

ВЪВЕЖДАНЕ НА ЕЛЕМЕНТИ ОТ ПАРАЛЕЛНО И МРЕЖОВО ПРОГРАМИРАНЕ В КУРС ПО ОБЕКТНО-ОРИЕНТИРАНО ПРОГРАМИРАНЕ

Филип Петров, Даниел Джолев

Представен е опит за по-ранно въвеждане на елементи от паралелно и мрежово програмиране в обучението по информационни технологии в Техническият Университет – София. Основната цел е да се направи пропедевтика и да се задълбочат междупредметни връзки. Допълнителните задачи, решавани по време на лабораторни упражнения, са с чисто практическа насоченост. Насоката е към основните принципи на обектно-ориентираното програмиране, а не към дълбочината и изчерпателността на предложените за допълнително изучаване знания. Резултатите от експеримента показват, че допълнителните задачи не повлияват на оценките по учебния предмет, но за сметка на това водят до повишена удовлетвореност от страна на студентите.

Въведение. В днешно време почти всеки компютър разполага с многоядрен централен процесор, а настолните компютри често и с мощна видеокарта с голямо количество ядра. Практическото програмиране отдавна се е насочило към усвояването на тези изчислителни ресурси и вече масово се пишат многонишковы приложения. С навлизане на облачните услуги вече сме на етап, в който редовите програмисти често се сблъскват със задачата да проектират разпределени софтуерни системи. Може да се каже, че в последните години е рядкост да се намери програмист, който не се е сблъсквал с паралелно или с мрежово програмиране в практиката си. Поради тази причина е задължително университетите да отговорят адекватно на тези нужди на бизнеса при обучението на програмисти и на софтуерни инженери.

Въпреки че има множество вариации и немалко изключения, като например [6, 7], може да се твърди [5, 8, 11], че масово наложената практика по цял свят в обучението по компютърни системи и технологии е да се започва с процедурно програмиране, последвано от обектно-ориентирано програмиране в комбинация с алгоритми и структури от данни, след което в различни последователности следва изучаването на различни декларативни езици (почти винаги SQL в курс за релационни бази от данни), както и паралелно и мрежово програмиране. Важно е да се отбележи, че не е важен езикът за програмиране, а парадигмата, защото има немалко курсове по увод в програмирането, които използват обектно-ориентирани езици, но те реално се използват за обучение по процедурно програмиране. Когато са подредени в такъв ред, се виждат ясно два големи прехода – от една страна императивното програмиране предхожда декларативното (изключения има, особено в учебни програми с изразена математическа насоченост), а от друга еднонишковото програмиране предхожда

многонишковото (авторите не успяха да открият курс, в който последното да не е вярно). Настоящото изследване е насочено към изграждане на по-плавен преход от обектно-ориентираното програмиране (ООП) към паралелното програмиране (ПП) и мрежовото програмиране (МП). Поставените цели са:

- да се изградят по-силни междупредметни връзки със следващи предмети;
- да се даде по-голяма практическа насоченост в курса по ООП.

В експеримента се добавят елементи на ПП и МП в курс по ООП. Тези елементи се включват в теми като задачи за упражнение на основните принципи на ООП. Изследването е проведено в рамките на 5 учебни години по време на лабораторните упражнения за специалност Компютърно и Софтуерно Инженерство (преди – КСТ) в Технически Университет – София.

Разпределението на учебните предмети по програмиране в Технически Университет – София е направено по следния начин [3][4] (пропуснати са профилирани и избираеми дисциплини, които не са в ядрото на учебния план):

- 1 курс, 1 семестър: Програмиране и използване на компютри 1 (ПИК1)
- 1 курс, 2 сем.: Програмиране и използване на компютри 2 (ПИК2)
- 2 курс, 1 сем.: Програмиране и използване на компютри 3 (ПИК3)
- 2 курс, 2 сем.: Синтез и анализ на алгоритми (САА) и Бази от данни (БД)
- 3 курс, 1 сем.: Операционни системи (ОС) и Програмни езици (ПЕ)
- 3 курс, 2 сем.: Програмни среди (ПС), а в последните години и Програмиране за мобилни устройства (ПМУ)
- 4 курс, 1 сем.: Компютърни мрежи (КМ) и Паралелно програмиране (ПП)

Въпреки че са отделни учебни предмети, ПИК1 и ПИК2 по същество представляват един двусеместриален курс по процедурно програмиране на езика С. Курсовете се надграждат и повеждат студентите от стандартните езикови конструкции и писане на прости алгоритми до по-сложни задачи, като например работа с файлове, с побитови операции и управление на данни в динамичната памет. Може да се каже, че до известна степен ПИК1 е ориентиран повече върху практически задачи, докато ПИК2 е по-теоретично насочен. Учебният предмет САА по същество продължава процедурното програмиране, като го обогатява със знания по структури от данни в програмирането, които се реализират отново процедурно на С. Също така по САА усилено се изучават алгоритми – оценка на сложност, сортиране на масиви, хеширане и др. Курсът по БД се води в паралел със САА, като преподавателите максимално се синхронизират и допълват.

Обучението по обектно-ориентирано програмиране се застъпва в учебните предмети ПИК3 и ПЕ. Подобно на ПИК1 и ПИК2, ПИК3 и ПЕ също могат да бъдат разгледани като двусеместриален курс, въпреки че учебните предмети са отделени един от друг с „пауза“ от един семестър. Отново може да се каже, че ПЕ разглежда значително по-сериозно концепциите на ООП в теоретичен план. На упражненията по ПИК 3 се използва Java, а по ПЕ – C++.

За първи път в учебната програма елементи от ПП се изучават в курса по ОС. Там студентите получават знания за основни понятия като процес, нишка, мъртва хватка, синхронизация и др., но предимно в теоретичен план, а не толкова от приложна гледна точка. В предметите ПС и ПМУ студентите неизбежно използват нишки (най-малкото заради създаването на отзивчива графична среда), но чрез готови библиотеки в съответните среди за програмиране (.Net и *Android Studio*).

Едва в четвърти курс студентите се сблъскват в детайли с паралелното програмиране в специализирания курс по ПП. Колкото до МП – то се застъпва частично при предмета КМ, но само като малка част от целия курс.

В създадения **експериментален модел за курс по ООП с елементи на ПП и МП** се модифицират лабораторните упражнения по ПИКЗ. На студентите се дава базова представа за управление на нишки и работа със сокети в края на курса, където в хорариума по принцип е отделено време за упражнения с практически задачи. Основната концепция е:

- Синхронизацията и работата със споделени ресурси се представя предимно концептуално – подобни задачи се разглеждат и се дават примери, но не се изисква от студентите да се справят с тях. Липсата на атомарност на операциите и неправилната синхронизация на споделени ресурси са най-честите грешки в ПП [10], но авторите оставят справянето с тези задачи като приоритет на по-горните курсове, където студентите ще имат специално отделено време в хорариум;
- Както е отбелязано в [9], студентите, които учат на принципа „проба-грешка“, изпитват сериозни затруднения при първия си сблъсък с ПП. Техният код става объркан и значително по-сложен от необходимото, а това нормално предпоставя множество грешки. Поради тази причина се оказва, че спазването на основните принципи на ООП за абстракция и капсулация значително подпомага процеса на създаване на многонишкови сървърни приложения, като програмният код става по-ясен, лесен за управление и изчистен от грешки. В този смисъл може да се каже, че сблъскването на студентите със знания от тяхната зона на близко развитие [1] подпомага затвърждаването на вече изучените знания;
- По същия начин елементите на МП се въвеждат като упражнение за полиморфизъм, защото използването на TCP/IP сокети в Java не е значително по-различно от работата с бинарни или с текстови файлове, печатането на екран или четенето от клавиатура. То също така спомага и за утвърждаването на принципа за капсулация на данните, защото на студентите вече се налага да споделят ресурси между различни програми, тоест капсулацията е задължителна. Освен това при мрежовото програмиране коректната обработка на изключения не е просто препоръчителна, а задължителна ако програмистът иска да направи добре работещ сървър.

В края на курса се очаква студентите да могат да създадат многонишков сървър, в който всяка нишка обслужва различен клиент на системата. Споделянето на ресурси между различните клиенти на сървъра се свежда до минимум и по възможност се избягва.

Резултатите от проведения експеримент са получени от сравнението на две учебни години, в които експерименталният модел не е въведен – 2010/11 и 2011/12 – с три, в които е – 2012/13, 2013/14 и 2014/15. В контролната група на студентите по ПИКЗ е преподаван традиционният курс по ООП, в края на който са упражнявани предимно тривиални за подобен курс задачи, а на експерименталните групи са преподавани допълнителни знания по мрежово и паралелно програмиране. Изследването е направено в две посоки:

- Проверява се дали значителното усложняване на учебния материал влияе на

средният успех не само по ПИКЗ, но и по ПЕ, т.е. ООП като цяло;

- Провежда се анкета за проверка дали студентите одобряват промените.

В таблица 1 е представен средният успех в края на семестъра на студентите в различните учебни години по учебния предмет ПИКЗ. Разгледани като обединени извадки, контролните групи по ПИКЗ са статистически еднакви с експерименталните според U-тест на Ман-Уитни със z-стойност 0,992. В таблица 2 е представено процентното разпределение на оценките по ПИКЗ, а в таблица 3 са представени резултати по учебния предмет ПЕ. Разгледани като обединени извадки, групите по ПЕ са статистически различни със z-стойност от 5,569. Това потвърждава хипотезата, че средният успех се е повишил след прилагане на експерименталния модел по ПИКЗ.

Проведената анкета е с незадължително участие сред студенти от експерименталните групи. Зададен е един въпрос относно учебното съдържание в експерименталния модел. Резултатите от анкетата са показани в таблица 4. При проведена съпътстваща дискусия с 40 от по-активните студенти се оформиха следните тенденции:

- Студентите одобряват задачите с елементи на ПП и МП;
- Сложността на някои от задачите с елементи на ПП е прекалено висока. Проблемни са местата, при които нишки трябва да споделят общ ресурс;
- След примера с въвеждането на експерименталния модел, повечето от критиките на студентите се насочват към предишните два учебни предмета – ПИК1 и ПИК2. Забелязва се изявено желание у студентите за по-интензивно обучение с по-практически насочени задачи и при тях;
- Наблюдава се отчетливо желание на студентите да се увеличи хорариумът за лабораторни упражнения по ПИКЗ.

Направеният извод е, че добавянето на елементи от ПП и МП в експерименталния модел, не оказва значително влияние върху подготовката по ПИКЗ на студентите – техните резултати в края на семестъра са сходни с тези от контролната група. Все пак се забелязва тенденция към намаляване на отличните оценки. За сметка на това се наблюдава леко повишаване на успеха по учебния предмет ПЕ, което кара авторите да изведат хипотеза, че практическите задачи с елементи на МП и ПП по ПИКЗ са довели до по-добро затвърждаване на знанията по ООП.

На базата на проведената анкета и съпътстващата я дискусия са отчетени някои недостатъци и впоследствие са предложени следните корекции на модела:

- Липсата на предварителна теоретична подготовка по паралелни алгоритми, която да предшества кодирането и практическото програмиране, е вероятна причина за трудностите, които изпитват студентите. В този смисъл би било добре моделът да бъде разширен с известна теоретична подготовка по ПП по време на лекции.
- „Ръчното“ синхронизиране на достъп до споделени ресурси се явява прекалено сложно за студентите на този етап от тяхното обучение. Вероятно би било по-добре да се използват готови библиотеки от пакета `java.util.concurrent`;
- Курсът ще бъде облекчен от гледна точка на писане на алгоритми на тривиални задачи, които вече са разглеждани в курсовете по ПИК1 и ПИК2. За целта може по-активно да се използва библиотеката `Stream` в Java 8.

Таблица 1. Среден успех по ПИК3 за периода 2010–2014 г.

2 курс, 1 сем.	Контролна група		Експериментална група		
	Година	2010/1	2011/2	2012/3	2013/4
Брой студенти	200	124*	167	197	235
Среден успех	3,49	3,18	3,30	3,22	3,06

* За 2011/12 не са включени студенти, участвали в пилотен експеримент

Таблица 2. Процентно разпределение на оценките

2 курс, 1 сем.	Контролна група		Експериментална група		
	Година	2010/1	2011/2	2012/3	2013/4
Слаб 2	50,50%	56,45%	44,31%	53,80%	56,17%
Среден 3	7,00%	11,29%	22,15%	13,20%	15,32%
Добър 4	9,00%	9,68%	9,58%	8,12%	8,09%
Мн. Добър 5	9,50%	2,42%	7,19%	7,11%	7,66%
Отличен 6	24,00%	20,16%	16,77%	17,77%	12,76%

Таблица 3. Среден успех по ПЕ за периода 2011–2014 г.

3 курс, 1 сем.	Контролна група		Експериментална група	
	Година	2011/2	2012/3	2013/4*
Брой студенти	196	154	144	185
Среден успех	2,80	3,17	3,45	3,40

* За 2013/2014 г. е правена допълнителна контролна работа за повишаване на успеха, която не е взета под внимание в обобщените резултати

Таблица 4. Резултати от проведена анкета

Въпрос	Смятате ли, че има нещо в програмата по ПИК3, което трябва да се промени?			
	Да не се премахва нищо, а да се добави нов материал	Да се премахнат трудни теми (например нишки) и да се заместят с нови (Java 8 и др.)	Да не се променя нищо – курсът е много добър!	Материалът трябва да се опрости значително, като се премахнат по-трудните теми
Резултати	47	14	7	1

Изследването подпомогна за промяна в учебната програма по учебния предмет ПИК2 – от 2014/15 в упражненията по ПИК2 се разглеждат значително повече практически ориентирани задачи. Обмисля се и по-нататъшно продължение на практиката за засилване на междупредметни връзки, особено що се отнася до учебния предмет Бази от данни.

Авторите изказват специална благодарност на проф. Даниела Гоцева за предоставената възможност да проведат експеримента в упражненията по нейния учебен предмет. Експериментът се финансира частично от ФНИ в Софийски Университет „Св. Климент Охридски“ с договор №140/17.04.2015 г.

ЛИТЕРАТУРА

- [1] Л. ВИГОТСКИ. Мислене и реч, Наука и изкуство, София, 1983 г.
- [2] Ив. ГАНЧЕВ, Й. КУЧИНОВ. Диалогови обучаващи програми – същност, ръководни идеи, принципи и етапи на разработването им. *Обучението по математика*, **2** (1987), 1–8.
- [3] Учебен план на специалността „Компютърни Системи и Технологии“, професионално направление: 5.3. Комуникационна и компютърна техника, Технически Университет – София, 12.04.2011 г.
- [4] Учебен план на специалността „Компютърно и Софтуерно Инженерство“, професионално направление: 5.3. Комуникационна и компютърна техника, Технически Университет – София, 08.04.2014 г.
- [5] P. BURTON, R. BRUNN. Teaching Programming in the OOP era, ACM SIGCSE Bulletin, ACM SIGCSE Bulletin, Pages 111-114, ACM New York, NY, USA, ISSN 0097-8418, 2003.
- [6] S. COOPER, W. DANN, R. PAUSCH. Teaching Objects-first In Introductory Computer Science, Proceeding SIGCSE '03 Proceedings of the 34th SIGCSE technical symposium on Computer science education, 2003, 191–195, ACM New York, NY, USA, ISBN:1-58113-648-X.
- [7] M. KÖLLING. The problem of teaching object-oriented programming. *Journal of Object Oriented Programming*, **11**, No 8 (1999), 8–15.
- [8] R. MASING, G. COOPER, M. RAADT. Trends in Introductory Programming Courses in Australian Universities – Languages, Environments and Pedagogy. Proceedings of the Fourteenth Australasian Computing Education Conference (ACE2012), Melbourne, Australia, 2012, ISSN 1445-1336.
- [9] C.-K. SCHENE. Multithreaded Programming in an Introduction to Operating Systems Course. Proceedings of the Twenty-Ninth SIGCSE Technical Symposium on Computer Science Education, Atlanta, Georgia, USA, February 26–March 01, 1998 242–246.
- [10] C. SUNG-EUN, C. E LEWIS. A Study of Common Pitfalls in Simple Multi-Threaded Programs. Proceedings of the Thirty-First SIGCSE Technical Symposium on Computer Science Education, Austin, Texas, USA, March 07–12, 2000, 325–329.
- [11] J. YAU, M. JOY. Introducing Java: The Case for Fundamentals-first. International Conference on Education and Information Systems – Technologies and Application, Orlando, FL, 21-25 Jul 2004, International Conference on Education and Information Systems, Technologies and Applications (EISTA 2004), **2**, 2004, 229–234.

Филип Петров Петров
Факултет по математика
и информатика
Софийски Университет
„Св. Климент Охридски“
бул. Джеймс Баучър № 5
1164 София
e-mail: philip@abv.bg

Даниел Иванов Джолев
Факултет по компютърни
системи и управление
Технически Университет – София
бул. Климент Охридски № 8
1756 София
e-mail: daniel.djolev@abv.bg

INTRODUCTION OF ELEMENTS OF PARALLEL AND NETWORK PROGRAMMING IN THE OBJECT-ORIENTED PROGRAMMING COURSE

Philip Petrov, Daniel Djolev

In this paper we present an attempt to introduce some elements of parallel and network programming in the Object-Oriented Programming (OOP) subject with focus to increase the interdisciplinary relationships with the subjects Operating Systems and Parallel Programming. The additional tasks are introduced at the end of the course and are quite practical. They are designed to not dislocate the main purpose of the OOP course – teaching the principles of abstraction, encapsulation, inheritance and polymorphism are still the subject core and the new elements are planned to straighten their understanding by exercises with real-life practical examples. We compare the students' results from 3 years of teaching using the new approach as an experimental group with 2 years teaching the traditional OOP course as a control group. Despite the increased difficulty, the results of the experiment show that the additional tasks do not significantly affect the students' grades. Our survey shows a clear trend for better overall satisfaction from the subject. Such pragmatic elements in the teaching of the introductory IT subjects should be increasing the motivation of the students as they will feel the education closer to the business.