

# SYSTEM OF PREPARING HOMEWORKS

Martin Tuček, Master Degree Programme (5)  
Faculty of Information Technology, BUT  
E-mail: xtucek04@stud.fit.vutbr.cz

Supervised by: Ing. Bohuslav Křena, Ph.D.

## ABSTRACT

This paper deals with reimplementation of the system for exercising basic data structures and algorithms among them. This system is used within the course Algorithms which takes place in the 3rd term of the bachelor study programme Information Technology at FIT BUT. The system has used Pascal as an implementation language while current needs forced the teachers to use programming language C. We will discuss the process of this transformation in this paper.

## 1 ÚVOD

Studenti druhého ročníku Fakulty informačních technologií v Brně, konkrétně předmětu Algoritmy (IAL), jehož náplní je seznámit studenty se základními datovými strukturami a algoritmy [1] nad nimi, po dlouhá léta odevzdávali projekty do tohoto předmětu v syntaxi jazyka PASCAL [1]. Bylo rozhodnuto, že se předmět IAL zásadně změní a to přechodem výuky veškeré syntaxe uvedené v tomto předmětu do programovacího jazyku C, tzn. včetně odevzdávaných úloh. Vzhledem k tomu, že systém pro odevzdávání domácích úloh byl pochopitelně naprogramovaný kompletně v PASCALU, bylo třeba předělat tento systém včetně zadávaných úloh do jazyku C. Na tomto systému jsem pracoval již ve čtvrtém ročníku (jako ročníkový projekt) a navázal na něj v pátém ročníku diplomovou prací. Tento rok jsem na něm již nepracoval sám, přidal se student Václav Topinka (viz kapitola 2.1).

## 2 STRUKTURA VYTVÁŘENÉHO SYSTÉMU

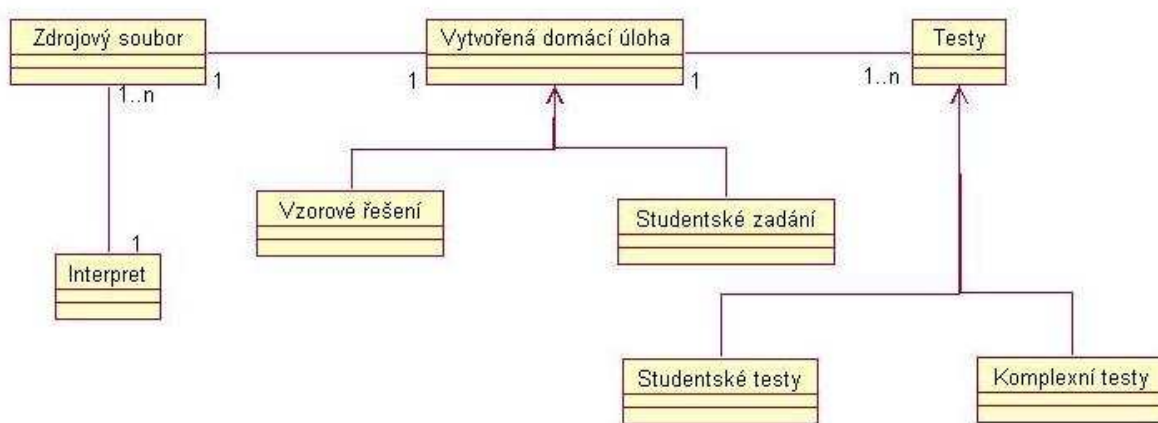
Vytvářený systém se skládá z několika samostatných souborů. Jejich základem je zdrojový soubor obsahující surová data (tj. zadávané příklady včetně vzorových řešení) a metaznaků. Na tento soubor je použit interpret. Ten na základě detekovaných meta znaků generuje z daného zdrojového souboru 2 verze úloh.

- první verzi je verze pro studenty, ta obsahuje pouze zadání, což jsou hlavičky funkcí, které je třeba vypracovat. Student doplňuje tyto procedury či funkce, tak aby fungovaly podle zadání. To je specifikováno jako komentář v úvodu každé úlohy

- druhou verzí je ta včetně řešení – obsahuje nejen hlavičky funkcí, ale i vzorová řešení. Ta se používá pro vygenerování správných výsledků, které se pak srovnávají s výsledky dosažené studenty

Každá úloha obsahuje sadu dvou testovacích souborů. Jeden test je určený pro studenty, aby si mohli zkontrolovat výsledek své práce, druhý test pro opravovatele těchto úloh. Tento test je komplexnější, obsahuje více testovacích procedur volaných v různém pořadí, včetně různých nestandardních stavů, které je třeba pochopitelně rovněž vyhodnocovat a očekávat.

Interpret jsem implementoval jako konečný automat, který podle pozice v souboru a příchozích meta značích generuje buď ono vzorové či studentské zadání. Vše dokumentuje obrázek 1 zpracovaný v notaci UML (Rational Rose), jehož komentář následuje.



**Obr. 1:** *System zadavani domácích úloh*

V praxi celý systém funguje následujícím způsobem. Nejprve se vygeneruje pomocí interpretu domácí úkol z připravených zdrojových souborů (viz obrázek 2).

```

void InitList (tList *L)  {
  /*v*/
  L->act = NULL;
  L->fst = NULL;
  /*v*/
  // solved = FALSE; /*v případě řešení, smažte tento řádek!*/
}
  
```

**Obr. 2:** *Ukázka zdrojového souboru*

Tento vygenerovaný domácí úkol může být buď se vzorovým řešením nebo se studentským zadáním. To zpracuje student. Po té proběhnou testy (jejichž podoba je vidět na obrázku 3), tj. zavolají se příslušné testy pro konkrétní úlohy s parametrem vytvořených úloh.

```
int test_LCopy()    {
    solved=TRUE;
    return(use_copy(&TEMPLIST,&obsah));
}
```

**Obr. 3:** Ukázka z podoby testovacích souborů

Porovnájí se výsledky oproti těm, které dostaneme, když daný test voláme na úlohu se vzorovým řešením.

## 2.1 DOPLNĚNÍ SYSTÉMU

Na systému pracoval i student Václav Topinka (rovněž jako součást diplomové práce), který jej doplnil o další funkce:

- makefile [2], který určuje pořadí generování jednotlivých souborů,
- plně automatizované hodnotící skripty, které testují dané soubory (ošetřeny timeouty atd.) a vytvářejí příslušné adresáře pro každého studenta s výsledky testů včetně výpisů chyb,
- skript na odhalování plagiátorů.

## 3 VYPRACOVANÉ ÚLOHY

Přepočoval jsem konkrétně 6 úloh – lineární jednosměrný seznam, kruhový seznam, dvousměrný lineární seznam, binární strom rekurzivní, binární strom nerekurzivní a vyhledávací tabulka v neseřazeném poli [1]. Václav Topinka přetvořil hash table a práci se zásobníky. Celý systém byl již zprovozněn a to zadáním první sady úloh v říjnu tohoto roku. Vše proběhlo uspokojivě, včetně ohodnocení. Objevil se problém s kopírováním úloh, ale míra plagiátorství byla na předpokládané úrovni. S vydáním druhé sady úloh (listopad 2005) rovněž nebyly žádné zásadní problémy. V současné době se soustředím na napsání dokumentace k tomuto systému.

## LITERATURA

- [1] Honzík, J. M., Hruška, T., Máčel, M.: Vybrané kapitoly z programovacích technik. Třetí redukované vydání. Vysoké učení technické v Brně, 1991. s. 218. ISBN 80-214-0345-4.
- [2] Herout, P.: Učebnice jazyka C. Třetí upravené vydání. Kopp, České Budějovice, 1998. s. 272. ISBN 80-85828-21-9.