

ПРОБЛЕМИТЕ В ОБУЧЕНИЕТО НА НАЧИНАЕЩИТЕ ПРОГРАМИСТИ И ВЪЗМОЖНОСТИ ЗА ТЯХНОТО ПРЕОДОЛЯВАНЕ

Галина Атанасова¹, Пламенка Христова²

Русенски Университет “Ангел Кънчев”,
катедра Информатика и информационни технологии
7017, гр. Русе, ул. „Студентска” 8

¹gea@ami.uni-ruse.bg, ²ptx@ami.uni-ruse.bg

Резюме: *Начинаещите програмисти намират въвеждането в програмирането за доста трудно. Множеството от нови знания, които трябва да бъдат усвоени, действат демотивиращо. На по-късен етап от обучението се регистрира наличие на пропуски в познанията относно основните алгоритмични конструкции и парадигми. Представя се една възможност за повишаване на способностите за решаване на задачи чрез среда за анимирано представяне на алгоритми.*

Ключови думи: *алгоритми, начинаещ програмист, обучение, среда за анимирано представяне на алгоритми, проблеми в обучението*

Предварителната подготовка на начинаещите програмисти

За „начинаещ програмист” може да се счита всеки студент от първи курс от специалност в областта на компютърните науки, който по време на средното си образование не е посещавал извънкласни форми по информатика и не се е занимавал самостоятелно с алгоритми и програмиране. В учебните програми на МОМН за настоящия момент последните две се изучават в 9 клас на СОУ и включват преподаване на алгоритми. В учебника по информатика [1] са предвидени 10 урочни единици, посветени на алгоритмите. Те са разгледани чрез определение, същност, свойства, начини за описание и основни видове. Обучението е подкрепено с определен брой примерни задачи, които създават основна представа у учениците за понятието алгоритъм. Формулираните очаквани резултати на ниво учебна програма, одобрена от МОМН [17], са сведени до умения за разчитане на „кратки програми, написани на алгоритмичен език”, познаване на основните алгоритмични конструкции и елементарни структури от данни.

Усвоените знания и умения от средното образование може да се използват като основа за обучение във висшето образование, но трябва съществено да се доразвият и обогатят, за да се осигури възможността за придобиване на професионални компетенции в областта на компютърните науки.

Често наблюдавани проблеми в обучението на начинаещи програмисти

Известен факт е, че начинаещите програмисти намират въведението в програмирането за доста трудно [7]. Това е така в повечето случаи, когато уменията се въвеждат паралелно с конкретиката на даден език за програмиране. Тогава множеството на знанията за усвояване включва: синтаксис и семантика на езика за програмиране, разбиране и усвояване на основните концепции в програмирането, особености и интерфейс на средата за програмиране и не на последно място умения и стратегии за проектиране на ефективни алгоритмични решения. Може да се види, че единиците нови знания са големи по обем и да се прецени, че един начинаещ лесно може да се почувства обезнадежден и демотивиран да се справи с предизвикателството да научи всичко. Началните курсове по програмиране се определят като трудни и процентът на студентите, неуспели в тях, е висок [14]. За повечето студенти, като изключение правят само най-добрите, програмирането изглежда трудно достъпно, особено през първата половина на въстпителния курс. Тези факти в голяма степен се дължат на големия обем нови знания, които споменахме по-горе. Синтаксисът и семантиката на езика за програмиране играят важна роля в обучението по програмиране и проблемите с тях са свързани с факта, че доста незначителни синтактични грешки могат да затруднят процеса на обучението, като забавят и/или демотивират студента. Примери за трудности при използването на езици за програмиране в обучението на начинаещи програмисти са посочени в [3, 6, 11, 12]. За преодоляване на споменатите проблеми са проведени няколко опита чрез изолиране на езикови подмножества за начинаещи на базата на някой от съществуващите езици за програмиране. В тези подмножества участват само подобрани ключови думи от езика. Изследване [5] не показва категорични предимства пред традиционното обучение, използващо конкретен език за програмиране. В [16] са обобщени резултатите от множество изследвания, които доказват, че няма полза от преподаване на сложен материал във въвеждащите курсове по програмиране, тъй като студентите не владеят основните циклични конструкции. Той намира за целесъобразно отделянето на повече внимание и време за усвояване на базови алгоритми.

Друг основен проблем в обучението на начинаещи програмисти е изграждането на умения за съставяне на ефективен алгоритъм на дадена задача. Липсата на тези способности се забелязва успоредно със затрудненията относно синтаксиса и семантиката на езика за програмиране. Изследване [15] доказва, че обучаваните срещат значителни затруднения да „съберат всичко в едно“, да съставят алгоритъм, който вярно и ефективно да решава поставената задача. Липсата на такива умения у студентите оказва влияние върху резултата от цялостното им обучение в областта на

компютърните науки и засяга много от надграждащите дисциплини. Друго изследване по темата показва, че преминали въстпителните курсове по програмиране студенти, притежават знания по синтаксиса на езика, но имат проблеми със съставянето на алгоритъм на задачата, която им е поставена [10].

От казаното до тук може да се обобщят следните проблеми:

- Многообхватност на въвеждащия курс по програмиране. Въстпителните курсове по програмиране съдържат няколко компонента за усвояване: синтаксис и семантика на език за програмиране, програмни концепции, особености и интерфейс на среда за програмиране, умения за съставяне на ефективни решения. Мащабността на материала обезкуражава и демотивира начинаещите, като част от тях се отказват.
- Въвеждане на основните алгоритмични конструкции, успоредно със знания относно конкретен език за програмиране. Съпътствано е с проблемите при усвояване на синтаксиса и семантиката на самия език, които придобиват доминираща роля и изместват акцента на обучението.
- Липса на умения за съставяне на ефективно решение на дадена задача. Породено от недостатъчно отделено време за разбиране и усвояване на основните алгоритмични конструкции, както и изграждане на умения за абстрактно мислене.

Преодоляването на обобщените проблеми би могло да доведе до повишаване на качеството на обучение и мотивацията у начинаещите програмисти.

Една възможност за повишаване способностите за решаване на задачи

В [13] авторите правят следното обобщение: *“Програмирането е високо интелектуална дейност, която изисква програмистът да разработи високо ниво на абстракция за представяне на реалността от същности чрез логически структури. [...] Добре разработеният и точен модел на абстракция може значително да благоприятства успеха на начинаещия програмист във въвеждащите курсове, [...] подпомагането на студентите да създават добри модели на абстракция би трябвало да остане основна цел за въстпителните курсове по програмиране.”* В обучението на начинаещи програмисти акцентът трябва да се постави върху решаването на задачи и съставянето на алгоритми. Изчерпателното изясняване на същността на основните алгоритмични конструкции, комбинирането им за специфичните и градиращи по сложност решения развиват необходимите умения за абстрактно мислене и способността за решаване на реални проблеми.

Алгоритмите са начин за представяне на подробно решение как от входните данни да се достигне очакван и верен резултат на дадена задача. Има различни начини за тяхното представяне. Най-подходящи за целите на въстпителното обучение остават блок-схемите. Те по традиция се използват за добиване на ясна зрителна представа относно програмния алгоритъм и са отлично средство за визуализацията му за новаци. Леки са за усвояване, дори и без предварителна подготовка, и осигуряват на начинаещите възможност за точно описание на даден алгоритъм. Освен това, блок-схемите помагат при прехода между условието на задачата и синтактичното ѝ решение. Този преход се явява едно от основните затруднения за начинаещите програмисти.

Среди за анимирано представяне на блок-схеми

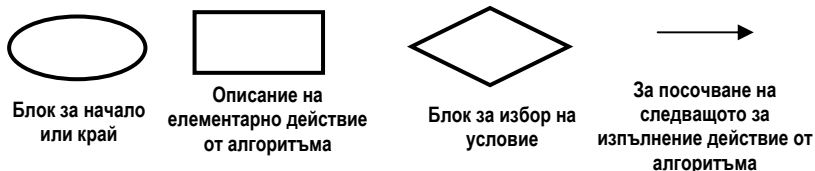
Разработени са много среди за визуализация на логиката и действието на дадена програма с цел подпомагане процеса на обучение на начинаещи програмисти. Различни са акцентите, а от там и възможностите при работа с тях. Сравнително ранен пример за подобна среда е Balsa [4]. Предоставя визуализация чрез анимирани изображения на поведението на множество специфични алгоритми като например сортировки, търсене и други. Основен недостатък на Balsa, от гледна точка на начинаещите програмисти, е липсата на възможност те да анимират собствен алгоритъм. По-скорошен пример за такъв тип среда е Jeliot-2000 [8]. Заложената цел е да се създаде ясна зрителна представа за действието на всяка програмна инструкция върху данните, използвани в дадена Java програма. Това е постигнато чрез анимация на структурите от данни, резултати от логически и математически изрази. Недостатък на Jeliot 2000 е, че за потребителя остава скрито взаимодействието между отделните компоненти на алгоритъма. В [2] Крушков и колектив описват система за обучение по програмиране, чиято цел е преодоляването на бариерата на абстрактността при съставянето на алгоритми. Авторите предлагат готов набор от анимирани алгоритми за разясняване на основните структури от данни и оперирането с тях. Липсата на възможност за визуализация и анимация на потребителски алгоритъм се явява като недостатък от гледна точка на начинаещите.

Споменатите три среди за подпомагане процеса на обучението на начинаещи програмисти са малка част от множеството, създадени с тази цел. В различна степен всички те носят ползи за начинаещите. В случай, че отличителните и положителни черти на тези инструменти се доразвият, разширят и комбинират в единствено приложение, това би могло да доведе до по-широк обхват на споменатите проблеми и да допринесе за тяхното преодоляване. Целите в работата ни са да редуцираме във възможно най-голяма степен формулираните проблеми при обучение на начинаещи програмисти, като намалим нивото на абстракция и използваме предимствата, които предлагат визуализацията и анимацията на даден алгоритъм. За

реализирането на поставените цели считаме за удачно проектиране, прототипизиране и изследване на среда за обучение на начинаещи програмисти.

Среда за обучение на начинаещи програмисти

Средата е предназначена за обучение през първите няколко седмици или семестър на начинаещи програмисти без или с ограничен опит в областта на алгоритмите. Обхванати са основните алгоритмични конструкции – линейна, разклонена и циклична. Акцентирано е върху присвояването на стойности на променливи. Тези концепции са общи за всички езици за програмиране от трето поколение. Придобиването на солидни знания относно тях осигурява надеждна основа за надграждане на умения и компетенции в областта на програмирането.



Фиг. 1а Основни блокове



Фиг. 1б Обръщение към подалгоритъм

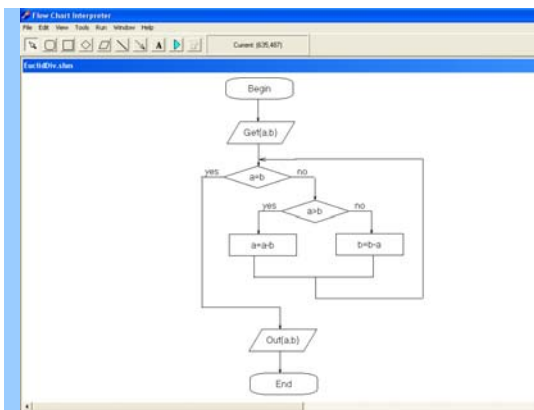
Проектирането на потребителския интерфейс на средата е съобразено с целевата група. Установено е, че добре проектираният потребителски интерфейс засилва концентрацията и спомага мотивацията на потребителите [9]. Приложението (фиг. 2) има изглед, подобен на разпространените офис приложения, които предлагат възможности за създаване на графики и диаграми, като например Microsoft Visio и Drawing от MS Office. Потребителят има възможности да посочва инструмент за чертане и свободно да разполага съответния елемент на избрана от него позиция. Всяка блок-схема може да бъде изчертана чрез основните дефиниции от нотацията ISO 90004.4. (фиг. 1а). В допълнение е предвиден блок за посочване на обръщение към готов подалгоритъм (фиг. 1б), тъй като за начинаещите са от особено значение надграждането и възможностите за комбиниране на готови стандартни или преподавателски решения.

Чрез избор от множеството графични примитиви в средата може да се представи който и да е алгоритъм. Ползите от използването на средата в обучението на начинаещи програмисти са в два аспекта: ползи при

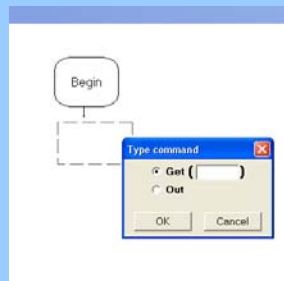
проектиране и разработване на даден алгоритъм и ползи от анимация на действието на вече разработен алгоритъм.

На етапа на проектиране и разработване за улеснение на начинаещите и предпазване от грешки при построяването на дадена блок-схема, в средата са заложени различни асистенции и контроли. За блоковете, които изискват вход от потребителя (блок за вход/изход, условен блок, блок за обозначаване на действие върху данните), е предвиден допълнителен прозорец, в който обучавания да избира от контекстно-зависими възможности (фиг. 3). Следи се в условния блок задължително да се разположи израз от логически тип.

Средата контролира логическата коректност и последователност при съставяне на алгоритъм и сигнализира потребителя при регистрирани противоречия. Обобщено може да се твърди, че на етап на проектиране и съставяне на дадена блок-схема ползите от средата са избягване на формални грешки от липса на опит, логически противоречия при разполагане на блоковете, пропуснати елементи от схемата.



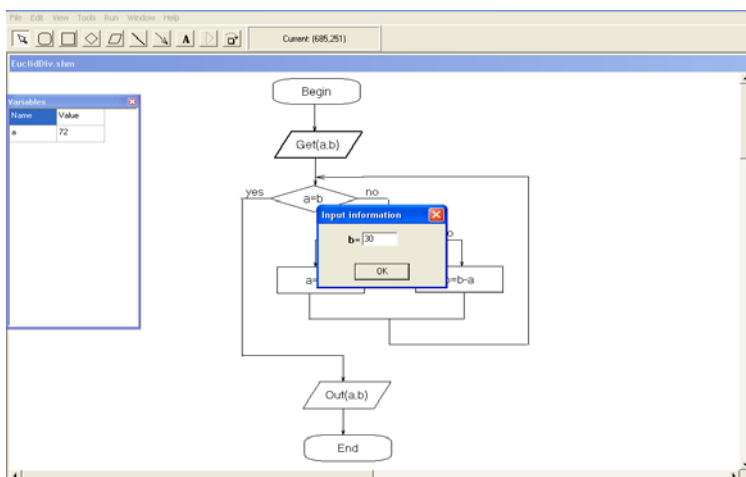
Фиг. 2 Общ изглед на средата



Фиг. 3 Асистенция при въвеждане на данни

За изпълнението на създадена блок-схема са предвидени две възможности: автоматично извеждане на резултата от действието на алгоритъма и постъпково (анимирано) показване на действието на всяка елементарна операция от алгоритъма върху данните и влиянието ѝ за получаване на крайния резултат. Анимацията на даден алгоритъм предоставя възможност за нагледно демонстриране на действието на всяка инструкция. За визуално въздействие и по-добър педагогически ефект блокът, в който тя е записана се маркира. В допълнителен прозорец е показан списък с имената и текущите стойности на използваните променливи (фиг. 4). Анимацията има особено влияние върху възможността за разбиране на действието на

основните алгоритмични конструкции и елементарни действия върху данните и носи съществената полза за начинаещите програмисти.



Фиг. 4 Анимация на алгоритъм и списък с използваните променливи

Заклучение и бъдеща работа

Несъмнени са трудностите, които начинаещите срещат в първите няколко упражнения или началото на първия семестър. Съществена е необходимостта от осигуряване на качество на обучението по програмиране и възможност за развитие на знанията и компетенциите до професионално равнище. Началото на обучението по програмиране е многокомпонентно и действа демотивиращо на начинаещите. Подходящо е да се отделят знанията относно алгоритмите и способностите за решаване на задачи. Използването на среда за обучение намалява значително бариерите и затрудненията при съставяне на абстрактни модели и алгоритмични решения и осигурява надеждна основа за надграждане на знанията в последващите дисциплини.

Използвана литература

- [1] БЪРНЕВ&ТОТКОВ&ДОНЕВА&ШКУРТОВ&ГЪРОВ, 2001, Информатика 9. клас, изд. Лета, стр. 135-235
- [2] Крушков, Х., М. Крушкова, В. Атанасов, М. Крушкова, 2010, Компютърна система за обучение по програмиране, Математика и математическо образование, Proceedings of the Thirty Ninth Spring Conference of the Union of Bulgarian Mathematicians, Albena
- [3] Biddle R and Tempero E, 1998, Java Pitfalls for Beginners, ACM SIGCSE Bulletin, Volume 30 issue 3, ACM Press, New York, pp: 48-52

- [4] Brown, M. H. and Sedgewick, R. 1984, A system for algorithm animation. In Proceedings of the 11th Annual Conference on Computer Graphics and interactive Techniques, SIGGRAPH '84. ACM Press, New York, NY - USA, pp: 177-186
- [5] DePasquale P, Lee J & Pérez-Quirñones A , 2004, Evaluation of Subsetting Programming Language, ACM SIGCSE Bulletin - Volume 36 - issue 1, ACM Press, NewYork – NY-USA pp 260 -264
- [6] Garner S, Haden P and Robins R, 2005, My Program is Correct But it Doesn't Run, Proceedings of the 7th Australasian Computing Education Conference, Australian Computer Society, Newcastle NSW Australia, pp: 173-180;
- [7] Kolling M, Patterson A, Quig B, and Rosenberg J, 2007, BlueJ, The Interactive Java Environment, Viewed 6th February 2007, <<http://bluej.org>>
- [8] Levy B, Ben-Ari M, and Uronen, P, 2003, The Jeliot 2000 Program Animation System, Computers and Education - Volume 40 – Number 1, Elsevier Science Ltd, Oxford – UK, pp:1-15
- [9] Lockwood L, Software for Use, ACM Press, 1999, ISBN 0-201-92478-1
- [10] McCracken, M. Almstrum, D. Diaz, M. Guzdial, D.Hagen, Y. Kolikant, C. Laxer, L. Thomas, I. Utting and T. Wilusz, 2002, A Multi-National, Multi-Institutional Study of Assessment of Programming Skills of First-year CS Students, SIGCSE Bulletin – Volume 33 – Issue 4, ACM Press, NewYork – NY –US, pp 125-140
- [11] Mclver L, 2000, The Effect of Programming Language on Error Rates of Novices, 12th Workshop of the Psychology of Programming Interest Group, Psychology of Programming Interest Group, Cozenza Italy, pp: 181-192;
- [12] Payne J & Myrers B, 1996, Usability Issues in the Design of Novice Programming Systems, School of Computer Science Technical Report CMU-CS-96-132, Carnegie Mellon University, Viewed 06 February 2007, <<http://web.cs.cmu.edu/~pane/ftp/CMU-CS-96-132.pdf>>;
- [13] RamalinghamV, LaBelle D, Weidenbeck S, 2004, Self-Efficacy and Mental Models in Learning to Program, SIGCSE Bulletin Volume 36 Issue 3, ACM Press, NewYork – NY –US, pp: 171-175
- [14] Robins A, Rountree J, and Rountree N, 2003, Learning and teaching programming: A review and discussion, Computer Science Education - Volume 13 Issue 2, Routledge, Oxford England, pp: 137-172
- [15] Soloway E and Spohrer J, 1989, Novice Mistakes Are The Folk Wisdoms Correct, Studying the Novice Programmer, Lawrence Erlbaum Associates, Hillsdale, New Jersey - USA, ISBN 0-8058-0003-4, pp: 401 – 146
- [16] Winslow L., 1996, Programming Pedagogy - A Psychological Overview, SIGCSE Bulletin – Volume 28 Issue 3, ACM Press, New York USA, 17-22
- [17] http://www.minedu.government.bg/opencms/export/sites/mon/top_menu/gener al/educational_programs/9klas/informatics_9kl.pdf