

LAMSFET - LONGTERM AND MULTIPURPOSE STORAGE FOR EXERCISE TASKS

BSC. PETER JURČO, MGR. PAVEL PETROVIČ, PHD

Katedra aplikovanej informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Bratislava,
e-mail: peter.jurco@st.fmph.uniba.sk, ppetrovic@acm.org

ABSTRAKT

Študenti bakalárskeho štúdia Aplikovaná informatika na FMFI UK absolvujú niekoľko kurzov programovania. V kurzoch študenti dostávajú povinné domáce úlohy, riešia úlohy na cvičeniach, rozcvičky, testy pri počítači, skúšky, dostávajú projektové zadania. Vytvorili sme open-source nástroj Lamsfet, ktorý je určený na správu takýchto kurzov a funguje predovšetkým ako dlhodobý archív úloh zadanych v minulosti, ktoré su označené kľúčovými slovami zaradenými vo viacerých hierarchiách. Nástroj umožňuje správu skupín študentov, ich riešení i bodov. Podporuje polo-automatizované opravovanie riešení i zadanie spätnej väzby od učiteľov smerom k študentom. Na rozdiel od bežných a univerzálnych nástrojov ako Moodle ponúka špecializovanú alternatívu zameranú na konkrétnu potrebu, vďaka čomu poskytuje efektívnejšiu prácu, jednoduchšie a prehľadnejšie použitie pre učiteľov i študentov. Nástroj sme úspešne overili v priebehu jedného celého semestra na predmete Programovanie III - Algoritmy v Jave, v súčasnosti obsahuje okolo 370 zadaní úloh a využívame ho v ďalšom kurze. V článku uvádzame naše i študentské skúsenosti s použitím nástroja a opisujeme jeho funkcionálnosť.

Kľúčové slová: *learning management system, archív úloh, programovanie*

1. ÚVOD

Zvyšovanie podielu populácie, ktorá študuje na vysokých školách, prináša zvýšené množstvo práce s prípravou výuky, na čo pedagóg potrebuje dostatok v praxi overených materiálov. V prípade predmetov, ktorých obsahom je programovanie, sú to predovšetkým zadania domácich úloh, príklady na cvičenia a testy. Pedagóg by nemal opakovať rovnaké sady úloh z roka na rok. Motivoval by tým časť študentov k hľadaniu „alternatívnych“ spôsobov riešenia využitím svojich starších kolegov. Navyše, pedagóg by sa mal prispôbiť úrovni a špecifikám daného ročníka a materiál obmieňať a podľa možnosti inovovať. Druhý z autorov po absolvovaní dvoch semestrov v úlohe asistenta, podieľajúc sa na príprave zadaní, pochopil, že jeho práca a práca ostatných kolegov, ktorí pripravujú obsah cvičení, by mohla byť využitá nad rámec jediného ročníka a jediného predmetu. Efektívna recyklácia by sa dala dosiahnuť, ak by sme mali k dispozícii nástroj na archiváciu pripravených úloh. Nástroj by mal umožniť efektívne vyhľadávať podľa rozličných kritérií a vybrané úlohy kombinovať do zadaní úloh. Pripravené materiály sa nemusia obmedzovať na zadania úloh, ale mali by umožniť vkladať obrázky, dátové súbory, príklady testovacích vstupov, vzorové riešenia. Ideálne by systém mohol evidovať ako a kedy boli jednotlivé úlohy vo výuke použité, aby pedagóg mohol zväziť, či sa úloha neopakuje priveľmi často. Ak by systém zároveň vytvorené zostavy priamo zverejňoval pre študentov, ktorí sú na predmet zapísaní, cvičiacim by odpadla potreba administratívy, spojená so zverejňovaním zadaní. Ak by systém navyše študentom umožnil odoslať svoje riešenia a cvičiacim by umožnil ich poloautomatické hodnotenie, získali by sme univerzálny nástroj na zadávanie, odovzdávanie a opravovanie domácich úloh, písomiek, príkladov na cvičenia.

Na základe uvedenej predstavy bola vypísaná téma bakalárskej práce, ktorú si prvý z autorov zvolil, vypracoval špecifikáciu a návrh, podľa ktorého systém v základnej podobe implementoval a neskôr prácu úspešne obhájil [2]. Nasledovalo rozšírenie systému tak, aby maximálne vyhovoval požiadavkám praxe. Nasadenie v praxi umožnilo spriechodniť a zlepšiť navigáciu aj funkcionálnosť a doplniť ďalšie komponenty, ktoré neboli pri špecifikácii známe – napríklad výber projektov, automatické strhávanie bodov za neodovzdané úlohy a pod.

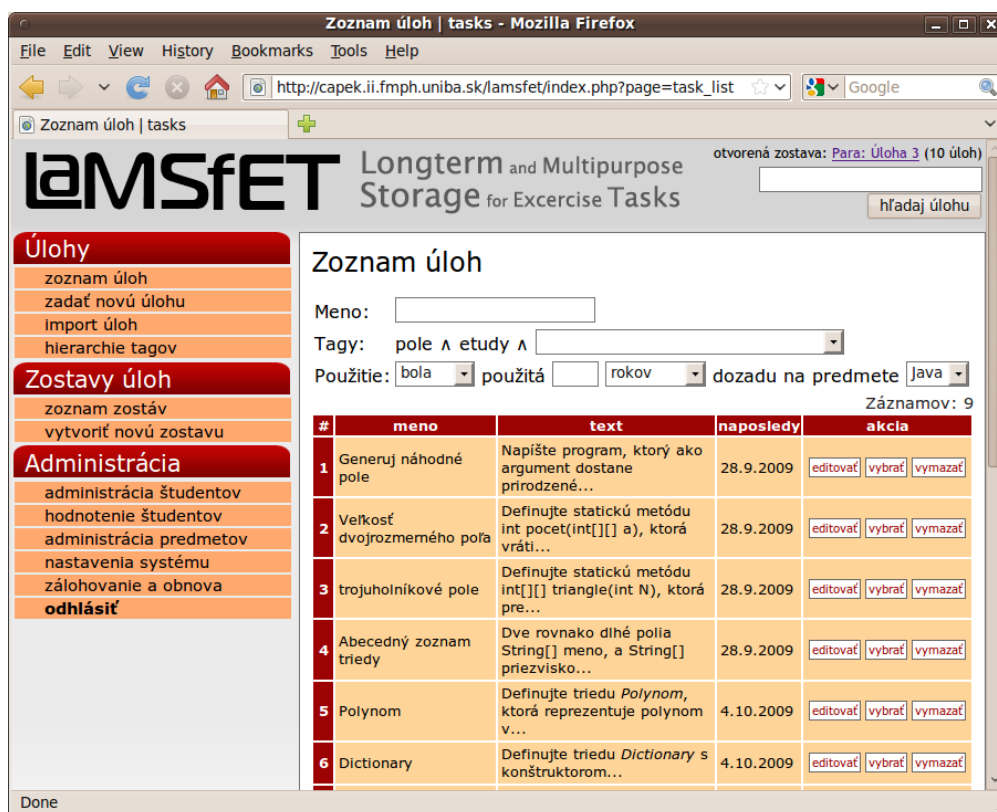
Dnes je doba zavádzania rôznych informačných systémov podporujúcich výuku. Od rôznych akademických informačných systémov na správu štúdijských programov, rozvrhov a študentov, cez systémy LMS (Learning Management System), ktoré dovoľujú vytvárať on-line diaľkové kurzy a on-line doplnujúce prostredia, ktoré podporujú bežné kurzy dennej výuky. Aj náš systém sa zaraďuje do tejto skupiny, odlišuje sa však tým, že nemá veľmi široký záber a univerzálne použitie, ako napr. známy systém Moodle. Naopak, snaží sa ponúknuť špecializované riešenie pre istý druh kurzov a jednoduché a prirodzené používateľské rozhranie, ktoré sa študenti ani pedagógovia nemusia učiť, ale môžu ho využívať priamo a okamžite. Rozhranie učiteľa je na obr. 1.

Kľúčovými vlastnosťami aktuálnych informačných systémov, ktoré podporujú ukladanie dát, sú také formáty, ktoré umožňujú ich dlhodobú prístupnosť, čitateľnosť a využiteľnosť [1]. V nasledujúcich stadiách

opíšeme filozofiu systému, viacnásobnú hierarchiu pre klasifikáciu úloh, užívateľské scenáre, ďalšie funkcie systému, skúsenosti s využitím systému a v závere uvedieme našu predstavu o ďalšom využití a prípadných vylepšeniach systému.

2. FILOZOFIA SYSTÉMU

Systém Lamsfet sa inštaluje na počítač pripojený na sieť Internet, na ktorom beží webserver (napr. Apache2) s podporou PHP skriptov a server PostgreSQL s vytvorenou databázou. Používatelia k systému prístupujú zásadne cez internetový prehliadač (IE7, Mozilla, Opera, a pod.), cez ktorý vykonávajú všetky operácie. Používatelia sú jednak učители, ktorí zdieľajú jeden spoločný „administrátorský“ používateľský účet a jednak študenti, ktorí majú vytvorené svoje vlastné účty. Používateľské rozhranie administrátorov a študentov je celkom odlišné: administrátori spravujú a vytvárajú úlohy a ich zostavy, opravujú a bodujú riešenia, zatiaľ čo študenti prezerajú zadania, ktoré sú pre nich určené, odovzdávajú svoje riešenia a prezerajú si hodnotenie svojich riešení.



Obr. 1 Používateľské rozhranie systému Lamsfet – hlavné menu administrátora/učiteľa.

Systém integruje nasledujúce koncepty:

Úloha – je základnou jednotkou v systéme, pozostáva z jedného zadania konkrétnej úlohy špecifikovaného ako čistý text alebo v jazyku HTML. Úloha môže obsahovať obrázky a doplnujúce dátové súbory. Úloha je označená skupinou kľúčových slov. K úlohe môže byť priradené jedno alebo viacero vzorových riešení a testovacích vstupov a očakávaných výstupov. Úloha je nezávislá od predmetov a zostáv, ale o každej úlohe sa eviduje kedy a v akých zostavách, teda aj na akých predmetoch bola použitá. Príklad úlohy je na obr. 2.

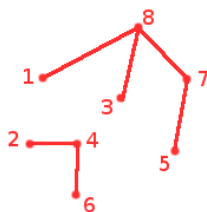
Zostava úloh – predstavuje spôsob, akým sa úlohy zoskupujú do jednotlivých zadaní a zverejňujú pre študentov. Okrem samotných úloh zostava obsahuje spoločný textový komentár. Aj každá úloha v zostave môže mať svoj komentár a priradený počet bodov, či názov súboru, v ktorom sa má odovzdať, ak je to potrebné. Zostavy sú priradené ku konkrétnym kurzom (t.j. semestro-predmetom). Existujú zostavy rôznych typov, napr. povinné domáce úlohy, rozcvičky, cvičenia s dobrovoľne riešenými príkladmi, testy riešené pri počítači a pod. Typy zostáv je možné konfigurovať pre každý predmet zvlášť. Zostavy môžu a nemusia vyžadovať od študentov odovzdávanie riešení. Je možné nastaviť presné obdobie, kedy má byť zostava zverejnená a dokedy je možné odovzdať riešenia. Zostavy je možné klonovať. Nastavenie zostavy je na obr.3.

Predmet – združuje skupinu zostáv, ktoré sa týkajú jedného predmetu. Predmet je ďalej rozdelený na jednotlivé semestre, v ktorých sa vyučoval.

Študent – je používateľ systému, ktorý má zároveň vytvorený používateľský účet. Študent môže byť zaradený na jeden alebo viacero predmetov, prípadne si ich sám vybrať (jeho výber potvrdí administrátor). Študent vidí zoznam svojich zadaní, ktoré sú podľa ich nastavení v tom čase prístupné, môže odovzdať svoje riešenia.

1. Grafová postupnosť editovať x

Grafová postupnosť je postupnosť stupňov vrcholov grafu. Napríklad vrcholy v grafe na obrázku majú stupne postupne 1,1,1,2,1,1,2,3. Napíšte program, ktorý zo súboru načíta grafovú postupnosť pre graf s N vrcholmi (stupne sú oddelené čiarkami a riadok je ukončený bodkou) a do výstupného súboru vypíše, či taký graf existuje. Bodovanie: menej ako kvadratická časová zložitosť: 3 body, polynomiálna: 2 body, horšia: 1 bod.



Príklady:

1,1,1,2,1,1,2,3.
graf existuje

1,1,2,2,2,2,2,3.
graf neexistuje

6,1,4,2,3,1,4.
graf neexistuje

6,6,5,5,5,2,1,1,1.
graf neexistuje

Obr. 2 Príklad jednej úlohy zadanej v systéme.

Skupina študentov – študenti prihlásení na konkrétny predmet sú združení do štúdijských skupín. Skupiny majú určený čas a miesto (učebňu) cvičení. Niektoré zostavy úloh môžu byť určené len pre vybranú skupinu študentov, iné pre všetkých študentov, ktorí sú prihlásení na predmet. Skupiny majú nastavenú maximálnu kapacitu a študenti si po začiatočnom rozdelení môžu sami zvoliť skupinu, ktorú chcú navštevovať, podľa toho, ako im to vyhovuje vzhľadom na ich rozvrh a ak to kapacita skupiny ešte dovoľuje.

Študentské riešenie (submit) – k jednotlivým zostavám môžu študenti odovzdávať svoje riešenia formou automatického „upload“-u do systému. V prípade potreby môžu odovzdať aj opravné, alebo doplnujúce submity. Ku každému submitu sa eviduje čas odovzdania i IP adresa, odkiaľ bol súbor nahratý. Študenti majú ku svojim submitom prístup a môžu si ich spätne „download“-núť, nemôžu ich však zmazať. Učitelia majú možnosť zobrazit' všetky submity k zvolenej zostave príp. v danej skupine.

Hodnotenie riešenia – každý submit môže mať priradené svoje hodnotenie zadané učiteľom. Hodnotenie pozostáva z počtu získaných bodov a prípadne slovného komentáru – spätnej väzby pre študentov.

Projektové a verejné úlohy a zostavy úloh – Okrem základných typov zostáv, ktoré sú viditeľné až po prihlásení sa do systému, je možné vytvárať aj verejné zostavy, teda také, ktoré je možné zverejniť prostredníctvom ich URL mimo systém Lamsfet. Okrem toho systém dovoľuje režim, v ktorom si študenti vyberajú jednu z určitej množiny úloh. Pre danú zostavu je určený počet študentov, ktorí si môžu zvoliť rovnakú úlohu (napr. najviac 3 študenti). Svoj výber môžu aj viackrát zmeniť a priradenie študentov k zostavám je tiež možné zverejniť mimo systém, ak je to potrebné.

Úloha 3 (Editácia zostavy)

nastavenia zostavy
úlohy

Predmet: Para 2009/2010

Typ zostavy: domáca úloha [upravíť typy zostáv pre Para 2009/2010](#)

Názov: Úloha 3

Len pre skupinu: [vzdĺknutý výber]

Zverejniť: 2.3.2010 17:00

Prestať zverejňovať: nikdy 12:00

Odovzdať do: 9.3.2010 15:00

Komentár:

vyriešte VŠETKY úlohy tejto sady

ulož
ulož ako novú zostavu

Obr. 3 Nastavenia jednej zostavy úloh.

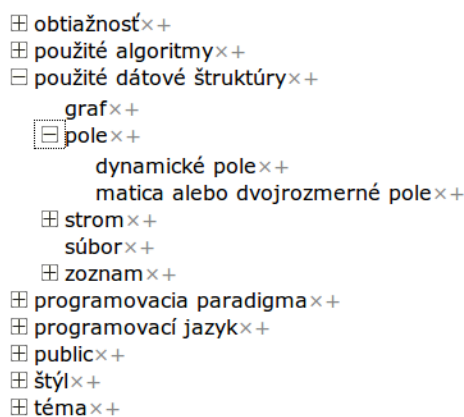
Systém Lamsfet je vyvinutý tak, aby bol pre používateľa dobre prístupný a prirodzene zvládnuteľný. Konfiguruje sa priamo cez webový interfejs v časti nastavení systému. Je koncipovaný tak, aby zásahy systémového správcu prakticky neboli vôbec potrebné. Užívateľské rozhranie je naprogramované v JavaScripte využitím frameworku Smarty, ktorý dovoľuje ergonomický design a grafické a riadiace prvky na profesionálnej úrovni.

Súčasťou našej filozofie je, že učiteľ by sa pri hodnotení riešení študentov mal dívať do zdrojových kódov. Hodnotenie študenta by nemalo byť iba čierno-biele: funguje/nefunguje, ale kvalitatívne. Z toho vyplýva poloautomatický spôsob hodnotenia riešení, ktorý náš systém poskytuje. Učiteľ si môže pozrieť jednotlivé submity priamo cez svoj prehliadač v okne. Tieto riešenia môže na serveri v kontrolovanom prostredí priamo spúšťať a automaticky porovnávať výstup odovzdaného programu s očakávaným vzorovým výstupom, ktorý bol priradený k zadaniu úlohy, alebo si výstup programu len jednoducho prezerat'. Má navyše možnosť v okne, kde je zdrojový kód zobrazený, opakovane vykonať zmeny a takto zmenený program znova spustiť. V tom istom dialógu pridáva body a zapisuje slovný komentár, ktorý je súčasťou hodnotenia.

3. VIACNÁSOBNÁ HIERARCHIA PRE KLASIFIKÁCIU ÚLOH

Typickým riešením zaužívaným v praxi na klasifikáciu dokumentov alebo iných zdrojov sú kľúčové slová. Tento spôsob sme si osvojili aj v systéme Lamsfet, avšak s istým zlepšením: kľúčové slová sú zaradené v hierarchii – a to nie jednej, ale vo viacerých hierarchiách, ktoré sú organizované podľa rôznych uhlov pohľadu. Každé kľúčové slovo môže mať svoje špecializácie a samotné môže byť špecializáciou nejakého iného, všeobecnejšieho kľúčového slova, obr. 4. Učitelia môžu vytvárať nové hierarchie, alebo upravovať existujúce hierarchie a podľa potreby ich rozširovať. Kľúčové slová sa priradujú k jednotlivým úlohám – systém po zadaní každej úlohy automaticky prejde na stránku, kde prebieha výber „tag“-ov. Kľúčové slová je možné priradiť aj dodatočne kedykoľvek neskôr. Napríklad, v našej aktuálnej inštalácii teda úlohy označujeme kľúčovými slovami rozdeľujeme podľa obtiažnosti, použitých algoritmov, dátových štruktúr, vhodnej programovacej paradigmy, štýlu, témy a iných aspektov.

Hierarchie tagov



Obr. 4 Hierarchie kľúčových slov priradovaných k jednotlivým úlohám.

Kľúčové slová sa využívajú pri hľadaní/prezeraní úloh v systéme. Množinu zobrazených úloh v zozname je možné zúžiť na také, čo spĺňajú podmienku zadanú v tvare konjunktívnej normálnej formy (CNF), napr. „matematická & (pole | rekúzia)“ zobrazí príklady, ktoré majú matematickú tému a buď používajú dátovú štruktúru pole, alebo je v nich vhodné použiť rekurzívny algoritmus. Keďže kľúčové slovo pole má ešte rôzne pod-slová, zoznam zobrazených úloh bude obsahovať jednak úlohy priamo s kľúčovým slovom pole a jednak úlohy, ktorým sú priradené ľubovoľné kľúčové slová hlbšie v hierarchii pod slovom pole, napr. dynamické pole, matica alebo dvojrozmerné pole. Samotné termy v CNF netreba vypísať, stačí si ich zvoliť z ponuky existujúcich slov.

4. POUŽÍVATELSKÉ SCENÁRE

V tejto časti opíšeme typické situácie použitia pre učiteľov a študentov.

Vytvorenie nového predmetu – učiteľ sa prihlási do systému cez prihlasovací formulár, v časti Administrácia predmetov buď pridá nový semester k už existujúcemu predmetu, alebo zadá plný názov a skratku nového predmetu. Následne môže prejsť do dialógu pridávania typov zostáv pre daný predmet – kde pre každú z nich môže určiť, či je verejná, či sa zverejňuje pravidelne v čase cvičení, alebo podľa osobitného nastavenia, či sa k nej riešenia do systému majú odovzdávať. V dialógu administrácie študentov učiteľ požadáva mená a adresy elektronickej pošty študentov, ktorí majú byť zapísaní na predmet – či už výberom zo zoznamu študentov, ktorí

navštevovali iný predmet v systéme, alebo nových študentov. Zároveň môže vytvoriť štúdijné skupiny pre daný predmet a zaradiť jednotlivých študentov do skupín. Na záver, v nastaveniach systému môže upraviť základné nastavenia (pridať povolené prípony, špecifikovať, či sú povolené viacnásobné submity, určiť spôsob strhávania bodov za neodovzdané úlohy a viaceré ďalšie položky).

Učiteľ pridáva novú zostavu do systému – najskôr v zozname zostav zvolí existujúcu zostavu, zobrazí ju cez voľbu editovania a uloží ju pod novým menom (klon), alebo jednoducho vytvorí celkom novú zostavu, v oboch prípadoch upraví dátumy a časy začiatku a konca zverejnenia a odovzdania zostavy. Učiteľ danú zostavu „otvorí“, čo znamená, že sa do nej môžu pridávať úlohy. Ak bola zostava vytvorená ako klon, môže najskôr niektoré z použitých úloh odstrániť. V oboch prípadoch môže cez zoznam úloh navliť úlohy, ktoré už v systéme sú a majú sa do zostavy pridať. Na zadanie nových úloh buď použije import existujúcich súborov HTML alebo PDF, alebo dialóg zadať novú úlohu, v ktorom on-line povkladá text novej úlohy, „upload“-ne a povkladá potrebné obrázky, či dátové súbory. Následne úlohu priradí kľúčové slová, môže pridať vzorové riešenie, vstupné a výstupné testovacie súbory a novú úlohu pridať do otvorenej zostavy.

Študent odovzdáva riešenie zostavy – študent sa prihlási pomocou svojho účtu, zobrazí sa mu študentské menu, kde má zobrazený zoznam platných zostáv. Volí si zostavu, nahrá archív (zip) súbor so všetkými zdrojovými a prípadne dátovými súborami. V časti hodnotení si prípadne pozrie koľko doposiaľ získal bodov a aké komentáre mu učiteľ napísal.

Učiteľ hodnotí študentovo riešenie – učiteľ sa prihlási do systému, zvolí si časť o hodnotení študentov. V prípade, že študent odovzdal riešenie na papieri (napr. riešenie rozcvičky), v dialógu zadávania bodov učiteľ iba pridá body, prípadne doplní komentár. Ak má študent odovzdané riešenie, učiteľ ho vidí v zozname všetkých odovzdaných riešení v danej skupine a zostave, prechádza do dialógu hodnotenia, kde riešenie môže prezerat', menit', spúšťať – podľa preddefinovaných scenárov, porovnávať výstup, prechádzať medzi jednotlivými odovzdanými súborami. Riešenie môže obodovať a napísať pre študenta komentár.

Hodnotenie

Predmet:

Skupina:

Študent:

Zostava:

Súbor:

Počet bodov:

Komentár:

Vstup:

Výstup:

```

11   for (int i=0; i<n; i++){
12       trojuholnik.add(i, new long[i+1]);
13   }
14   trojuholnik.get(0)[0] = 1;
15   trojuholnik.get(1)[0] = 1; trojuholnik.get(1)[1]
16   for (int i=2; i<n; i++){
17       trojuholnik.get(i)[0] = 1;
18       for (int j=1; j<i; j++){
19           trojuholnik.get(i)[j] = trojuholnik.get(i-1)
20       }
21       trojuholnik.get(i)[i] = 1;
22   }
23   }
24   public void vypis(){
25
26       for(int i=0; i<n; i++){
27           for (int k=0; k<(n-i); k++){ System.out.print
28               for (int j=0; j<=i; j++){
29                   System.out.printf("%7d", trojuholnik.get(i)
30                   System.out.print(" ");
31               }
32               System.out.println();
33           }
34       }
35   }
36   public static void main(String[] args){

```

		1			
		1	1		
	1	2	1		
1	3	3	1		

Obr. 5 Dialóg hodnotenia študentovho riešenia.

5. ĎALŠIE FUNKCIE SYSTÉMU

V priebehu niekoľkých rokov sme odskúšali rôzne verzie systému zadávania domácich úloh v predmetoch zameraných na programovanie. Príkladom môže byť sada úloh, z ktorých si študent vyberá niekoľko úloh a získava za ne body. Dospeli sme k záveru, že na baklárskom stupni je najefektívnejšie, ak je úloha povinná a neobodovaná – študent ju jednoducho musí odovzdať. Šikovných študentov naopak motivujeme osobitnými prémiovými úlohami, ktoré sú hodnotené bodmi navyše k základnému hodnoteniu. Každý týždeň sa vyskytuje nová povinná úloha. V prípade, že ju študent neodovzdá, stráca 3 body zo svojho celkového hodnotenia (ktoré je max. 100 bodov). Keďže za semester je približne 12 úloh, má však možnosť jednu (ľubovoľnú) úlohu neodovzdať. Pri druhej neodovz danej úlohe už stráca 1 bod, potom 2 a za každú ďalšiu neodovzdanú 3 body. Systém Lamsfet podporuje automatické strhávania bodov za príslušnú zostavu po obodovaní odovzdaných úloh jedným tlačídlom podľa uvedeného scenára. Prax však naznačuje, že ani toto nie je vhodný prostriedok, keďže

študenti tieto výhody využili v príliš veľkom počte a spôsobili si tak zbytočné nedostatky vo svojich programátorských zručnostiach. Systém preto podporuje aj druhý typ strhávania bodov – bezpodmienečne 3 body. Tento systém sme použili v druhom semestri nasadenia systému a ukázalo sa, že sa študenti oveľa skôr vyselektovali na poctivo riešiacich a „odpadlíkov“, ktorí predmet vzdali.

Ďalšou špecifickou funkciou je možnosť zverejňovať zostavy mimo systému Lamsfet. Systém na požiadanie zobrazí odkaz URL na ktorom visí príslušná voľná zostava, či úloha. Úlohy, ktoré sú označené špeciálnym kľúčovým slovom „public“ tiež môžu byť takto zverejnené. Systém tiež dovoľuje ku každému predmetu zvlášť vytvoriť zadania projektov, ktoré sú označené osobitným kľúčovým slovom. Tieto projekty zobrazí na jednej verejnej stránke, pričom po prihlásení si študenti zo svojho menu môžu jednu z takýchto projektových úloh vybrať. Svoj výber môžu aj viackrát zmeniť, ale každú úlohu môžu mať vybrať naraz najviac traja študenti. Riešenie projektu potom odovzdávajú do zostavy, ktorú pre tento účel pridal k predmetu učiteľ. V oboch predmetoch (Programovanie III a Programovacie Paradigmy) sme si overili, že projekt – hoci je pre študentov časovo náročný – je veľmi dôležitou súčasťou. Sami študenti nám opakovane po skončení predmetu potvrdzujú, že ich vlastný pokrok v zvládnutí látky nastal z veľkej miery vďaka práci na projekte.

Systém podporuje zálohovanie databázy a celého svojho dátového obsahu, zdrojové súbory sú umiestnené v CVS, takže v prípade nutnej reinstalácie je ho možné obnoviť do pôvodnej podoby.

Systém umožňuje učiteľom prihlásiť sa do študentského rozhrania ktorékoľvek zvoleného študenta a takto mať možnosť skontrolovať a nahliadnuť presný výstup, ktorý je študentom k dispozícii.

Systém dovoľuje exportovať všetky alebo vybrané úlohy vo formáte HTML tak, aby boli znovupoužiteľné v inom, podobnom systéme, alebo priamo.

6. SKÚSENOSTI S VYUŽITÍM SYSTÉMU

Systém sme nasadili vo výuke v plnej miere na dvoch predmetoch – v zimnom semestri 2009/2010 na povinnom predmete Programovanie III – Algoritmy v Java (143 študentov) a v letnom semestri 2009/2010 na povinnej voľiteľnej predmete Programovacie Paradigmy (83 študentov). Celkovo so systémom pracovalo 9 vyučujúcich. Okrem niekoľkých potrebných opráv drobných nedostatkov, ktoré jednotliví vyučujúci odhalili, boli všetci úspešne schopní systém používať. Na predmete Java bolo zadaných 111 zostáv úloh – vrátane štvrťročných testov pri počítačoch a skúškach - aj tie sme realizovali na systéme Lamsfet. Vysoký počet zostáv vyplýval najmä z toho, že každá štúdijská skupina dostávala inú domácu úlohu. Na predmete Paradigmy je v tretine semestra zadaných 20 zostáv úloh. Veľkou výhodou je pomoc pri administratívnej agende ohľadne zaradovania a preradovania študentov do skupín, ktoré sú vynútené komplikovanými časovými obmedzeniami vyplývajúcimi z rôznosti rozvrhov jednotlivých študentov kvôli kreditnému systému štúdia. Systém automaticky vytvára tabuľky s priebežným bodovaním študentov a tak nielen cvičiaci, ale aj prednášajúci, ktorý má tiež k systému prístup, môžu zareagovať zodpovedajúco na výkonnosť študentov. Po správnom zaregistrovaní študentov sme zaznamenali minimálne množstvo otázok od študentov ohľadne spôsobu použitia systému, alebo sťažností, že niečo nefunguje ako by malo. Naopak, študenti oceňujú osobnú spätnú väzbu, ktorú dostávajú ku svojim odovzdaným riešeniam.

7. ZÁVER A ĎALŠIE SMEROVANIE

V článku opisujeme systém na dlhodobú archiváciu zadaní úloh používaných pri príprave výuky v predmetoch zameraných na programovanie v rámci bakalárskeho programu Aplikovanej informatiky na Fakulte matematiky, fyziky a informatiky. Systém okrem archivácie zadaní úloh registruje predmety, študentov, skupiny študentov, odovzdané riešenia študentov, ich bodové hodnotenie. Systém je napísaný v jazyku HTML, JavaScript/Ajax, PHP, využitím databázy PostgreSQL a je open-source, voľne dostupný na použitie pre prípadných záujemcov.

V budúcom vývoji by sme systém chceli rozšíriť o možnosť plne automatického opravovania vybraných úloh, tam, kde to charakter úloh umožňuje a kde to učitelia uznajú za vhodný spôsob hodnotenia. Osobitnú pozornosť by sme chceli venovať zvýšeniu bezpečnosti systému a to jednak vytvorením podrobného logovania všetkých aktivít a jednak zvýšením bezpečnosti jednotlivých formulárov.

Veľmi by nás potešilo rozšírenie systému medzi širšiu komunitu učiteľov, v prípade záujmu uvažujeme nad vytvorením spoločnej databázy úloh, ktorú by sme mohli zdieľať medzi viacerými pracoviskami.

LITERATÚRA

- [1] RAYMOND A. Lorie: Long-Term Archiving of Digital Information. Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries, ACM, 2001.
- [2] JURČO P.: Dlhodobý viacúčelový sklad úloh na cvičenia. Bakalárska práca, FMFI UK, 2009.