

## ЕКСПЕРИМЕНТАЛНО ИЗСЛЕДВАНЕ ВЪРХУ ФОРМИРАНЕ НА ЗНАНИЯ И УМЕНИЯ ПО ПРОГРАМИРАНЕ

Тодорка Терзиева

Факултет по математика и информатика, Пловдивски университет, България  
dora@uni-plovdiv.bg

## EXPERIMENTAL STUDY ON FORMING OF KNOWLEDGE AND SKILLS IN PROGRAMMING

Todorka Terzieva

Faculty of Mathematics and Informatics, Plovdiv University, Bulgaria  
dora@uni-plovdiv.bg

**Abstract:** *The initial programming courses introduce the students to the technology of design, developing and implementation of a computer program. At the same time, we must encourage the development of skills necessary for the application of conceptual knowledge to create habits that can be applied for studying and developing the next disciplines in computer science. The programming is a specific type of human activity, the successful realization of which requires not only practical application of the knowledge and skills acquired during the learning, but it also requires a specific type of thinking.*

*The paper presents research results of experimental study on forming of knowledge, skills and competence in programming teaching to first year students majoring in Informatics from the Faculty of Mathematics and Informatics of Plovdiv University. Tools used to assess the results obtained by training include didactic tests, tasks and assignments that require writing a complete program. For the operationalization of the objectives of the study is used extended Bloom's taxonomy. The cognitive objectives of the extended taxonomy have universal nature and could be applied in programming teaching. The main questions, which must be answered, are related to whether the objectives are achieved, what is the efficiency of the learning work, how good is developed the educational environment and technology of teaching, etc. For diagnosis of the learning outcomes from the pedagogical experiment are developed and used three main criteria and twelve indicators.<sup>1</sup>*

**Key words:** *computer science education, programming, educational objectives*

Новото и бързо променящо се съдържание на обучението по информатика изисква разработване на методики, които да осигуряват не само възпроизвеждане на голям обем от знания, но преди всичко формиране и развитие у обучаемите на компетенции, позволяващи им активно да овладяват тези знания, изграждане на умения за самостоятелно придобиване на нови знания и критическото им осмисляне.

Според възприетото в европейската квалификационна рамка определение, резултатът от обучението се дефинира като показател за онова, което учащият знае,

---

<sup>1</sup> This work is partially supported by the NI11-FMI-004 project of Scientific Fund of the Plovdiv University.

разбира и може да направи при завършване на учебния процес (ЕКР 2008). Поради това се поставя акцент върху резултатите от обучението, които са конкретно описани в три категории – знания, умения и компетентности. В контекста на ЕКР компетентност означава *доказана способност за използване на знания, умения и личностни, социални и/или методологични дадености в работни или учебни ситуации и в професионално и лично развитие.*

### **Цел, задачи и етапи на изследването**

В работата се представят резултати от авторско изследване върху формиране на знания, умения и компетенции в обучението по програмиране на студенти от първи курс, специалност „Информатика” от ФМИ на ПУ „П. Хилендарски”. За получаване на обективна информация за достъпността на предложеното в изследването учебно съдържание и ефективността на разработената методика на обучение е организиран и проведен педагогически експеримент. Целта на изследването е повишаване ефективността на обучението по програмиране чрез разработване и теоретична обосновка на среда и технология на преподаване, осигуряваща развитие на алгоритмичното мислене (Гроздев, Терзиева 2011).

Понятието обучение е изключително сложно и не може да се разглежда едностранно. Процесът на обучение може да се представи чрез следния формален четири степенен модел (Андреев 1996): *цели на обучението* - формулират какво трябва да могат да правят обучаемите след самото обучение; *входно оценяване* - уточнява пригодността на обучаемите да постигнат целите на обучението; *процедури за обучение* - включват организиране на знанията в познавателни структури с цел облекчаване на ученето, изграждане на определени умения, съобразяване с особеностите на паметта и индивидуалните особености на учениците и др.; *оценяване* - идентифицира резултатите от обучението.

Моделът на обучение по програмиране е продължение и конкретизация на общата теория на обучението. В качеството на общодидактическа основа за модела на обучение приемаме системата на проблемно-базирано обучение, съгласно която дидактическият процес е последователност от проблемни ситуации (Давыдов 2005).

Традиционната структура на провеждане на педагогически експерименти включва три етапа (Бижков, Краевски 2007): предварителен (констатиращ) експеримент, процесуален (формиращ) и заключителен експеримент. Целта на констатиращия експеримент е да установи входното състояние на обекта на изследване. Формиращият експеримент е предложен след 4-5 седмици от процеса на експериментално обучение. Заключителният тест е проведен в края на изследвания период. Целта е да се проследи развитието на резултатите от прилагането на конструираната методика.

За диагностициране на резултатите от обучението са разработени съответни критерии и показатели (Terzieva 2011). Използвани са три основни критерия: *Критерий I* е свързан със знания и умения във връзка с анализиране и програмно решаване на проблеми, *Критерий II* – знания и умения, свързани с разбиране и изпълнение на алгоритъм, *Критерий III* – знания и умения, свързани с анализиране на алгоритми. Към всеки критерий има по четири показателя за по-точно и детайлно диагностициране на постигнатите резултати. Тъй като предлаганата методика обхваща съдържанието на курса по “Основи на компютърната информатика” и “Програмиране”, трудно е да се създадат критерии и показатели за предварително оценяване на обучаемите, които да бъдат използвани и при двата експеримента – констатиращия и контролния. Причина за това е фактът, че в последните два етапа от експеримента се наблюдават понятия и алгоритми, които няма как да бъдат известни на обучаемите предварително и няма как

степената на тяхното овладяване да се проследи още на етапа на констатиране. Затова повечето показатели, които са използвани за отчитане на резултатите, са с променена формулировка при констатиращия и заключителен експеримент.

За операционализация на целите се използва разширената таксономия на Блум (Anderson, Krathwohl 2001). Основните въпроси, на които се търси отговор, се отнасят до това, дали са постигнати поставените цели, каква е ефективността на учебната работа, доколко добра е разработената образователна среда и технология на преподаване и др.

### **Условия за провеждане на педагогическия експеримент**

Педагогическият експеримент е проведен в периода 2009/2011 г. със студенти от специалност "Информатика" от ФМИ на ПУ „Паисий Хилендарски“, по време на лабораторни занятия.

**Подборът на експериментална (ЕГ) и контролна група (КГ)** е на случаен принцип, от административно разделените на 3 групи студенти от специалност „Информатика“ – първи курс, като всяка група се състои от 23 студента. Студентите са изучавали в предишния семестър курс по „Основи на компютърната информатика“. Експериментът е проведен на три етапа.

Използваният инструментариум обхваща:

**Тестове** – основният за експеримента диагностичен инструмент. Тестове са провеждани на всеки от етапите. Поради спецификата на обучението по програмиране само резултати от тестовете не са достатъчни за точното оценяване на нивото на знания и умения – изисква се повече време за наблюдаване на практическите умения за работа с интегрираната среда за разработка, откриването и отстраняването на грешки по програмите и др. Затова в експеримента са използвани и други диагностични инструменти.

Специално създадените за целта **контролни упражнения** включват съставяне на модел на дадена предметна област, описание на етапите на разработване на алгоритъм, написване на програмен код по зададена спецификация, откриване и отстраняване на синтактични и семантични грешки в програма, задачи за намиране на оптимално решение относно използван алгоритъм или структура от данни и др.

**Задачи**, които изискват написването на цялостна програма. Подходящи са за оценяване на приложните и практически умения на студентите. Такива задачи са решавани веднага след тестовото изпитване на всички етапи. Предмет на оценяване са и задачите, решавани самостоятелно по време на лабораторни занятия.

При провеждането на експеримента са съблюдавани следните условия:

- постиженията на студентите се отчитат за два учебни часа (90 мин.), т.е. спазва се условието за еднаквост на времето;
- максималният брой точки при всеки тест е 24. Оценката се изчислява по формулата:  $K = 2 + 4 * n / N$ , където  $n$  е сумарен брой точки от всеки студент,  $4$  – брой оценки, различни от 2, ( $2 \leq \text{Оц} \leq 6$ ),  $N$  – максимален брой точки.
- оценяването на задачите от всеки тест се извършва съгласно следните критерии: за верен отговор – 2 точки, частично верен (непълнен) – 1 точка, грешен отговор – 0 точки.

На всеки от трите етапа от провеждане на експеримента студентите попълват тест от 12 въпроса и решават допълнително по една задача на C++. Съответните по номера въпроси се различават съществено и резултатите между различните етапи на измерване не могат да се сравняват. Резултатите, получени от студентите на всеки един от въпросите, са отчитани персонално, но за по-голяма прегледност данните са отразени в обобщен вид – общият брой точки се получава след събиране на отделните

айтъмни резултати от теста на всеки студент от двете групи (ЕГ и КГ) от специалност „Информатика”, както и пресметнатата по посочената по-горе формула оценка.

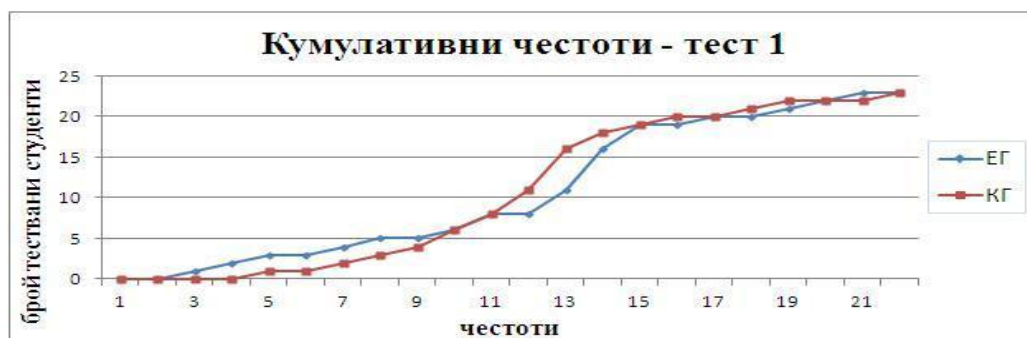
### Анализ на резултатите от проведените тестове

За сравняване на постиженията от различните етапи на измерване се използва дисперсионен анализ. Нулевите хипотези при този анализ гласят, че съответният фактор има нулев ефект, а всички проверяващи статистики имат F-разпределение. Наличието на статистически значимо взаимодействие между двата фактора – етап на тестване и принадлежност към контролна или експериментална група, при характерно разположение на средните стойности, се явява основният индикатор за положителен резултат на основния модел.

Класическият анализ за надеждност се извършва чрез пресмятане коефициента алфа на Кронбах с максимална стойност 1, както и на индивидуалните за всеки айтъм индекси на дискриминация с максимална стойност 1 (Бижков, Краевски 2007).

Проверка на хипотезата за равенство на дисперсиите на две нормално разпределени генерални съвкупности извършваме с F - критерий на Фишер. Двете групи са с нормално разпределена генерална съвкупност,  $F_{набл.}=0,628 < F_{крит.}=1,190$ , следователно останалите хипотези могат да се проверяват чрез параметрични методи. Изследването се извършва по три начина: графична проверка – по формата на разпределението; изчисляване на параметрите на разпределението ( $A_v$ ,  $SD$ ,  $A_c$ ,  $E_k$ ), точна проверка – чрез метода на Колмогоров-Смирнов.

Статистическият анализ на Тест 1 показва, че резултатите на студентите от двете групи са равностойни, т. е. имат приблизително равни възможности по „Основи на компютърната информатика” (фигура 1). Това ни даде основание да проведем експеримента с тези групи, като обучението на ЕГ се проведе по създадената методика и образователна среда, насочена към формиране и развитие на умения за разбиране и прилагане на алгоритъм, за моделиране и формализация на проблем, както и умения за анализиране на алгоритъм. Обучението в КГ се осъществи по традиционната методика.



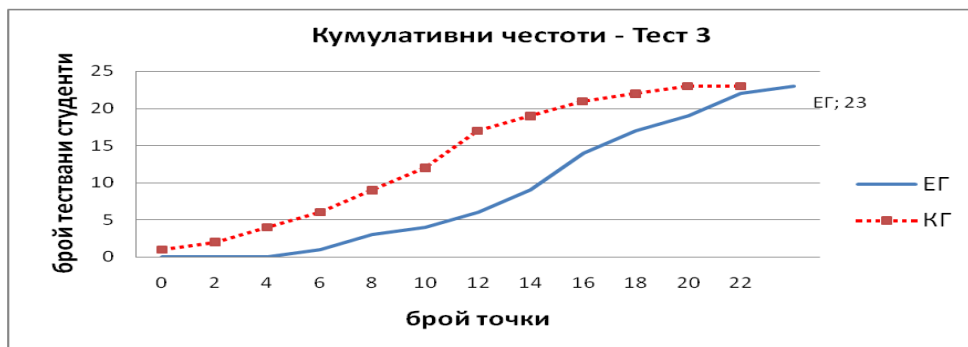
Фигура 1. Сравнителни резултати от тест 1

Анализът на резултатите е проведен след три месечно обучение по експерименталната методика. За целта е съставен критериален тест (тест 3).

**Нулева хипотеза  $H_0$ :** Разпределенията в генералните съвкупности ЕГ и КГ съвпадат, или

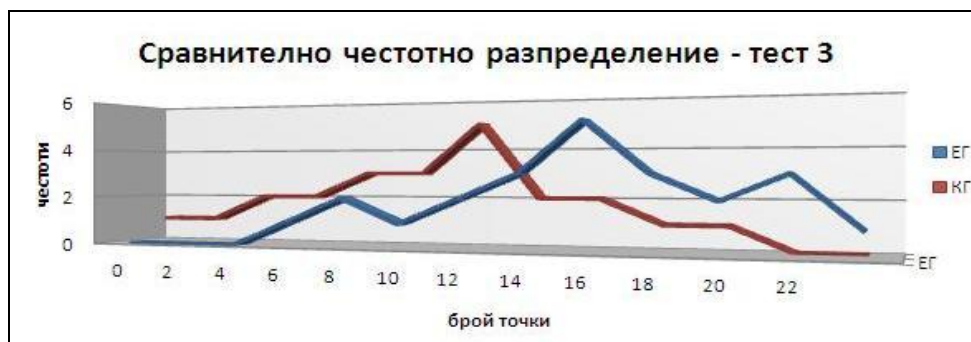
**Алтернативна хипотеза:** Разпределенията в генералните съвкупности ЕГ и КГ не съвпадат с избраната вероятност за грешка 0,05.

Прилагаме метода на Колмогоров-Смирнов,  $F_{набл.}=0,476$ ,  $F_{крит.}=0,350$ , следователно  $F_{крит.} < F_{набл.}$ , отхвърляме нулевата хипотеза и приемаме алтернативната.



Фигура 2. Сравнителни резултати от тест

От фигура 2 се вижда, че изобразените линии на кумулативните честоти съществено се различават, като тази на КГ се намира изцяло над линията на ЕГ, като в някои части разстоянието между тях е значително. За сравнение на резултатите от тест 3 извършваме и проверка на математическите очаквания, като прилагаме Т – статистика (*t-Test: Two-Sample Assuming Equal Variances*),  $t_{набл} = |-3,85597| > t_{крит} = 1,681952$  или  $|t_{набл}| > t_{крит}$ . Ако разгледаме хистограмите на двете групи ще забележим разликата в разпределението на оценките след проведеното обучение (фигура 3). Студентите от КГ показват значително по-ниски резултати в сравнение с експерименталната.



Фигура 3. Сравнително честотно разпределение от тест

От данните по показатели (фигура 4) следва, че най-съществена е разликата по отношение на резултатите по седми показател – умение за дефиниране на подходяща структура от данни, както и при умение за анализиране на проблем, умение за разделяне на проблема на подзадачи и умение за сравняване и анализиране на различни решения. В процеса на обучение се е развило в значителна степен умението за разбиране и изпълнение на алгоритъм. Най-значителна е разликата по отношение на умението за моделиране.



Фигура 4. Сравнителни резултати по показатели от тест

**Основен извод:** Статистическият анализ на педагогическия експеримент дава основание да се твърди, че разработената образователна среда чрез система от учебни задачи и технология на преподаване съдействат за управление на обучението по информатика и осъществяване на учебните цели. Предложената система от задачи и методика на преподаване създават условия за задълбочено и трайно усвояване на знанията, осмисляне на тяхното приложение при решаване на задачи от различни предметни области. Предложените дидактически подходи благоприятстват активното усвояване на знания, подпомагат формирането и развитието на умения за развитие на алгоритмичното мислене на студентите в базисните курсове на обучението по информатика.

## Литература

- Андреев, Марин.** Процесът на обучението. Дидактика. Университетско издателство „Св. Климент Охридски”, София, 1996, с. 421.
- Бижков, Г., Краевски, В.** Методология и методи на педагогическите изследвания. Университетско издателство „Св. Кл. Охридски”, София, 2007, с. 604.
- Гроздев, С., Гъров, К.** За системите от опорни задачи при подготовката за участие в олимпиади по информатика. Комбинирани обекти и алгоритми, Сборник доклади на 37 пролетна конференция на СМБ, Боровец 2-6 април 2008, с. 304-311.
- Гроздев, С., Терзиева, Т.** Изследование концепции алгоритмического мышления при обучении информатике, Международной научно-практической конференции „Информатизация образования – 2011”, Елец: ЕГУ им. Бунина, 14-15 июня 2011 г., Т 1. С. 112 – 119.
- Давыдов, Василий.** Лекции по общей психологии. Академия, 2005.
- Иванов, Иван.** Педагогическа диагностика, Университетско издателство „Еп. К. Преславски”, Шумен, 2006, с. 354.
- Рахнев, Асен.** Интензификация на обучението по програмиране чрез използване на информационни технологии, Хабилитационен труд за присъждане на научното звание “професор”, София, 2010.
- Anderson, L., Krathwohl, D.** A Taxonomy for learning, teaching, and assessing. New York: Longman, 2001.
- Grozdev, Sava.** For High Achievements in Mathematics: The Bulgarian Experience (Theory and Practice), Sofia: Association for the Development of Education, 2007.
- Terzieva, Todorka.** Some criteria and indicators for diagnosis of the forming of algorithmic thinking in Computer Science, Scientific Works, Plovdiv University, vol. 38, Book 3, Mathematics, 2011, p.
- Европейска квалификационна рамка за учене през целия живот (ЕКР), 2008, [http://ec.europa.eu/education/pub/pdf/general/eqf/broch\\_bg.pdf](http://ec.europa.eu/education/pub/pdf/general/eqf/broch_bg.pdf), последно посетен 12.10.2012 г.