

ПРОБЛЕМИ НА ОБУЧЕНИЕТО ПО ПРОГРАМИРАНЕ

Теодоси Кирилов Теодосиев

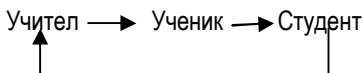
9712 Шумен, ШУ "Еп. К. Преславски", ФМИ
катедра "Компютърна информатика"
t.teodosiev@fmi.shu-bg.net

Abstract: Presented paper considers and comments some problems in teaching of programming. Part of them is common of education. When add to specifically problems with difficulties of programming: creating of model, pure algorithmic thinking and constructing helplessness, the problems becomes much. It is comments an approach for supporting of training – improving program's style.

Ключови думи: програмиране, обучение, език за програмиране, стил на програма

1. Увод. Предлаганата работа разглежда някои основни (според автора) проблеми на обучението по програмиране на студенти и ученици. Част от тях са общи за цялото образование (масовизацията на висшето образование на базата на демографския срив, липсата на мотивация, ниската общообразователна подготовка и др). Не може да се мотивира студент станал такъв, само за да си реши някои социално-битови проблеми (да отложи влизането в „армията на безработните“ или отбиването на редовна военна служба) или да води безгрижен живот и приет в специалност, към която няма никакъв афинитет.

Като се прибавят към тях и специфичните проблеми свързани с трудността на дисциплината програмиране: слабо алгоритмично мислене и конструктивно безсилие, проблемите стават много. По-страшното според мен е това, че попадаме в „порочен“ кръг. Моите наблюдения показват, че всяка година новоприетите студенти в специалност „Математика и информатика“ (бъдещи учители) идват в университета все по- неподготвени. Този „нисък“ старт, прибавен към недобрния учебен план [1], липсата на време и субективни причини не позволява „висок“ финал. Така се затваря кръга.



В точка 2 са коментирани въпросите за мястото на програмирането в учебната програма по Информатика и избора на първи език за програмиране в обучението. Няколко проблема в обучението по програмиране са разгледани в

точка 3. В точка 4 се предлага подход за подпомагане на обучението – подобряване на стила на програмиране.

2. Програмирането в учебната програма. Първият въпрос е свързан с това, трябва ли всички ученици да изучават програмиране в училище. Безспорно е, че сравнително малко имат афинитет към тази дейност. Застъпвам мнението, че програмирането има мирогледно значение за човека. То е творческа, конструктивна дейност. Приучва на точност и дисциплина. Изисква добра обща култура при решаване на практически задачи и е отличен пример за междупредметните връзки. Полезно е за всеки интелигентен човек да има представа от алгоритми и начините за изпълнението им от машина. За голяма част от учениците трябва да се спре дотук, само в профилираните паралелки да се продължи с задълбочено изучаване на програмирането.

Вторият въпрос е свързан с първия език за програмиране (ЕП). Да не забравяме, че съответният модул в учебната програма се нарича „Програмиране“, т.е. ЕП е второстепенен въпрос. Без да отдавам особено значение на този елемент, разглеждайки го само като средство за усвояването на основите на програмирането, ще споделя няколко съображения.

Първите стъпки в нова дейност трябва да се правят внимателно с прости, евтини и леки за използване помощни средства, които да ни предпазват от грешки и контузии. В този ред на мисли, смятам езика *PASCAL*, създаден за обучение, за по-подходящ като първи език за програмиране в сравнение с професионалния по-моцнен, по-модерен език *C* [2]. Все още не само у нас този въпрос е спорен. Въпреки, че много университети и училища у нас и в чужбина преминаха на начално обучение по програмиране на *C* на някои места се търси алтернатива (напр. *Python*). Освен това в университета може да се разчита на подготовка по програмиране от училище, т.е. по идея това не е начално обучение по програмиране. Подготвените преподаватели на *PASCAL* са повече, а се знае че най-добър е онзи ЕП, който знаеш най-добре.

Разбира се с казаното дотук в никакъв случай не отричам обучението по програмиране на базата на *C*, дори напротив. То е подходящо за обучаеми със сериозен интерес към програмирането (бъдещи професионални програмисти). А защо не и като надграждане върху знанията по програмиране на *PASCAL*.

3. Проблеми на обучението по програмиране. Обучението по програмиране поставя за разрешаване сериозни проблеми пред преподавателите.

3.1. *В училище.* Как да се развие у обучаемите алгоритмичното мислене? Как да се научат обучаемите на синтаксиса и лексиката на нов език за изразяване. Последователността следвана в учебниците по информатика за 9 и 10 клас най-напред модула „Алгоритми“, а след това модула „Програмиране“

е правилна и естествена. Така се разделят трудностите свързани със съставяне на алгоритми и усвояване на език за програмиране. Алгоритмите е добре да се описват на псевдокод (ограничен естествен език) като по този начин се спестяват затрудненията свързани с ново инструментално средство.

3.2. *В университета.* Тук нещата малко се разместват. Опитът показва, че има голямо разслоение на студентите по отношение на знанията им по програмиране от училище. Разчитайки на наученото в училище в курса по „Програмиране“ се набляга на програмирането на избран ЕП. За малка част от обучаемите този подход е удачен, тъй като вече могат да съставят алгоритми. За останалите за съжаление се натрупват двата проблема. Налага се паралелно обучение по алгоритми и програмиране. Така възниква конфликта между учебната програма и ефективността на обучението.

Проблемите се усложняват в другите дисциплини свързани с програмиране (ООП, СОД, ЕФЛП и други). Пропуските в знанията по „Програмиране I част“ правят почти невъзможно ефективното усвояване на „Програмиране II част“, а още повече други парадигми в програмирането като обектно-ориентираното, функционалното и логическото. Например в дисциплината „Езици за функционално и логическо програмиране“ възниква необходимостта да се използва рекурсия (тема от „Програмиране 2 част“). Тя е една от най-трудните теми в курса по програмиране (независимо от езика за програмиране). Това се дължи на липсата на популярност на рекурсията като мощно средство в математиката и противоречието между императивния характер на ЕП и декларативната (скрита за програмиста) същност на механизма на рекурсията.

Важността на темата се определя от естествения подход за реализация на рекурсивните структури от данни (списък, дърво, стек опашка и др.) и като подготовка за изучаване на ЕП, основаващи се изцяло на рекурсията. Знае се, че в ЛИСП и ПРОЛОГ основен инструмент е рекурсията. Сблъскваме се с неподготвеността на студентите да намират рекурентни връзки.

Илюстративните примери трябва да са внимателно подбрани. Добре би било рекурсивния процес да се онагледява (например с графично изчертаване на дърво). Програмите да се реализират с различна посока на рекурсивния процес - нарастване и намаляване на параметрите („таван“ и „дъно“). Избраните примери да са положителни - рекурсивното решение да е по-ефективно, по-ясно и по-кратко от нерекурсивното такова.

3.3. *Моделиране.* Как да се повиши интереса на обучаемите към програмирането? Разбира се чрез показване на ефектни и интересни програми, но без да се забравя, че те трябва да са по силите на обучаемите. Например някои познати игри. Интересът към компютърните игри може да се

използува за увеличаване на ефективността на началното обучение по програмиране [3].

Често обучаемите изявяват претенции за решаване на по-интересни (практически) задачи. Това е безспорно правилен подход. Тук възниква още един проблем, свързан с модела на задачата и познанията на обучаемите в предметната област. Сблъскваме се с познатите проблеми при решаване на „текстовите“ задачи по математика. Слабата общообразователна подготовка, проблемите с извличане и интерпретация на информацията затруднява много създаването на математическия и информатичен модел на задачата, а оттам и написването на вярна програма.

4. Стил на програмиране. Един нетрадиционен подход в посока на преодоляване на проблемите е подобряване на стила на компютърните програми в процеса на обучението.

Първостепенно значение има верността на програмата. Стилът има роля по-скоро за облекчаване на разбирането на програмата в процеса на обучение. Стилът на програмиране е пряко свързан с професионалната подготовка и рутината на преподавателя [4].

В последните година-две, използвайки предимствата на новите технологии, често преподавателите използват готови програмни кодове, които да коментират. Това води до интензификация на обучението, по-добро обмисляне и предпазване от грешки. Готовите програми имат приложение и в модерния начин за изпитване - тестовете. В тези случаи стилът е още по-важен. Опитът ме убеди, че в началното обучение по програмиране тестовете стимулират „играта на тото“. Привърженик съм на съставянето на програмата пред обучаемите. Така те могат да проследят процеса на програмиране дори и корекциите свързани с подобряване на верността и стила на програмата.

Алгоритъмът и програмата трябва да се отличават с подреденост и яснота на идеите. Да се спазват принципите на структурното и модулно програмиране. Важно в тази посока е използването на подалгоритми и подпрограми. Разбира се това не трябва да е самоцел, а примерите убедително да подкрепят избрания подход. В противен случай обучаемите приемат писането на подпрограми като излишно усложняване на процеса на решаване на задачата.

Програмният код трябва да е ясен и кратък. Текстът на програмата трябва да се подреди така, че да следим логиката ѝ, да четем отгоре надолу. За това може да помогне избягването на многократни изчисления на изрази. По този начин се повишава и ефективността на програмата.

Написването на програмата трябва да е съобразено и с особеностите на компютърните пресмятания. Така освен постигане на по-добър стил в много случаи се предпазваме от грешки при изчисленията. Съставните условия ни помагат да избягваме обемистите и тромави вложени условни оператори. Правилното конструиране на съставни булеви изрази е особено важно при работа с масиви и операции, определени не за всяка стойност на аргументите [5].

Има два погледа към стила: откъм простота за четене и коригиране и относно ефективността на програмата - добрият стил я прави по-ефективна и икономична на ресурси. Често тези два погледа си противостоят, така че трябва компромис между двата подхода в зависимост от предназначението на програмата.

Не трябва да се забравя, че има разлика между програми за обучение и практически използвани програми. В обучението вниманието трябва да се съсредоточи върху четивността на програмния код дори за сметка на ефективността при изпълнение. Преподавателят пише програми с цел обучение, т.е. те трябва да са лесни за разбиране. Често програмата се доработва или разширява не от този, който я е написал. Ето защо проблемът за стила е от съществена важност. Формата трябва да илюстрира, да обяснява и документира съдържанието и замисълът [6].

За практическите програми основна е ефективността- ресурсна и времева. По отношение на първия аспект съществен е изборният и модифициран алгоритъм. По отношение на втория аспект определящо значение има отличното познаване на особеностите на ЕП и системата за програмиране както и особеностите на хардуера.

Стилът на програмиране в обучението има още по-важна роля за специалистите, защото те трябва да се научат да пишат ефективни от гледна точка на време и ресурси програми, но и програми с ясни идеи, позволяващи работа в колектив и лесна модификация.

5. Заключение. Вниманието ми е съсредоточено върху профилираните паралелки по информатика в средните училища и студентите от специалност „Математика и информатика“, защото считам, че при тези две категории обучението по програмиране има решаваща роля. Трябват ни добри учители, за да подготвят качествено бъдещите специалисти (информатици и учители по информатика). Разбира се казаното е валидно и за другите категории обучаеми. Повече са проблемите и много по-малко решенията, които предлагам. Голяма част от тях се коренят в образователната ни система: трябва да се реши: масовост или качество ще търсим в образованието, а може би и двете, но чрез различни типове средни и висши училища.

Литература

- [1] Христова, Р. и др. Двустранен поглед върху обучението по информатика, In: Proceeding of the 26th ICT&P'2001,София, 170-174.
- [2] М. Бърнева, Т. Теодосиев, Паскал или Си като първи език за програмиране, Сборник доклади на 31-вата Пролетна конференция на СМБ,2002, София, 234-238
- [3] Г.Теодосиева, Т.Теодосиев, Игровите задачи в началното обучение по програмиране, Сборник доклади на 31-вата Пролетна конференция на СМБ,2002, София, 285-289
- [4] Теодосиев, Т., Теодосиева Г. Бележки по обучението в стил на програмиране, Сборник доклади на Национална конференция " Информационни изследвания, приложения и обучение- i.tech '2003", Варна, 2003, 84-90
- [5] Т.Теодосиев, Г.Теодосиева, Математически проблеми в обучението по програмиране, Сборник доклади на Научно-приложна конференция „Математика, Информатика и Компютърни науки”, В.Търново,2006, 458-462
- [6] Теодосиев, Т., Теодосиева, Г. Още за обучението в стил на програмиране, In: Proceedings of the 30th International Conference ICT&P'2005,София, 2005,118-124.