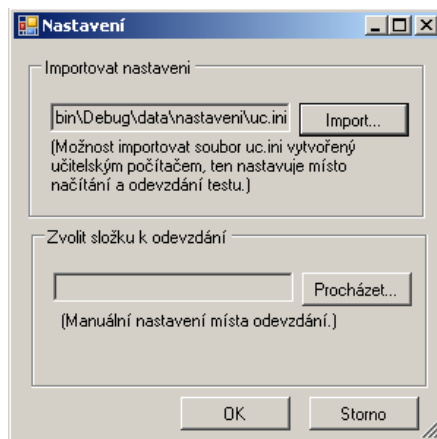
Aplikace CIT je vytvořena tak, aby mohl student načíst test, který učitel zadal, vyřešit jej a odevzdat zpět na místo k tomu určené. V této kapitole si vysvětlíme, co je k tomu potřeba.

2.4.1. Nastavení testu

Důležitou věcí pro řešení testů je nastavit místo jejich načítání a místo jejich odevzdání. To lze kliknutím na položku *Nastavení...* v nabídce *Soubor*. Nastavení lze provést dvojím způsobem (viz. Obr. 17.):

- Importováním souboru *uc.ini*, který vytvoří učitel. Je to textový soubor s příponou *.ini, ve kterém je na prvním řádku zapsaná síťová nebo lokální adresa složky, kam učitel zadává testy. Po kliknutí na tlačítko *Import...* se otevře dialogové okno, které vyzývá k tomu, abychom našli příslušný soubor, označili jej a klikli na tlačítko *Otevřít*.
- Manuálním zadáním složky pro odevzdání přes tlačítko *Procházet...*, které otevře dialog pro výběr složky.

Potvrzením OK se nastavení uloží a můžeme začít s testováním.



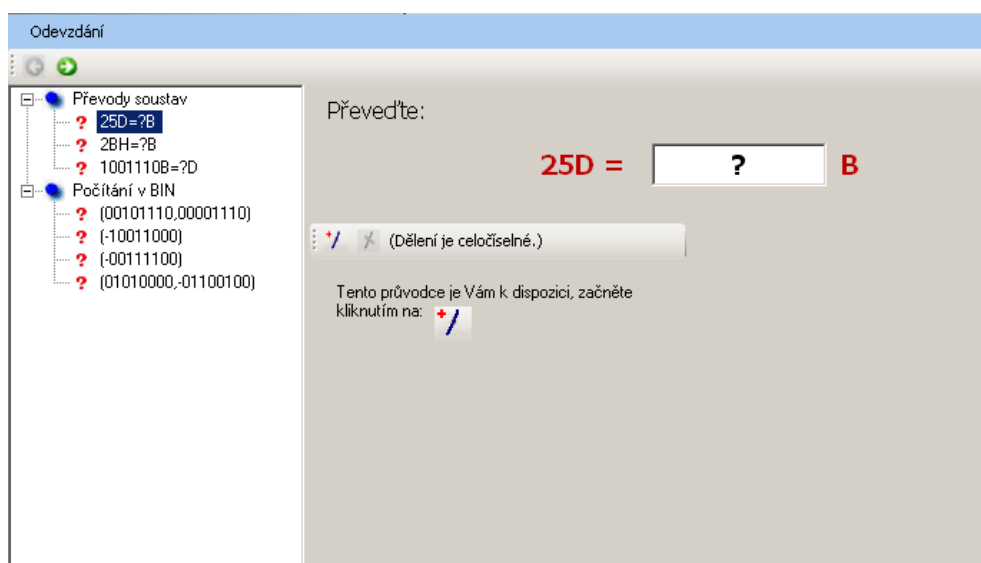
Obrázek 17. Formulář pro nastavení testu

2.4.2. Test

Klikneme-li na položku *Spustit test...*, zobrazí se jednoduchý formulář pro zadání identifikačního jména studenta a volbu konkrétního testu. Tyto dvě položky je nutno udat a kliknutím na tlačítko *Spustit* zahájíme test. Formulář spuštěného testu obsahuje výčet všech úloh ve stromové struktuře, která slouží k přepínání mezi řešením jednotlivých úloh testu, panely pro řešení jednotlivých úloh podobné panelům pro cvičení (viz. 2.3.), které se zobrazují v pravé části okna, a konečně menu, ve kterém je jen položka *Odevzdat*.

U každého zadání úlohy je na začátku ve stromové struktuře symbol **?**. Tento symbol značí, že tato úloha ještě není zodpovězená. Pokud student zodpoví danou úlohu tím, že vepíše výsledek do textového pole se symbolem „?“, tato úloha se označí symbolem . Zodpovězení úlohy se zde nekončí kliknutím na žádné tlačítko.

Student má za úkol vyřešit všechny úlohy a může k tomu použít příslušné průvodce, které jsou v režimu úrovně obtížnosti: *těžká*. Po kliknutí na položku menu *Odevzdat* program uloží vyplněný test na místo k tomu určené 2.4.1..



Obrázek 18. Formulář testu

3. Administrátorská část

V předešlých kapitolách jsme se seznámili se základními možnostmi užívání aplikace CIT. Tato další kapitola se bude zabývat možnostmi rozšířenými pro učitele (administrátora), dále jen admina. Čím se tyto dva režimy odlišují? Jsou to tyto možnosti:

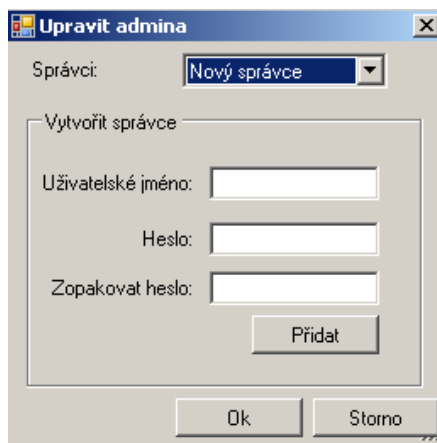
- Nabídka **Administrátor v hlavním menu**
- **Správa testů**

Pro uvedení aplikace do administrátorského režimu je třeba se přihlásit v nabídce *Soubor/Přihlásit...* V otevřeném formuláři pro admina se vyplňuje textové pole *Uživatel* a *Heslo*. Tlačítko *Přihlásit* přihlásí zadaného uživatele, je-li na seznamu uživatelů, a otevře mu výše uvedené rozšířené možnosti.

3.1. Nabídka Administrátor

3.1.1. Správci

Ve formuláři *Upravit admina* máme možnost změnit své heslo, admina přidat a odebrat.



Obrázek 19. Formulář Upravit admina

Vytvořit nového admina (v horním rolovacím menu je vybrána položka Nový správce) – Vyplníme textové pole *Uživatelské jméno*, *Heslo* a *Zopakovat heslo* a přidáme jej kliknutím na tlačítko *Přidat*.

Změnit své heslo (v horním rolovacím menu je vybrána položka s mým uživatelským jménem) – Vyplníme textové pole *Heslo* a *Zopakovat heslo* a změníme jej kliknutím na tlačítko *Upravit*.

Odstranit admina (v horním rolovacím menu je vybrána položka s uživatelským jménem jiného admina než jsem já) – Odstraníme jej kliknutím na tlačítko *Odstranit*. Po ověření, zda-li chcete opravdu admina odstranit, bude odstraněn.

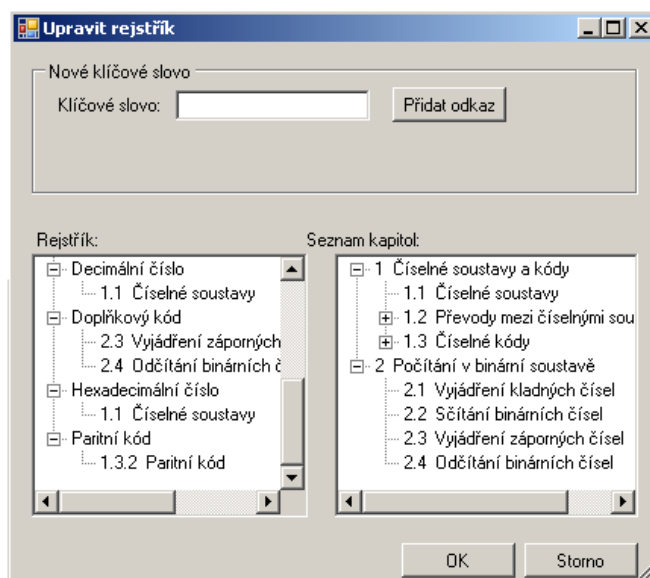
Přidání nového klíčového slova V horním textovém poli formuláře pro úpravu rejstříku můžeme napsat nové slovo a kliknout na tlačítko *Přidat odkaz*, tímto se přidá nové klíčové slovo do rejstříku, kde k němu mohou být přidány odkazy.

Přidání odkazu na existující klíčové slovo Kliknutím na klíčové slovo ve stromové struktuře rejstříku a následným kliknutím na kapitolu, na kterou chceme u vybraného slova odkazovat, vznikne relace, kterou potvrdíme stisknutím tlačítka *Přidat odkaz*. K vybranému slovu bude následně přiřazena vybraná kapitola.

Odebrání existujícího klíčového slova nebo odkazu Odebrání klíčového slova provedeme pomocí kontextového menu, které se zobrazí kliknutím pravého tlačítka myši na klíčové slovo nebo některý z jeho odkazů.

DŮLEŽITÉ je, že změny v rejstříku se uloží až po kliknutí na tlačítko *OK*. Pokud formulář zavřeme nebo stornujeme, námi provedené změny rejstříku budou ztraceny.

3.1.2. Upravit rejstřík



Obrázek 20. Formulář Upravit rejstřík

3.2. Správa testů

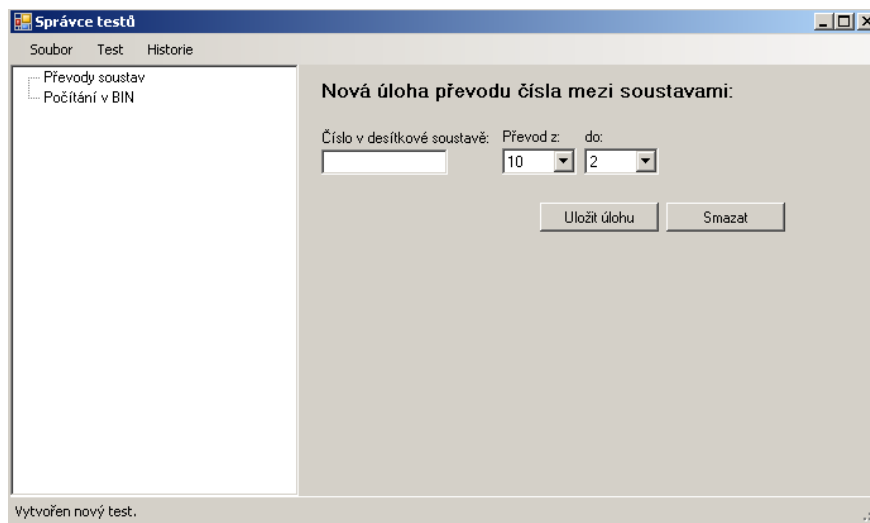
Správa testů se provádí prostřednictvím formuláře, který vyvoláme kliknutím na položku nabídky *Soubor/Správce testů*. Admin má tyto možnosti:

- Vytvořit test a uložit jej – ukládání testu se děje klasicky pomocí položek *Uložit* a *Uložit jako*
- Upravit již vytvořený test – úprava testů pomocí položky *Otevřít test...*
- Zadat test.
- Vyhodnotit odevzdané testy a uložit si jejich výsledky do historie.
- Prohlížet výsledky testů v historii.

3.2.1. Nový test

Nový test vytvoříme pomocí nabídky *Soubor/Nový test...* Na Obr. 21. vidíme, že vlevo v okně je umístěna stromová struktura obsahující na začátku dva prvky: *Převody soustav* a *Počítání v BIN*. Kliknutím na každý z nich se v první části okna objeví editor úlohy pro zvolené téma.

Vytvoříme-li úlohy, zapíše se nám do stromové struktury pod příslušná témata. Jejich výběrem se nám načítá do pravé části okna tentýž panel jako pro novou úlohu, tento je však předvyplněný a slouží k editaci vybrané úlohy. Její úprava je poté provedena po stisknutí tlačítka *Uložit úlohu*. Tato úloha může být zároveň smazána ze seznamu úloh testu tlačítkem *Smazat*.



Obrázek 21. Formulář pro správu testů s vytvořeným novým testem

Úloha z převodů soustav (Viz. Obr. 22.) Do textového pole zapisujeme dekadickou hodnotu čísla, které chceme převádět. Výběr soustav, mezi kterými převádíme, provedeme pomocí rolovacích menu. Náhled příkladu se nám zobrazuje pod textovým polem. Příklad uložíme kliknutím na tlačítko *Uložit úlohu*.

Úloha z počítání v binární soustavě (Viz. Obr. 23.) Zadávání čísel probíhá podobně jako u panelů pro sčítání a odčítání v kapitole 2.3.4. Vedle panelu pro zadávání sčítanců v binární soustavě je zobrazen náhled v soustavě dekadické.

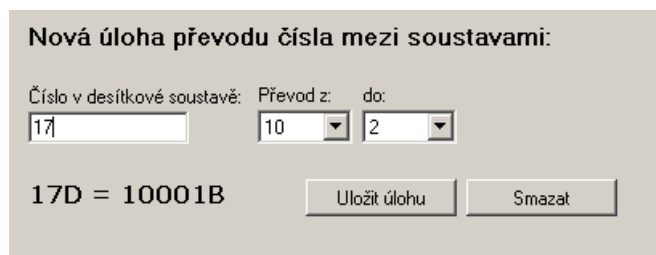
Co je třeba si uvědomit při vytváření úloh na téma počítání v binární soustavě je to, že 8. bit neboli Most Significant Bit [1] označujeme jako bit znaménkový, tedy interval hodnot, které mohou být vyjádřeny takovým číslem, je $< -127, 127 >$ a to platí i pro mezivýsledky operací s množinou těchto čísel. Editor tedy sám kontroluje, zda čísla splňují zadanou podmínku nebo ne. Pokud ji nesplňují, objeví se pod panelem pro zadávání součtu červený nápis se zněním: *POZOR: Vzájemné součty sčítanců musí být z intervalu $< -127, 127 >$.*

Necháme-li jeden ze sčítanců nulový, aplikace jej bude ignorovat a vytvoří úlohu jen ze sčítanců nenulových. Jediný sčítanec značí úlohu pro převod na zápornou reprezentaci čísla.

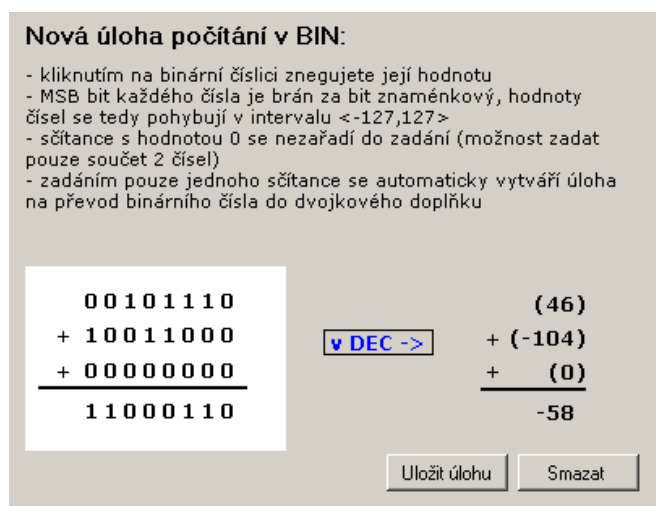
Úlohu ukládáme, podobně jako u převodů, stisknutím tlačítka *Uložit úlohu*.

3.2.2. Správa souborů

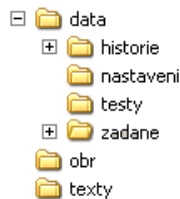
V tuto chvíli je vhodné říct si též něco o správě souborů, které jsou potřebné ke správnému fungování programu a o kterých by měl administrátor alespoň něco



Obrázek 22. Panel pro vytváření a editování úloh převodů mezi soustavami



Obrázek 23. Panel pro vytváření a editování úloh počítání v binární soustavě



Obrázek 24. Soubory příslušné k aplikaci

vědět.

Jedná se o složky na Obr. 24.

Data Tato složka obsahuje dva soubory *admin.txt* a *rejstrik.xml*. První z nich aplikace užívá k přihlašování administrátorů, do druhého zapisuje rejstřík.

nastaveni Zde je umístěn soubor *uc.ini*, ve kterém je nastavená složka zadávání a odevzdávání testů. Do této složky se v kapitole 2.4.1. importuje nastavovací soubor.

testy Složka, která je určena pro ukládání nezadaných testů. Tyto testy však může admin ukládat i jinde bez jakýchkoliv problémů.

zadane Tato složka je nastavená v souboru *uc.ini* při prvním spuštění programu jako složka pro zadávání a odevzdávání testů.

historie Po vyhodnocení testu se program dotáže, má-li test uložit do historie. Jestliže jej admin chce uložit, uloží jej program automaticky právě zde.

Texty Z této složky se načítá obsah teoretické části, jsou zde uloženy texty k jednotlivým kapitolám.

3.2.3. Zadání testu

Na začátku této kapitoly je třeba říct, že i ve formuláři Správa testů může učitel měnit soubor pro nastavení složky zadávání a odevzdávání testů a to kliknutím na položku *Soubor/Nastavení...* Takto zobrazený formulář je podobný jako formulář v kapitole 2.4.1., jen v něm chybí možnost importovat soubor *uc.ini*.

Učitel může zadat test, který před tím vytvořil. Postup zadávání je jednoduchý. Ve formuláři Správce testů načteme již existující test nebo vytvoříme nový a tento uložíme — to je nutná podmínka pro zadání, totiž, že test je již uložený — poté klikneme na položku *Test/Zadat test* a, je-li správně nastavená cesta pro zadání v nastavení (v souboru *uc.ini*), proběhne úspěšné zadání.

The screenshot shows a window titled 'Správce testů' with a menu bar containing 'Soubor', 'Test', and 'Historie'. Below the menu bar, there is a dropdown menu for 'Zobrazený test:' with 'Jan Horáček' selected. The main area has two tabs: 'Celkový přehled' (selected) and 'Prohlížeč testů'. The 'Celkový přehled' tab displays a table with the following data:

	Př. 1	Př. 2	Př. 3	Př. 4	Př. 5	Př. 6	Př. 7
▶ Jan Horáček	Chyba	Chyba	Chyba	OK	Chyba	Chyba	Chyba
Jindra Hojer	OK	OK	OK	OK	OK	OK	Chyba

Obrázek 25. Panel vyhodnoceného testu – celkový přehled

V čem vlastně zadání spočívá? Program vytvoří ve složce k tomu určené kopii zadávaného testu (odtud nutnost test před zadáním uložit) a zároveň složku, kterou pojmenuje stejně jako test. Do této složky pak žáci ukládají vyplněné testy pod svým jménem. Z právě řečeného plyne, že není možné, aby student spustil test, který není předtím zadán učitelem, protože by jej neměl kam odevzdávat.

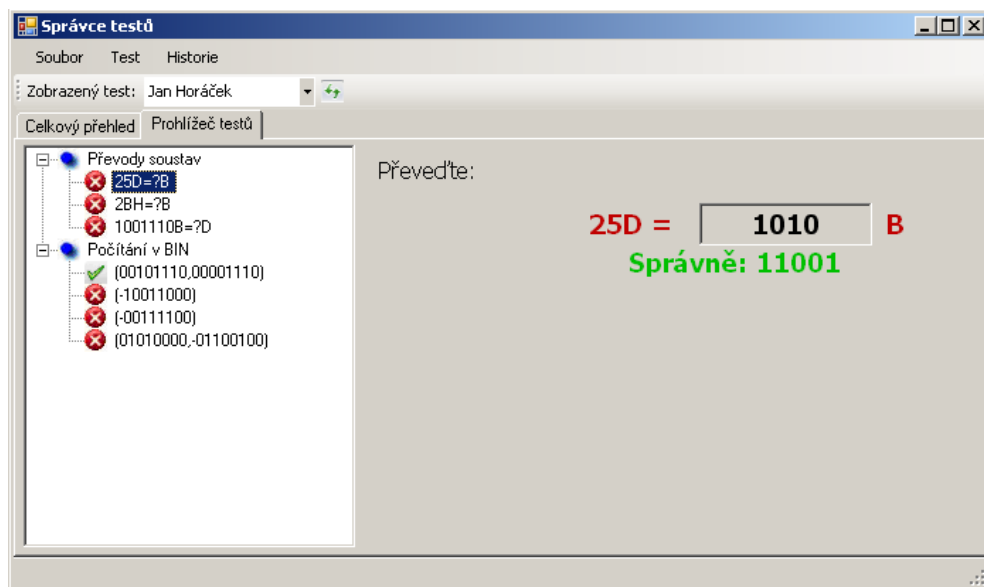
3.2.4. Vyhodnotit test

Možnost vyhodnotit test je v nabídce *Test* kliknutím na položku *Vyhodnotit test...* Po tom, co se objeví formulář *Vyhodnotit test* je třeba vybrat konkrétní ze seznamu zadaných testů. Potom klikneme na tlačítko *Vyhodnotit...*

Nyní si popíšeme panel pro vyhodnocení testů, který se zobrazí ve formuláři *Správa testů*.

Tento panel obsahuje panel nástrojů s rolovacím menu, kde je vypsán seznam studentů, kteří test odevzdali, a tlačítkem pro aktualizaci, pro případ, kdy učitel vyhodnotí test dříve, než jej odevzdají všichni žáci. Pod panelem nástrojů se objeví panel se záložkami *Celkový přehled* a *Prohlížeč testů*. Na první záložce je tabulka, kde je přehledně zobrazena správnost řešení jednotlivých příkladů u každého žáka (Viz. Obr. 25.). Na druhé záložce je možnost prohlížet si konkrétní odevzdané testy. Jednotlivé testy studentů přepínáme pomocí rolovacího seznamu se jmény studentů v panelu nástrojů (Viz. Obr. 26.).

Po zavření pokna, nebo zavření vyhodnoceného testu položkou *Soubor/Zavřít*



Obrázek 26. Panel vyhodnoceného testu – prohlížení testů

test se program zeptá, zda-li chceme vyhodnocený test uložit do historie. Jestliže ano, automaticky jej uloží i se všemi odevzdanými testy do složky historie, v opačném případě vyhodnocený test vymaže ze složky pro zadání.

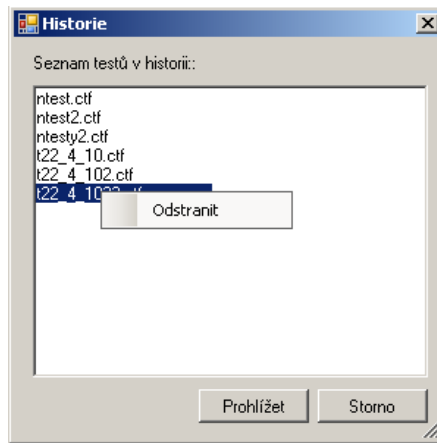
3.2.5. Historie

Historii lze prohlížet na formuláři *Správa testů* kliknutím na položku *Historie/Zobrazit historii...* Ve formuláři *Historie* můžeme test po jeho výběru ze seznamu testů v historii prohlížet nebo pomocí kontextového menu smazat (Viz. Obr. 27.).

3.3. Přidávání kapitol

Jak už bylo řečeno kapitoly jsou načítány ze složky *texty* Viz. 3.2.2.. Kapitoly mohou být přidávány, ale aplikace CIT pro to neposkytuje žádné rozhraní. Stačí text nakopírovat ve správném formátu do složky *texty*. Souboru musí splňovat tyto podmínky:

1. Soubor je ve formátu **.rtf*
2. Jeho název má následující strukturu: „*FFSSTT Název_kapitoly.rtf*“, kde:
 - **FF** – Číslo kapitoly. *Např. 01, 02, ..., 99*



Obrázek 27. Formulář Historie

- **SS** – Číslo první podkapitoly. *Např. 01, 02, ..., 99*
- **TT** – Číslo druhé podkapitoly. *Např. 01, 02, ..., 99*
- **Název_kapitoly** – Název je oddělený od čísla FFSSTT mezerou

Má-li se soubor načíst, musí mimo zmíněných podmínek splňovat to, že má svého předchůdce nebo je první podkapitolou. *Např. 010203 lze přidat, pokud existuje kapitola 010202.*

4. Programátorská část

Program CIT 1.0 byl vytvořen v prostředí Microsoft Visual Studio 2008 vybaveném Microsoft .NET Frameworkem 3.5 SP1 v programovacím jazyce Visual Basic.

4.1. Třída FormCIT

Instance *FormCIT* je hlavním formulářem programu.

- **Sub admin_LogIN** – Obsluha události formuláře admin, která přepíná *FormCIT* do administrátorského módu.
- **Sub NactiStrom** – Načte ze složky *texty* kapitoly do stromové struktury. Pomocí metody *existuje_predchudce* vyfiltruje načítané soubory podle 3.3..
- **Sub nacti_rejstrik** – Ze souboru *rejstrik.xml* načte klíčová slova do stromové struktury.
- **Sub kontrola_souboru** – Kontroluje existenci složek a souborů důležitých pro běh aplikace. Jestliže došlo ke smazání některého z nich, tato metoda jej vytvoří.
- **Sub cmdHledat_Click** – Obsluha události kliknutí na tlačítko hledat. Používá třídu *zaznam*, která reprezentuje text, ve kterém je hledaný výraz nalezen a tuto informaci doplňuje o počet výskytů pro následovné setřídění.
- **Sub nacti_ulohu** – Tato metoda podle vybrané kapitoly nastavuje příslušné nástroje pro zobrazení úloh, zmíníme se o nich dále.
- **Obsluhy událostí AfterSelect** – Tyto dvě metody načítají text k vybrané kapitole do RichTextBoxu ve formuláři FormCIT ve chvíli výběru kapitoly z Obsahu, Rejstříku nebo seznamu vyhledaných kapitol.

4.1.1. Nástroje pro administraci

Modul administrator Obsahuje procedury pro práci se souborem *admin.txt*, kde jsou v zašifrované podobě Uživatelská jména a hesla všech administrátorů.

Třída admin Formulář, který po zadání *uživatelského jména* a *hesla* zkontroluje existenci administrátora v kolekci, kde jsou načtení všichni správci. Jestliže kolekce obsahuje zadaného administrátora, vyvolá formulář událost LogIN.

4.1.2. Nástroje pro cvičení a testování

Každá úloha, kterou zobrazujeme v programu CIT, ať ve cvičení nebo v testu, má svůj základ ve vytvoření instance třídy *prevod*, jedná-li se o úlohu převodů mez soustavami, nebo *BIN_priklad*, jedná-li se o úlohu počítání v binární soustavě.

Enum uroven Tento výčtový typ obsahuje prvky označující úroveň obtížnosti. Jsou to prvky: *lehka*, *stredni*, *tezka*.

Modul prevodove_algoritmy V tomto modulu jsou k dispozici všechny algoritmy potřebné pro převod mezi soustavami i pro vyjádření binárního čísla v dvojkovém doplňku.

Třída prevod Tato třída v sobě nese informaci o převáděném čísle, soustavách původní a cílové, správné odpovědi a úrovni obtížnosti. Podle parametru, který dostane konstruktor této třídy *New(...)* se instance vytvoří způsobem:

- Program vygeneruje náhodně číslo i původní a cílovou soustavu.
- Program vygeneruje náhodně číslo, původní a cílová soustava jsou dány v parametrech.
- Číslo i soustavy jsou dány v parametrech.
- Úloha je načtena ze souboru **.ctf*.

Ve všech případech si převod pomocí převodových algoritmů vypočítá správnou odpověď.

Enum typ_prikladu Tento výčtový typ je použit pro identifikaci typu úlohy při počítání v binární soustavě. Obsahuje tyto prvky: *scitani*, *zaporne_cislo*, *odcitani*.

Třída Scitacka Třída zprostředkovávající součet dvou binárních čísel.

Třída BIN_priklad Podobně jako třída *prevod* u úloh převodů identifikuje úlohu, tak tato třída identifikuje úlohu u počítání v binární soustavě. Nese v sobě informaci o všech sčítancích, jejich znaménkách, správné odpovědi a úrovni obtížnosti. Podle parametru, který dostane konstruktor této třídy *New(...)* se instance vytvoří způsobem:

- Program vygeneruje náhodně typ příkladu a k němu sčítance.
- Program dostane informaci o typu příkladu a k němu vygeneruje sčítance.

- Program dostane seznam sčítanců, z nich pak rozpozná o jaký typ příkladu jde.
- Úloha je načtena ze souboru **.ctf*.

Ve všech případech si *BIN_příklad* pomocí převodových algoritmů a třídy sčítačka vypočítá správnou odpověď.

Třídy průvodců Když nyní známe význam tříd *prevod* a *BIN_příklad*, můžeme se krátce zmínit o tom, co jsou to třídy průvodců. Těchto tříd máme několik:

3. Průvodce pro převody mezi soustavami

- **Třída DEXx** – Pro převod z dekadické soustavy do binární nebo hexadecimální.
- **Třída xDEC** – Pro převod binární nebo hexadecimální soustavy do dekadické.
- **Třída BINHEX** – Pro převod binární soustavy do hexadecimální a naopak.

4. Průvodce počítáním v binární soustavě

- **Třída pocty_BIN**

Představu o jejich funkcionalitě si můžeme udělat v kapitolách 2.3.1.—2.3.6.. Containersy těchto tříd jsou komponenty *gui_prevod* a *gui_scitani*, které podle aktuální instance třídy *prevod* resp. *BIN_příklad* průvodce vytváří a nastavují.

Třídy nápovědy Jsou komponenty obsahující RichTextBox a metody umožňující zápis do něj. Ve spolupráci s modulem *prevodove_algoritmy* vepisují do RichTextBoxu postupy při zadání konkrétních příkladů.

Třída gui_prevod Komponenta, která zprostředkovává řešení úlohy převodu mezi soustavami. Zobrazuje v sobě zadání, průvodce a nápovědu. Pro lepší představu o této třídě nahlédněte do kapitoly 2.3.. Veškeré nastavení této komponenty je dáno instancí třídy *prevod*, podle níž nastavuje zadání, volí vhodného průvodce a nápovědu. Změnou úlohy na jinou úlohu převodu se tato komponenta přenastaví vytvořením nové instance třídy *prevod*. Tato změna je způsobena buď tlačítkem *Další příklad*, které tato komponenta obsahuje, nebo voláním metody *nacti_ulohu*, kterou volá její Container, FormCIT.

Třída `gui_scitani` Ekvivalentní komponenta s `gui_prevod`, akorát pro počty v binární soustavě. Její nastavovací třída je *`BIN_priklad`*.

Obě naposled popsané třídy jsou v programu použity ve třech režimech:

- pro cvičení – s aktivním panelem nástrojů s tlačítkem pro přechod k dalšímu příkladu a volbou obtížnosti, podle které se nastavuje chování průvodce i zobrazování nápovědy
- pro testování – panel nástrojů je deaktivovaný, automaticky nastavená těžká obtížnost průvodce a deaktivovaná možnost nápovědy
- pro prohlížení odevzdaných testů ve vyhodnocování – zobrazí jen zadání s načtenou odpovědí studenta při řešení testu, deaktivovaný panel nástrojů, průvodce i nápověda, aktivované zobrazení správné odpovědi v případě chybné odpovědi studenta pod textovým polem pro odpověď

4.2. Třída `Form_sprava`

Její instance je formulář pro správu testů, vyvolaný kliknutím na položku *`Soubor/Spravce testů...`*. Tento formulář slouží k vytváření, prohlížení a vyhodnocování testů. Obsahuje dialogy pro otevření a ukládání testů.

- **Event `ukonceni_testu`** – Tato událost se vyvolá ve chvíli, kdy se vykoná metoda, která zavírá stávající načtený test.
- **Event `dotaz_na_ulozeni`** – Tato událost se vyvolá ve chvíli, kdy chce učitel zadat test, který není uložený nebo je editovaný.
- **Sub `Form_sprava_ukonceni_testu(...)`** – Obsluha události *`ukonceni_testu`*, která zjišťuje, zda je test editovaný. Pokud editován je, vyvolá dotaz na uložení.
- **Sub `ulozit_test(...)`** – Tato metoda vytvoří ve složce zadané pomocí `SaveFileDialogu` soubor `*.ctf` a uloží do něj vytvořený test.

4.2.1. Vytváření a editování testů

Třída `editor_prevodu` Tato komponenta umožňuje učiteli vytvářet úlohy na převod mezi číselnými soustavami, konkrétně: vytvářet instance třídy *`prevod`*, které poté předává svému Containeru, jímž je třída *`novy_test`*. Z nového testu poté umí také úlohy načítat a odstraňovat.

Třída `editor_poctu` Tato komponenta umožňuje učiteli vytvářet úlohy pro počítání v binární soustavě, respektive instance třídy *`BIN_priklad`*, které poté předává svému Containeru, jímž je třída *`novy_test`*. Z nového testu poté umí také úlohy načítat a odstraňovat.

Třída `novy_test` Tato komponenta obsahuje seznamy instancí tříd `prevod` a `BIN_prikklad`, které jsou přidávány metodou `pridat_ulohu(...)` od výše popsaných editorů úloh. Pro editaci úloh slouží metoda `edit(...)`, kterou třída `novy_test` od editorů přijímá změny načtených úloh. Seznamy úloh jsou umístěny ve stromové struktuře, kterou tato třída též obsahuje.

4.2.2. Spouštění a vyhodnocování testů

Třída `nacteny_test` Komponenta podobná třídě `novy_test`. Používá se při načítání testů ze souborů `*.ctf`. Obahuje stromovou strukturu s načtenými úlohami a třídy `gui_prevod` a `gui_scitani` v režimu testování nebo vyhodnocování pro každou úlohu. Tato komponenta bývá načítána do formuláře pro spuštěný test a do třídy `vyhodnoceny_test`.

Třída `vyhodnoceny_test` Tato třída přijímá ve svém konstruktoru cestu k zadanému testu i adresáři pro odevzdávání vyplněných textů. V prvním kroku si načte všechny odevzdané testy, v druhém pak z jednotlivých úloh vytvoří instance tříd `prevod` a `BIN_prikklad` přičemž ke každé této třídě přidá ještě výsledek, který v testu uvedl student. Nakonec porovná správné odpovědi s odpověďmi studenta a vygeneruje tabulku (`DataGridView`) s celkovým přehledem odpovědí žáka. Zároveň ještě vytvoří ke každému testu instanci třídy `nacteny_test` v režimu vyhodnocování, která umožňuje prohlížet všechny odevzdané testy.

- **Sub `aktualizovat(...)`** – Metoda pro znovunačtení všech testů do třídy `vyhodnoceny_test`.
- **Sub `smazat_zadane(...)`** – Metoda pro odstranění vyhodnoceného testu ze seznamu zadaných testů. Před vyvoláním této metody je vyvolán dotaz na uložení do historie.
- **Sub `ulozit_do_historie(...)`** – Metoda, která zkopíruje zadaný test i všechny odevzdané ze složky pro zadání testů do složky historie.

4.3. Třída `Form_test`

Formulář pro řešení testu zadaného učitelem. Tento formulář obsahuje komponentu `nacteny_test` v režimu testování.

4.4. Třída `Form_historie`

Formulář pro zobrazení testů uložených v historii. Načte testy uložené ve složce `data/historie`. Pomocí kontextového menu se tyto načtené testy dají ze složky odstraňovat a nebo pomocí třídy `vyhodnoceny_test` prohlížet.

4.5. Soubory typu *.ctf (CIT test file)

Soubory s příponou **.ctf* používá CIT pro ukládání testů. Do těchto souborů aplikace CIT zapisuje v kódu XML.

5. Závěr

Program CIT 1.0 byl vytvořen pro podporu výuky na VOŠ a SPŠ Elektrotechnické Božetěchova 3, Olomouc. Jsou zde použité materiály vytvořené učiteli předmětu Číslicová technika této školy. CIT 1.0 má sloužit zejména k zafixování postupů při řešení daných témat bohatými možnostmi procvičování. Dále program poskytuje učiteli nástroj pro testování znalostí žáků v hodinách. Po dohodě s učiteli je možnost program rozšířit i na jiná témata.

6. Conclusions

The CIT 1.0 programme was created for the teaching support at secondary school VOŠ a SPŠ Elektrotechnické Božetěchova 3, Olomouc. The materials to this programme, which were created by the numerical technology teachers of this school are used here. CIT 1.0 should serve mainly to fix the processes when solving the given topics through a variety of exercises. Furthermore, the programme gives the teacher a tool for testing the students' knowledge in the lessons. After agreement with the teacher, it is possible to extend the programme also on other topics.

Reference

- [1] VESELÁ, Z., KYSELÝ, L.: Číslicová technika, díl I., Olomouc, 2006.
- [2] PETROUTSOS, E.: Myslíme v jazyku Visual Basic .NET, díl I., Grada, 2003, Praha, ISBN 80-247-0371-8.
- [3] PETROUTSOS, E.: Myslíme v jazyku Visual Basic .NET, díl II., Grada, 2003, Praha, ISBN 80-247-0372-6.
- [4] Microsoft Developer Network [online], 2010,
Dostupné z WWW: <<http://www.msdn.com>>.
- [5] Visual Basic .NET [online], 2010,
Dostupné z WWW: <<http://www.vbnet.com>>.
- [6] Aplikace JHmatika [online], 2009,
Dostupné z WWW: <<http://www.slunecnice.cz/sw/matika/>>.
- [7] Aplikace emu8086 [online], 2007,
Dostupné z WWW: <<http://www.emu8086.com/>>.
- [8] Aplikace Binary Adder 1 [online], 2010,
Dostupné z WWW: <http://www.softlist.net/program/binary_adder-image.html>.