ВНИМАТЕЛНО С МОДЕРНИТЕ МЕТОДИ ЗА ОБУЧЕНИЕ И ОЦЕНКА

Теодоси Теодосиев

Шуменски университет ,Факултет по математика и информатика
Шумен, ул. “Университетска” 115
t.teodosiev@fmi.shu-bg.net

Абстракт: *В предлаганата работа се разглежда използването на тестовете и готовите програмни кодове в началното обучение по програмиране. Коментират се недостатъците на тестовата проверка на знанията на обучаемите. Във връзка с тези коментари се цитират данни за качеството на обучението в средното училище. Проблемите на началното обучение по програмиране се разглеждат в контекста на използване на модерните методи на обучение. Специално внимание е отделено на стила на програмата в процеса на обучение и оценка.*

Ключови думи: *обучение, тестова оценка, програмиране, стил на програма.*

Увод. Наближава отново “часът на истината” за абитуриентите и седмокласниците, т.е. провеждането на Държавни зрелостни изпити (ДЗИ), кандидатстудентски изпити и изпити за “елитните” гимназии. Те предизвикват много и разнопосочни дискусии в обществото. Надеждата е избраната форма – тестови въпроси с избираем и свободен отговор да обективизира и уеднакви критериите за оценка на знанията на учениците. Конструктивната част в тестовете по математика, БЕЛ и др. на ДЗИ намалява влиянието на шанса. Във връзка с тази нова форма естествено в последно време много се говори и залага на въвеждането на тестовете при проверка на знанията на ученици и студенти и в процеса на обучение. Акцентира се върху това, че тази форма е приета в много страни. Подчертават се предимствата на теста: стандартизира се изпита, повишава се обективността, автоматизира се проверката, увеличава се обхвата на изпитния материал и др. В част 1 на предложения материал са коментирани недостатъците на тестовата проверка на знанията. В част 2 се прави анализ на качеството на образованието у нас и по специално по математика на базата на резултатите от международни изследвания. В част 3 са разгледани проблемите в обучението по програмиране в контекста на стила на програмата. Коментирана е и ролята на стила на програмата в тестовете и готовите програмни текстове използвани в началното обучение по програмиране.

1. **Тестът като форма за проверка и оценка на знанията.** Дали обаче този нов начин на изпитване има само положителни ефекти? Дали това не се прави главно за повишаване на успеваемостта на фона на падане на нивото на образованието у нас? Защо се акцентира предимно върху тестове от затворен тип? Правя уточнението, визирам главно тестовете по математика и информатика! И двете науки са главно конструктивни и теста от затворен тип не способства за обучение в тази посока. Какво проверяваме (умения за налучкване на правилния отговор, късмет или знания)? Каква е целта на тестовото изпитване? Дали ни вълнува обективна оценка на качеството на обучението или постигане на удовлетворителни резултати за ученици, родители, просветни чиновници и преподаватели. Намаляването на слабите оценки, значи ли по-добро качество на обучението? Притеснително е, че отговорни фактори в образованието (ректор на реномиран столичен ВУЗ, председател на Съвета на ректорите) посочват, че тестът значително повишава успеваемостта (което било по-реално и приемливо)?!

“Няма подсказване, няма преписване, няма проверяващи комисии, няма субективизъм при оценките, а *самите оценки са готови на следващия ден* след изпита. При старата система нерядко двойки получаваха повече от 60% от кандидатите, сега те са около 14%. Вярвам, че по-справедливи са тези 14%” [1].

От казаното следва, че само смяната на формата на проверка повишава успеха. Звучи малко несериозно. Най-малко, защото при теста се разчита освен на знания, и на късмет, а и може да се постави „подходящ праг за успех” (справка: границата за успешно полагане на изпита в теста след 7 клас през 2007г- 16/100т. и прага за успешно положен ДЗИ от миналата година 23/100т). Публично Министърът на образованието и науката, успокои обществеността, че нежеланието на МОН да обяви предварително критериите за оценяване за предстоящите матури се прави, за да може след проверката критериите да удовлетворят всички. По отношение на ДЗИ, това ми се струва недопустимо, защото „зрелостен” предполага ниво на знания и умения на прага на зрелостта, а не съотносителност спрямо останалите. Показателно е, че голяма част от обучаемите също адмирират тестовете като форма на изпит (защо ли ?!). Много от университетите предлагат на кандидат-студентите възможност да се явят на тест или тема по една и съща дисциплина. Кандидатите, явяващи се на тест, са няколко пъти повече спрямо избралите тема!

2. **Качеството на обучението.** Анализът на резултатите от ДЗИ през 2008 година показват, че абитуриентите не могат (или не искат) да пишат свързани текстове. Има значителна разлика в

успеваемостта на избираеми или свободни отговори. Тези факти се потвърждават и от последните резултати на нашите ученици в международните изследвания PISA и TIMSS.

Според резултатите за България на Програмата за международно оценяване на учениците - PISA 2006, около половината от нашите 15-годишни ученици не могат да анализират и оценяват критично получената информация и да приложат наученото на практика.

Високият процент на зрелостниците, които не бяха отговаряли на задачите със свободен отговор, според интерпретацията на просветния министър Д. Вълчев показва, че “учениците не могат да мислят самостоятелно”.

“Тази "новина" изобщо не е нова и организаторите на матура 2008 можеха да я предвидят, ако си бяха направили труда да прочетат внимателно цитирания по-горе доклад за резултатите от PISA 2006. Според изнесените в приложенията към доклада данни, докато при задачите с изборен отговор средният процент на неотговорилите български ученици е приблизително 5%, то при задачите със свободен отговор този процент е над 40%, като при някои задачи дори надхвърля 50%” [2].

“Преди два месеца констатирахме поредния спад на България в TIMSS-2007. Този път той не е така голям, но резултатите на България са вече значително под международната средна стойност. В TIMSS са определени 4 нива на ученическите постижения - много високи, високи, ниски и много ниски. Данните от TIMSS-2007 показват, че по математика 26%, а по природни науки 24% от българските ученици не са достигнали дори до нивото с много ниски постижения. Описанието на нивото за много ниски постижения по математика изглежда така: Учениците имат елементарни познания за целите числа и десетичните дроби. Те могат да извършват основни пресмятания с тях. Имат познания за прости диаграми и графики. Забележете, че 26% от нашите ученици в VIII клас НЕ МОГАТ да правят това” [3]

Този проблем се задълбочава и в университета, където “.... ВУЗ-овете все повече заприличват на търговски дружества предлагайки това, което искат и за което плащат клиентите, вместо това, което повече би способствало за дългосрочната перспектива. Студентите се съсредоточават на това, което ще им даде най-добри шансове в търсене на работа, т.е. на овладяване на навици, необходими в дадения момент за работа в индустрията.

... Очевидно, ние сме в устойчив порочен кръг: преподавателите не могат да изменят своите курсове, тъй като трябва да привличат и удовлетворяват студентите; студентите искат това, което се практикува в промишлеността; а индустрията използва и възпроизвежда това, на което са обучени нейните работници” [8] (*не се отнася само за България, ако това може да е успокоение*- бел. моя*).*

3. **Стил на програма.** Само малцина имат известен опит за създаване на алгоритми – дисциплините в средния курс приучват учениците главно да разбират, запомнят и прилагат готови алгоритми. Нямат умения за детайлно описание на алгоритмите. Това води и до затрудненията при решаване на геометрични и текстови задачи по математика. Тук възниква още един проблем, свързан с модела на задачата и познанията на обучаемите в предметната област. 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.

Ето как в [8] е представена целта на обучението по програмиране:

“Що се касае за нашия предмет – информатика и компютърно програмиране – то крайната цел на образователните учреждения трябва да бъде доста по-широка от овладяване на някакъв език за програмиране. Тя трябва да бъде не по-малко от изкуството на проектиране на артефакти за решаване на сложни задачи. Понякога наричат това изкуство на конструктивното мислене”.

Този цитат от Н.Уирт както и споменатите по-горе проблеми (умение за създаване на модел, съставяне на алгоритъм, слаба общообразователна подготовка и др.) недвусмислено показват, че тестовите въпроси с избираем отговор не са подходящи, когато обучаваме в конструктивно мислене.

Този вид въпроси са подходящи за проверка на овладяване на синтаксиса и семантиката на ЕП (формалните аспекти), но неудачни за съставянето и разбиране на алгоритмите (творческите аспекти на програмирането).

В последните години, 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 програмни текстове имат приложение и в модерния начин за изпитване – тестовете. Доколкото тези форми са удачни в началното обучение по програмиране (според мен използването им трябва да е ограничено), те поставят въпроса за стила на използваните кодове.

Главният тезис се състои в това, че стилът на програмиране – това е стил на мислене, проявяващ се в умението да се изобрази алгоритъма за решение на задача на конкретен ЕП.

Много често коренът на злото е в неумението да се построи ясен и адекватен модел на решаваната задача, в неспособността да се отчетат всички факти, накратко – в неумението да се мисли ясно.

Следователно, главното, на което трябва да се отделя внимание е - развитието на мисленето и неговия стил в процеса на решаване на задача с компютър [5].

Курсовете по програмиране не бива да приличат на шофьорски курсове, които ни учат да управляваме автомобила, вместо да ни учат, как да използваме автомобила, за достигане на нужната цел.

Преподавателят пише програми с цел обучение, т.е. те трябва да са лесни за разбиране. При обучението този въпрос е от съществена важност, независимо че рядко на преподавателя му остава време и желание за такива коментари.

В програмирането стилът до голяма степен е в това да постигнем единство между формата и съдържанието на програмата. Формата трябва да илюстрира, да обяснява и документира съдържанието и замисъла.

В обучението вниманието трябва да се съсредоточи върху четивността на програмния код дори за сметка на ефективността при изпълнение. Четивността е интересна, защото ние четем, за да се учим и пишем, за да съхраним информацията, с чиято помощ след това да пишем.

Не трябва да пропускаме нито една възможност при обучението, в която да се направи естествен преход към коментари как написаната програма да стане по-вярна, по-лесна за четене, по-дружелюбна, а защо не и по-ефективна (стига да не е за сметка на яснотата й).

В [4] известният специалист в областта на информатиката Е. Дейкстра коментира така качествата на програмиста:

“Да бъдеш отличен програмист – означава да си способен да разработваш по-ефективни и достоверни програми и да знаеш как да го правиш ефективно. Към това се отнася икономията на клетки от паметта или машинни цикли, а също избягване на сложности, които увеличават разсъжденията, необходими за удържане на строго интелектуалния контрол над разработките. Това, което според мен е нужно за това е, усъвършенстване на математическите навици, при това аз използвам думата „математика” в смисъл „изкуство и наука за ефективни разсъждения”.

В [9] този коментар се доразвива:

“Да се обучи новото поколение програмисти с понижен праг на търпимост към сложността и да ги научим да търсят наистина простото решение това е един от най-важните проблеми в нашата област.... Как да убедим хората, че в програмирането простотата и яснотата – накратко: е това което математиците наричат елегантност – това не е излишен разкош, а жизнено важен въпрос, който определя избора между успеха и провала. “

В действителност задачата за разработка на висококачествени програми и построяване на висококачествени доказателства са толкова сходни, че не могат да се видят смислени различия между методологията на програмирането и математическата методология в общи линии [4].

Следването на правилата на добрия стил на програмиране значително намаляват вероятността от появяване на грешка на етапа на набиране на текста, прави програмата лесно четивна, това на свой ред облекчава процеса на проверка и внасяне на изменения. Тези правила са изключително важни в контекста на готовите програмни кодове и тестовете.

✓ *Простота.* Стилната програма е написана просто. Не трябва програмата да се превръща в ребус, пълен с хитри трикове и необичайни конструкции, да се използуват средствата на езика и операционната система за несвойствени цели [5]. Логиката трябва да е достъпна, използваните средства да са познати, изразите да са естествени, текстът да е малък по обем.

✓ *Прегледност.* Стилната програма се чете лесно. Текстът на програмата трябва да се подреди така, че да следим логиката й, да четем отгоре надолу. Това означава основната логика да е изнесена в главната програма (функция), а по-подробната логика да бъде в подпрограмите. Йерархията на текста да отразява йерархията на логиката.

Важна роля играе подреждането и групирането на входния текст по такъв начин, че структурните единици на програмата – блокове, съставни оператори, подпрограми лесно да се отделят визуално (напр. с използване на празни редове). Когато програмата има структура (взаимна вложеност на частите) това се постига най-добре, като началото на редовете се отмества надясно [6].

✓ *Коментиране и стандартизация.* Стилната програма е добре коментирана. Белег за добър стил е спазването на общовалидни правила и съглашения при програмирането, използването на носещи смислова стойност имена на променливи и функции. Важен елемент на програмния стил са коментарите. Коментарите трябва да поясняват само неща, които не са очевидни и да помагат на програмиста да се ориентира в алгоритъма и действието на програмата. При модулен метод на програмиране коментарите играят роля за поясняване на подпрограмите [6].

✓ Добрата програма трябва преди всичко да е надеждна и дружелюбна към потребителя. Програмата не разчита на „разумното” поведение на потребителя, а контролира входните данни, проверява резултатите от изпълнение на операции, които по някаква причина могат да не бъдат изпълнени.

Не трябва да се забравя, че надеждността в програмите се определя главно от качествата на програмиста, от неговата квалификация и интелект.

Някои характеристики на стила обаче са противопоказни поради спецификата на тестовата проверка на знанията. Неудачно е използването на коментари, мнемонични имена, ефективни изчисления, надеждност и дружелюбност. За сметка на това са изключително важни – подредеността, четивността, яснотата на алгоритъма и др.

4. **Заключение.** Спомагат ли готовите програмни текстове за развиване на мисленето на обучаемите? Каква е ползата от показването и коментарите върху вече написани и обмислени „добри” (умен съм бил, сетил съм се) програми?

Мнението по този въпрос на Е. Дейкстра е:

“Не ме удовлетворява и това, че алгоритмите се публикуват като готови изделия, без разсъжденията проведени в процеса на разработка и служещи като обосновка за окончателния вид на програмата”[7].

Важен е процесът на създаване, включително и корекциите на програмата. Това има и възпитателна роля за обучаемите, защото им показва, че написването на “хубава” програма не е еднократен акт, а трябва да се обмисли преди да се седне на компютъра, няма нищо страшно, ако се сбърка. Привърженик съм на съставянето на програмата пред обучаемите с използване на лаптоп с мултимедиен проектор или дори на бялата дъска, съпроводено от коментарите на преподавателя. Така те могат да проследят процеса на програмиране дори и корекциите свързани с подобряване на верността и стила на програмата.

Готовите програмни кодове са подходящи за обучението по структури от данни и ООП, където текстовете са много дълги. Там тежестта пада върху операциите със структурите и класовете.

Цитираното по-горе мнение на Дейкстра както и резултатите от моя опит ме убедиха, че не е удачно използването на тестове с избираем отговор в началното обучение по програмиране. На този етап тестовете стимулират „играта на тото”. Докато давах тестове от затворен тип (зададен е програмен фрагмент и има избираем отговор) обучаемите налучкваха (и понякога печелеха). За съжаление при поставяне на задача за самостоятелно съставяне на програма резултатите бяха слаби (в повечето случаи няма дори опит за писане на програма). По тази причина започнах да задавам тестове от отворен тип и там резултатите са съпоставими (еднакво трудни с цялостната задача). Смятам за най-удачно, да се използват тестове в комбинация с традиционните задачи за съставяне на програми.

Литература

[1]. Тестовете намалиха двойките 4 пъти, в. Стандарт /09.05.2007
[2]. И МОН заслужава двойка за работа с научни данни в. Сега /16.03.09
[3]. Докога ще затъваме по математика, в. Сега /23.02.09
[4]. Дейкстра Е, Почему американская информатика кажется неизлечимой, 1995
[5]. Боровин Г., и др., Ошибки-ловушки при програмировании на Фортране, „Наука”, Москва, 1987
[6]. Боровски Б. и др., Справочник по изчислителна техника- програмиране и програмно осигуряване на ЦЕИМ, Техника, София,1990
[7]. Дейкстра Е, Почему программное обеспечение такое дорогое?
Wirth N, Computing Science Education: The Road not Taken, ITiCSE Conference, Aarhus, Denmark, 2002, http://www.inr.ac.ru/~info21/texts/2002-06-Aarhus/en.htm