

*МАТЕМАТИКА И МАТЕМАТИЧЕСКО ОБРАЗОВАНИЕ, 2007
MATHEMATICS AND EDUCATION IN MATHEMATICS, 2007
Proceedings of the Thirty Sixth Spring Conference of
the Union of Bulgarian Mathematicians
St. Konstantin & Elena resort, Varna, April 2–6, 2007*

**СПИРАЛОВИДНО ОБУЧЕНИЕ ПО ПРОГРАМИРАНЕ НА
10-11 ГОДИШНИ ДЕЦА (НА БАЗАТА НА ЕЗИКА C++)***

Бисерка Б. Йовчева

Статията представя уникален подход при обучението по програмиране при 10-11 годишни ученици, който се базира на идеята, че обучението на децата за писане на алгоритми трябва да започне веднага след преподаването на първите елементи на езика за програмиране и да се развива с времето, като един и същ проблем се разглежда неколккратно, но всеки път с добавяне на нови знания. По описаната методика е издаден учебник, на базата на който е организирано експериментално обучение на ученици в продължение на четири години.

Спираловидния подход в обучението като дидактическа концепция. Изграждането на дадено понятие или система от понятия чрез многократно връщане към него или към системата на различни етапи от процеса на обучение е често прилаган подход. За съжаление, трудно може да се намери систематизирано теоретично описание на проблемите, възникващи при неговото прилагане. Могат да се посочат много примери за спираловиден подход при поднасянето на различни теми, по различни учебни дисциплини. Понятието “естествено число” в математиката, например се използва за първи път още в първи клас, но във всеки следващ се допълват нови свойства и характеристики на естествените числа. Подобен е и примера с изучаване на стереометрията в училищния курс по математика – за първи път основни понятия в стереометрията се въвеждат в VI клас, като след прекъсване от няколко години, те се доизграждат в гимназиалния курс. В приетите от МОН учебни програми по информационни технологии от V до VIII клас [14] е застъпен същият подход. Там темата “Компютърни системи”, например, се въвежда в V клас, но се надгражда в следващи години на обучение.

Независимо от това, че както вече споменахме, спираловидният подход не е описан подробно като дидактическа концепция в литературата, можем да споменем основните аспекти при неговото прилагане:

– На базата на този подход дадено понятие или тема се поднася за първи път, без да се изчерпят напълно техните характеристики и описание. На един по-късен етап, след като знанията и практическият опит на учениците са достигнали определено

*Разработката на статията е финансирана от Шуменски университет “Епископ Константин Преславски”, Фонд “Научни изследвания”, Факултет по математика и информатика (Проект № 28/2006 г.)

Ключови думи: обучение, информатика, дидактика

ниво, се прави ново връщане към същото понятие и на базата на известните до момента свойства, то се обогатява и разширява.

– Обемът на поднесените учебен материал на даден етап от обучение (при по-редното завъртане на спиралата) се определя от възрастовите особености, знанията и жизнения опит на учениците.

– Много е важна и “хоризонталната” организация при поднасяне на различните понятия, т.е. връзките, които съществуват между различните понятия, поднасяни спираловидно, да се определят така, че на определен етап от завъртане на спиралата, учебното съдържание да има завършен вид.

2. Някои примери за приложението на спираловидния подход в обучението по програмиране. Учебната програма, на която е базиран представеният подход, е разгледана в други статии на автора ([10] и [11]). Тук целта е да се опишат някои конкретни идеи и реализации на спираловидния подход в обучението по програмиране на 10-11 годишни деца. Те се основават на възрастовите особености на малките ученици, на възможностите им за обобщения и абстрактно мислене, които играят много важна роля при обучението по програмиране.

Все още в голям процент от случаите дебатът кога да започне обучението по програмиране се свързва с алгоритмичния език, на който ще се провежда то. Позицията на автора по този въпрос е изложена в друга статия [8] и тук няма да се спираме подробно на нея. Изводът е, че езикът C++ е едни от най-подходящите. Тук ще приемем този извод без допълнителен коментар и ще се съсредоточим върху конкретиката на поставения проблем.

В резултат на училищното и извънучилищно обучение точно на 10-11 годишна възраст става качествена промяна в мисленето на детето – от конкретно индуктивно, то постепенно включва и елементи на абстрактно дедуктивно (откриване на закономерности и обобщение, пространни логически връзки, необходимост от аргументация, наченки на алгоритмизиране).

На 10-11 годишна възраст децата имат достатъчно фактологически знания по други учебни дисциплини и най-вече по математика.

На тази основа може да се направи опит за първи стъпки в програмирането. Опитът за преподаване на програмиране в училище в световен мащаб, за съжаление, не е много голям. Обикновено то се провежда по начин, подобен на този във висшите учебни заведения, тъй като до скоро това беше единственото място, където се осъществяваше подобно обучение. Като всяка университетска дисциплина, и програмирането изисква пълно изчерпване на фактологията и проблематиката на разглежданата тема. При излагане на учебното съдържание в университетските курсове с основно предимство се ползва принципът за научност, като принципите за достъпност и нагледност остават на заден план. Пренасяйки този подход в училищния курс, голяма част от преподавателите предварително обричат на неуспех опита да поднесат по достъпен начин една твърде абстрактна теория.

Вече споменахме, че на 10-11 годишна възраст в мисленето на учениците започват постепенно да се наблюдават наченки на алгоритмизиране и обобщение, но те са все още в начален етап и подлежат на непрекъснато развитие. От своя страна предметът на обучението по програмиране е създаване на алгоритми, чиято основна характеристика е “масовост”, т.е. възможност за решаване на колкото е възможно

по-голям брой задачи. Това автоматически означава високо ниво на абстракция и параметризация при решаването на поставените задачи. Затрудненията на малките ученици са именно в тази насока. Те успешно възприемат синтаксиса на различните езикови конструкции, но се затрудняват при осмислянето на семантичните им особености и от там на тяхното прилагане.

Точно тази особеност във възможностите на обучаемите изисква разделяне на учебния материал на малки, лесни за възприемане части и усвояването им с помощта на множество примери, в резултат на което сравнително бавно, но надеждно се изграждат първите навици за създаване, кодиране и използване на различни алгоритми.

Какви знания са необходими на един обучаем, за да може да напише първата си програма? Като изключим стандартното "Hello, World!", което в общия случай се използва при запознаване с всяка среда за програмиране, за да може да напише и най-простия алгоритъм, обучаемият трябва да се е запознал и усвоил следните основни понятия:

- елементи на езиците за програмиране – азбука, идентификатори, величини (константи, променливи и литерали), оператори и др.

- програма и структура на програма;

- типове данни, операции и изрази,

както и елементарни средства за въвеждане и извеждане на данни.

На тази основа могат да се разглеждат и описват тъй наречените линейни алгоритми и точно тук е добре да се поставят основите на изграждането на бъдещото алгоритмично мислене. Могат да се представят някои стандартни алгоритми, които се разглеждат при минимална степен на абстракция. Целта е да се сведе неизбежната параметризация до минимум, като се зададат колкото може по-конкретни стойности на елементите на алгоритъма. Например, ако трябва да се обработват числа, те трябва да са определен брой – 3, 4 и т.н., а не N , където N е произволно естествено число.

Да разгледаме примера с отделяне на цифрите на число, който има голямо приложение в програмирането. Този проблем има няколко нива на параметризация: първото е, че самото число се записва като променлива (естествено означена с буква), което само по себе си вече затруднява учениците, свикнали да решават задачи само с конкретни данни. Второто ниво на параметризация е отделянето на цифрите на число с фиксиран брой цифри – например цифрите на трицифрено число. Третата степен на параметризация е предварително неизвестния брой цифри и тяхното отделяне.

В повечето учебници описаният проблем се представя за решаване в последния му вариант – дадено е цяло положително число с неизвестен брой цифри, при което се иска да се състави програма, която да намери цифрите на числото, записано със същите цифри, но в обратен ред и др. За да реши подобна задача, обучаемият трябва да е запознат с голяма част от езиковите конструкции – оператори за цикъл, за проверка на условие и др. Подходът, който излагаме тук, предполага първото запознаване с посочения проблем да е във варианта с фиксиран брой цифри. В този случай могат да се формулират множество интересни задачи, които помагат на учениците да усвоят *само* техниката на отделяне на цифрите на число и да я прилагат, без да се налага на този етап да следят за организация на цикли, евентуално

натрупване на резултати и др. Ето няколко примера за такива задачи:

Задача 1. Сума от цифри на число [13]

Дадено е цяло трицифрено число k . Да се състави програма **SUMCIF.CPP**, която прочита от клавиатурата числото k и извежда на екрана сумата от цифрите му.

Примерен вход

365

Примерен изход

14

Задача 2. Намиране на огледалното число на дадено [13]

Да се състави програма **OBRATNO.CPP**, която прочита от клавиатурата цяло, трицифрено число K и извежда на екрана огледалното му (числото, записано със същите цифри, но в обратен ред).

Примерен вход

365

Примерен изход

563

Задача 3. Число [13]

Да се състави програма **NUMBER.CPP**, която прочита от клавиатурата цяло четирицифрено число k и отпечатва сумата от числата, които се получават като се задраскат съответно втората и третата цифра отляво надясно на числото k .

Примерен вход:

1234

Примерен изход:

258

В тази поредица могат да се посочат няколко задачи, давани на национални турнири по информатика за група Е: **Задача Е1. ЧИСЛА**, от Национален есенен турнир по информатика и информационни технологии “Джон Атанасов” – Шумен’2005; **Задача Е2. СЕЙФ**, от същия турнир – 2004, **Задача Е1. ОБРАТНО ЧИСЛО**, от Зимни математически празници – Бургас’2005 и др. [12]

Към тази, вече позната тема, учениците се връщат отново по-късно, когато вече са запознати с операторите за цикли, механизмите на натрупване на суми в една променлива и др. Тогава се поставя за разрешаване проблема с обработка на естествени числа, броят на цифрите на които не е известен предварително.

Друг подобен пример са задачите за намиране на даден оптимум. Отново може да разграничим поне две нива на решаване на проблема: намиране на оптималния от точно определен брой случаи (например 3) и впоследствие – същата задача, но без да е известно предварително колко случая ще се обработват. Ето пример за такава задача, която после може да се разгледа в общия случай:

Задача 4. Състезание [13]

В състезание по вдигане на тежести при равни резултати на по-предно място се класира по-лекият състезател. Да се състави програма **COMP.CPP**, която прочита

от клавиатурата резултатите и теглата на трима състезатели и отпечатва постижението и теглото на златния медалист.

Примерен вход:

180 67

180 66

170 70

Примерен изход:

180 66

По подобен начин стои и проблемът с алгоритмите за сортиране. Когато трябва да се сортират три или четири числа, на учениците трябва да се поднесе стандартен метод за сортиране, независимо че на този етап те не са запознати нито с масиви, нито с цикли. Познаването на даден алгоритъм за сортиране в частен случай само ще улесни изучаването му в абстрактния вид, където се създава алгоритъм, работещ за произволен брой елементи.

Подобно приложение на описвания подход има и при преподаването на темата за текстообработка, в частност – запознаването на учениците с типа **char**. Първото разглеждане на този тип включва описание на стойностите му, средствата за вход и изход, както и различните операции, които могат да се извършват със знаци – отпечатване на следващата поред буква, превръщането на малки латински букви в главни и обратно, изчисляване на стойността на цифра и др. Разглеждането на тези операции на този етап като самостоятелни действия, без да са включени като част от някакъв алгоритъм, прави осмислянето им по-лесно. Когато по-късно се разглеждат алгоритми за обработка на текст, детайлите необходими за описанието им вече са познати на учениците и те могат да се съсредоточат в същността на алгоритъма.

Друга тема, която е подходяща да се поднася спираловидно е темата “Компютърна геометрия”, която се въвежда на няколко етапа:

– Първи етап: непосредствено след запознаването с всички аритметични операции и средствата за вход-изход. Тогава се разглежда само понятието координатна система, екранна координатна система и се определя разстоянието между две точки с равни абсциси или с равни ординати. Разглежда се за първи път и правоъгълник със страни, успоредни на координатните оси и се намират неговото лице и периметър.

– Вторият етап е след запознаването на обучаемите с условен оператор **if** и възможностите за обработка на различни случаи. Тогава се разглежда определянето на местоположение на точка спрямо зададена фигура: квадрантите в координатната система, координатните оси, правоъгълник със страни, успоредни на координатните оси. На този етап наборът от езикови конструкции, който учениците познават, е достатъчен за писане на програми за определяне на взаимно положение на два правоъгълника със страни, успоредни на координатните оси.

– Третият етап изисква познаване на оператори за цикъл и включва обработка на последователно въведени точки в равнината, зададени със своите координати, преброяване на точките с цели координати, които принадлежат на даден правоъгълник или група правоъгълници (разбира се, правоъгълниците са със страни, успоредни на координатните оси).

3. Експериментални резултати, изводи и препоръки. В заключение ще обобщим предимствата на описания подход:

– запознаването с конструкциите на езика за програмиране става постепенно с много различни примери и има достатъчно време, за да бъдат добре осмислени и усвоени от учениците.

– запознаването с основните алгоритми за програмиране в частни случаи води до по-добро им осмисляне и възприемане в общия случай.

– предварителното разглеждане на детайли от по-сложни алгоритми води до успешно разбиране и реализиране на тези алгоритми на един по-късен етап.

Подходът, който предложихме в настоящата статия е експериментиран в продължение на четири години в Школа за извънучилищно обучение по математика и информатика в гр. Шумен с ученици на възраст 10-12 години. За резултатите от прилагането му ясно говорят успехите на учениците от Школата на всички национални ученически състезания в България през последните години [12]. Издаден е учебник, базиран на описаната методика, съдържащ множество решени задачи и задачи за упражнение [13].

ЛИТЕРАТУРА

- [1] П. Азълов. Информатика с Паскал. Asio (АСИО), София, 1995.
- [2] П. Азълов. Сборник Информатика с Паскал в примерни тестове и задачи. Asio(АСИО), София, 1995.
- [3] О. Богданов, Д. Мустакеров. Език за програмиране на С. Техника, София, 2001.
- [4] М. Годорова. Програмиране на Паскал, София, 1993.
- [5] М. Бърнева, Т. Теодосиев. Паскал или С като първи език за програмиране. *Математика и математическо образование*, **31**, София, 234–238.
- [6] Е. Келеведжиев, З. Дженкова. Алгоритми, програми и задачи. Регалия 6, 2004.
- [7] Б. Йовчева, В. Спасова, Р. Христова, Д. Челебиева, Т. Теодосиев. Аспекти и проблеми на обучението по информатика в средното училище. Юбилейна научна сесия на Шуменския университет, 29.10.2001-1.11.2001 г.
- [8] Б. Йовчева, И. И. Апостолова, П. Д. Димитрова. Темата “Програмиране” в задължителния курс по информатика. Шуменски Университет “Еп. К. Преславски”. Научни трудове. Педагогически колеж, Добрич.
- [9] Б. Йовчева, А. Моллов. Обучение по компютърни технологии за деца от начална училищна възраст на базата на ОС Windows. Шуменски Университет “Еп. К. Преславски”. Научни трудове. Педагогически колеж, Добрич, том III, 114–118.
- [10] Б. Йовчева, А. И. Моллов. Тематика и цели на обучение по програмиране за 10-11 годишни деца (на базата на езика C++). Юбилейна научна конференция на ДИПКУ – Варна (сборник от научни трудове на конференцията).
- [11] Б. Йовчева, А. И. Моллов. Основни проблеми свързани с разработването на учебна програма по програмиране за 10-11-годишни деца (на базата на езика C++)”. Сб. Осигуряване и оценяване качеството на обучение. Доклади на Трета есенна научна конференция на Факултета по начална и предучилищна педагогика на СУ “Св. Климент Охридски”, “Веда Словена – ГЖ”, София, 2005.
- [12] www.infoman.musala.com.
- [13] Б. Йовчева, И. Иванова. Първи стъпки в програмирането на C/C++. Издателство КЛИМН, София, 2006
- [14] http://www.minedu.government.bg/opencls/opencls/left_menu/documents/educational_programs/

Бисерка Бончева Йовчева
Шуменски Университет “Еп. К. Преславски”
Факултет по математика и информатика
Катедра “Компютърна информатика”
гр. Шумен
e-mail: bissy_y@yahoo.com

**A “SPIRAL” APPROACH TO TEACHING ON PROGRAMMING FOR
10-11 YEAR OLD PUPILS (C++ BASED)**

Biserka B. Yovtcheva

The article presents an unique approach on teaching programming to the 10–11 year old pupils which is based on the idea, that the teaching of the children about writing algorithms must start immediately after it is thought about the first elements of the programming language, and to develop with time, while one and the same problem is viewed several times, but every time adding new knowledge. A manual has been published based on the described methods and on this basis experimental four year teaching of pupils has been organized.