
МЕТОДИЧЕСКИ ПРОБЛЕМИ И ТЕХНИТЕ РЕШЕНИЯ ПРИ ПРЕПОДАВАНЕТО НА ПРОГРАМИРАНЕ В ЗАДЪЛЖИТЕЛНА ПОДГОТОВКА НА ГИМНАЗИАЛНИЯ ЕТАП НА БЪЛГАРСКОТО УЧИЛИЩЕ

СТЕФАН Й. ПЕЙЧЕВ

METHODICAL PROBLEMS AND THEIR SOLUTIONS IN TEACHING PROGRAMMING AT THE COMPULSORY EDUCATION COURSE IN BULGARIAN SECONDARY SCHOOLS

STEFAN Y. PEYCHEV

***ABSTRACT:** Teaching programming at the compulsory education course is at a stage when it is necessary to analyze and synthesize the positive experience of a large number of practicing teachers. Some methodical problems of teaching programming as well as their solutions based on the author's practical experience are described in this article.*

***KEYWORDS:** teaching programming, methodical problems and solutions*

Обучението на компютърни специалисти не е лека задача, тъй като сферата на информатиката и информационните технологии непрекъснато се развива. Необходимо е непрекъснато усъвършенстване и обогатяване на учебното съдържание по информатика и информационни технологии, осъвременяване на учебния процес чрез прилагане на иновативни методи и средства за обучение. Тези фактори от своя страна допринасят за формиране и изграждане на конкурентноспособни специалисти-информатици с изградени професионални компетентности, които успешно да се справят с предизвикателствата на информационното общество и динамиката на съвременния свят [1].

Поради факта, че обучението по програмиране в средното училище е въведено сравнително скоро все още липсват методически разработки, които преподавателите по информатика да използват при организация на учебния процес по предмета и в момента е на етап, когато е необходимо да се анализира и синтезира положителния опит на голям брой практикуващи учители.

В тази статия са синтезирани някои от методическите проблеми, които автора е срещнал през годините, както и предложения за тяхното преодоляване.

Програмирането като част от учебния предмет Информатика се изучава в 9 клас съгласно сега действащия учебен план. За успешното усвояване и реализация на тази човешка дейност се предполага наличието на специфичен вид мислене – алгоритмично мислене [7]. Основната задача при началното обучение по алгоритми е усвояването от обучаемите на базови алгоритми, които да не се обмислят по време на решаване на поставената задача, а да се прилагат директно. Обучените по този начин печелят много време, тъй като се съсредоточават върху решаване на същинския проблем, посредством използване на готов инструментариум. Този подход превръща решаването на една голяма по обем и сложна задача в логическо подреждане на вече готовите по-прости модули [8].

Въпреки, че от обучението по математика и физика учениците са свикнали да съставят алгоритми и формално да следват стъпките на предварително изучен алгоритъм едва тук те срещат научната дефиниция на това понятие.

Практиката показва, че понятията елементарно действие, стъпка, алгоритъм и подалгоритъм по-лесно се усвояват от учениците, ако са въведени чрез решаването на конкретна задача.

И тук се появява основният проблем не само в изучаването на тази тема, но и в последващото обучение по Програмиране. Без да се коментират понятията променлива и компютърна памет учителят трябва да въведе като част от алгоритъма действието „Задаване на стойност на дадена величина /константа или променлива/“, което предизвиква недоумението на учениците.

Моят опит показва, че рядко в някой випуск има един или двама ученика, които успяват бързо да вникнат в смисъла на това действие.

Учителят може да използва няколко различни подхода за преодоляване на този проблем.

Първият подход е да се даде указание на учениците за „наизустяване“ на това действие и прилагането му при решаване на задачи, без да се налага допълнително обяснение. Смисълът на действието се доизяснява с първите уроци в темата „Основи на програмирането“.

Вторият подход, който отнема повече учебно време, и който аз успешно прилагам в часовете по Информатика е чрез решаване на следната задача:

Решете уравнението

$$(1) ax+b = 0 \text{ /линейно уравнение/}.$$

Учениците в 9 клас идват от различни основни училища и съответно притежават различна степен на математическа подготовка. Намирането на баланс между достатъчно на брой упражнения, които да са полезни за учениците и количеството на преподавания нов материал се превръща в основна задача за преподавателя. Разбира се, ако обучението е в някоя от специализираните гимназии /математически, езикови и др./, където учениците биват приемани след съответния изпит по математика това различие в нивото на подготовка не е толкова ярко изразено. Изборът на това кой от двата подхода ще се използва е предоставен на преподавателя и зависи от степента на математическа подготовка на учениците в класа.

Логично е следващата задача да бъде за съставяне на алгоритъм за решаване на квадратно уравнение:

$$(2) ax^2+bx+c=0$$

Изучаването на този алгоритъм обаче върви паралелно по дисциплините Математика и Информатика и учениците все още не са затвърдили знанията си за решаване на пълни и непълни квадратни уравнения. Поради тази причина е необходимо да се направи едно много добро разпределение на учебното време и съгласуваност между преподавателите по двата учебни предмета, разглеждането на алгоритъма в часовете по Информатика се явява естествено продължение на математическите знания.

При клас с недостатъчно подготвени по математика ученици учителят трябва или да „иземе“ функциите на преподавателя по математика, което не е препоръчително и представлява сериозна методическа грешка или изобщо да не решава тази задача.

Различията в първоначалната подготовка на учениците създават и доста други предизвикателства пред преподавателя по Информатика, които няма да разглеждаме подробно в тази статия, като напр. съставянето на алгоритми за решаване на построятелни задачи и др.

Препоръчително е след разглеждане на различните начини на описание на алгоритмите и видовете алгоритми всеки алгоритъм да бъде представен, чрез подходяща задача, която да бъде решена, тествана и подробно анализирана.

Съдържанието на учебния предмет Информатика в средното училище се определя чрез Държавните образователни изисквания [4], които формулират пет ядра на учебното съдържание: Информация и формални модели, Компютърни системи, Операционни системи, Алгоритми и структури от данни и Програмиране. Ядрото програмиране включва усвояване на конкретен език за програмиране, но стандартите на ДООИ не конкретизират кой език да се изучава. Тенденцията е усвояването на програмния език С или С++.

Още от първия урок по темата Програмиране трябва да убедим учениците, че това, което ще научат, е важно, трябва да им покажем всички сфери, в които може да се приложи. Хубаво е да грабнем вниманието им и да увеличим ентузиазма им, като им покажем готови приложения, изготвени чрез съответния език за програмиране. Това веднага ще им докаже, че това, с което ще се занимават е модерно и приложимо [6].

Като пример в тази посока може да се посочи програма, която извършва основните математически действия – събиране, изваждане, умножение и деление.

След демонстриране на работещата програма е удачно на учениците да се даде възможност сами да наберат програма на С++, която да стартират и тестват. Въпреки, че това което ще наберат на компютрите си е ново и напълно неразбираемо у тях остава усещането, че сами са създали компютърно приложение. Този факт предизвиква ентузиазъм и практиката показва, че в повечето случаи е достатъчно да предизвика интерес към програмирането.

При началното въвеждане в темата трябва да се отчетат и специфичните възрастови и психологически характеристики на учениците. За разлика от студентите, които сами са избрали да изучават програмиране, на учениците това е наложено по задължение, а на тази възраст /15 – 16 години/ те естествено се бунтуват срещу всичко, което им е наложено насила. И тук ролята на учителя е много важна. Защото загуби ли интереса на учениците още в първите часове, то много трудно ще си го върне по-късно. Учениците ще зазубрят новите понятия за да изкарат отлични оценки, но без никаква практическа полза от това.

Учителят е поставен пред избор: да „препусна“ през учебния материал като сведе до минимум часовете за практически упражнения с цел предаване на всички предвидени теми или да не изпълни изцяло учебната програма, но в края на курса на обучение учениците да са усвоили в максимална степен основите на програмирането.

Някои автори [2] навлизат в темата с директно разглеждане на готова програма – т. нар. програма “Hello, World!” – придобила популярност като първата програма при изучаване на нов език за програмиране. Други [5] започват с описание на основните елементи на езика – азбука, видове думи, оператори, коментари, структура на програма и т. н.

И двата подхода имат своите преимущества и недостатъци. Първият подход от една страна предлага възможност веднага да се набере и изпълни програма на С++ и чрез нейното разглеждане да се дефинират основните понятия в езика. Вторият обаче предполага по-трайно усвояване на елементите на езика преди практическото им приложение.

Годините практически опит са доказали, че най-успешна е комбинацията от двата метода като малък превес се дава на втория. Ако в часовете по Информатика държим твърде дълго учениците далеч от компютрите те много бързо загубват интерес към дисциплината.

Поради тази причина е полезно да се набележат основните структурни компоненти на езика и да се премине към написването и анализа на проста програма.

Връзката с други дисциплини от учебния курс – математика, физика и др. – с информатиката тук отново е много силна. Обучаемите вече са запознати с термините променлива и константа. Знаят и множествата от числа – естествени, цели, реални. Новото и обикновено трудно за разбиране е абстрактното понятие „тип на променливата“. Тук на учителя отново се налага да обяснява процес, който остава скрит за учениците, а именно съхраняването на информация в паметта на компютъра.

Отлични резултати показва и използването на интерпретатор на C++ вместо компилатор, или интегрирана среда за разработка при първите стъпки в програмирането. Като пример за такъв интерпретатор може да бъде посочен *UnderC* разработен от Стийв Донован [3].

Всеки преподавател си е задавал поне веднъж въпроса защо някои ученици използват компютрите за игри, а други участват активно в час и по-лесно усвояват преподавания материал? Една част от учениците участват активно в часа, като следят указанията на учителя и решават задачи. Проучвания показват, че тези, които сърфират из интернет или играят, не могат да участват в часа, защото не разбират материала и не могат да решават задачите [6].

Когато учениците разбират материала, за тях и изучаването на програмиране и създаването на компютърни програми става удоволствие. Ако преподавателят не обяснява добре или дава прекалено сложни задачи, учениците се настройват враждебно към него и дисциплината му. За това биха допринесли по-интересни задачи, разбирането на материала, упражненията, преподавателя или просто личните предпочитания на ученика [6].

За преодоляването на тези проблеми учителя трябва:

- внимателно да проучи предварителната подготовка на учениците от съответния клас
- да избере подходящия подход за преподаване в зависимост от тази подготовка
- да работи активно с преподавателите по Математика и Физика
- да избира атрактивни примери
- да решава много конкретни задачи свързани с реалния живот
- да се прави внимателен и подробен анализ на условието на всяка задача
- при решаването на задачи да се преминава през всичките осем етапа за решаване на задачи с компютър /да не се пропускат етапи/;
- по-голямата част от създадените програми да се проиграват „ръчно“;
- да се използват подходящи нагледни средства – таблици, диаграми, информационни карти и др.;

Практиката показва, че е по-удачно в първите часове по програмиране да се използва интерпретатор на избрания език за програмиране, а не компилатор или среда за разработка. Така учениците много по-бързо и лесно усвояват новите понятия и навлизат в същността на езика.

От една страна учителят трябва да събуди интереса на учениците към програмирането и впоследствие да успее да го задържи. Тази задача може да бъде изпълнена чрез решаване на подходящи задачи, като се започне с по-лесните за решаване и се премине към по-сложни, които изискват специфичен вид мислене.

От друга страна той е длъжен да се съобрази с индивидуалните потребности и интереси на учениците.

Елементите на индивидуалния подход се проявяват като:

- Идентификация на близките педагогически задачи при работа с всеки ученик
- Фиксация и анализ на получените резултати

- Избор или промяна на най-ефективните средства при прилагане на индивидуалния подход спрямо конкретния ученик
- Системно изучаване на всеки ученик
- Поставяне на нови педагогически задачи

Въпреки, че индивидуализацията не реализира учебната дейност в пълен обем, в определени случаи е необходима, дори задължителна за да се постигнат минималните ДОИ.

Един вариант на формална организация на учебния процес е диференцирано обучение в рамките на класа на групи с различни нива на надареност по специални програми.

И не на последно място – предвиденото в учебните програми време за изучаване на програмиране е крайно недостатъчно. Добрият преподавател трябва да прояви гъвкавост и умение за да успее да преподаде основите и всички ученици, независимо от тяхната първоначална степен на подготовка, да покрият в максимална степен Държавните образователни изисквания.

ЛИТЕРАТУРА

1. **Анева** Ст., Модел за профилирано обучение по Информатика и информационни технологии в гимназиален етап, Автореферат на дисертация за присъждане на образователна степен „доктор“ в област на висше образование, Пловдив, 2013 г.
2. **Азълов** П., Информатика, профилирана подготовка, Просвета, 2005 г.
3. **Донован** С., С++ в примери, СофтПрес, 2004 г.
4. Държавни образователни изисквания по Информатика
5. **Иванова** Л. и др., Информатика – втора част, Програмиране на С++, Коала Прес, 2006 г.
6. **Крушков** Хр., Крушкова М., Крушкова Д., Как да подобрим обучението по програмиране, ФМИ при ПУ „Паисий Хилендарски“, Национална конференция „Образованието в информационното общество“ 2010
7. **Терзиева** Т., Развитие на алгоритмичното мислене в обучението по информатика, Автореферат, Пловдив, 2012 г.
8. **Христова** П., Атанасова Г., Подготовка на ръководители на школи по информатика, Доклад на Национална конференция „Образованието в информационното общество“