

**ФОРМИРАНЕ НА КОМПЕТЕНЦИИ
ПРИ ОБУЧЕНИЕТО ПО МОДУЛ
„УВОД В ПРОГРАМИРАНЕТО“
ПО СПЕЦИАЛНОСТТА „ПРИЛОЖЕН ПРОГРАМИСТ“
НА НАЦИОНАЛА ПРОГРАМА
„ОБУЧЕНИЕ ЗА ИТ КАРИЕРА“ НА МОН**

Мухарем Моллов^{1,*}

¹ Факултет по математика и информатика,
Пловдивски университет „Паусий Хилендарски“, бул. „България“ № 236

* Автор за кореспонденция: muharem.mollov@uni-plovdiv.bg

Резюме. В статията се представят варианти на задачи за формиране на компетенции при обучението по модул „Увод в програмирането“ по специалността „Приложен програмист“ на Национала програма „Обучение за ИТ кариера“ (НПОИТК) на МОН. Описана е спецификата на курса, целите, използвания инструментариум и очакваните компетентности. Разгледан е раздел „Проверки“, включващ три теми, развиващи специфични компетенции, съгласно нивата по Европейската квалификационна рамка (ЕКР).

Ключови думи: обучение; ИТ кариера; приложен програмист; алгоритми, компетентностен подход, компетентност.

Увод

Европейският съюз следва и развива идеите, залегнали в Стратегия „Европа 2020“ за интелигентен, устойчив и приобщаващ растеж, както и Препоръката на Съвета на ЕС от 22 май 2018 година относно ключовите компетентности за учене през целия живот. Допълнително Европейската квалификационна рамка (ЕКР) за учене през целия живот и съгласуваната с нея Национална квалификационна рамка, поставят акцент върху резултатите от ученето (РУ), а не върху вложените средства. РУ се състоят от – знания, умения и компетентности. С термина „компетентност“ се означава доказана способност за използване на знания, умения и личностни,

социални и/или методологични дадености в работни или учебни ситуации и в професионално и личностно развитие. В контекста на ЕКР способностите се описват с оглед степента на поемане на отговорност и самостоятелност. Компетенциите представляват задължителните знания, умения и нагласи, които са зададени в образователните програми и са необходими за решаването на конкретни проблеми и задачи [1, 2]. Следователно, на понятията „компетенции“ и „компетентности“ при съвременното им тълкуване се задава различно значение.

Развитие на ключови и специфични компетенции у обучаемите по специалността „Приложен програмист“ в модул „Увод в програмирането“ от НПОИТК

Обучението по специалността се провежда на основата на Държавния образователен стандарт (ДОС) за придобиване на специалността „Приложен програмист“, в който са описани очакваните резултати (РУ) по общата, отрасловата и специфичната професионална подготовка с необходимите професионални компетентности, които гарантират на обучаемия възможността за упражняване на професията след завършването му. Развиват се, както ключови, така и специфични компетентности, като акцентът е върху три компетентностни области: математическа компетентност и в областта на точните науки, технологиите и инженерството; цифрова компетентност; личностна компетентност, социална компетентност и компетентност за придобиване на умения за учене. Формират се личностни качества: логическо и алгоритмично мислене; ясно и точно изразяване; дисциплиниране на мисълта и действието; отзивчивост и толерантност; активно отношение и способност за самообучение. Подобни качества и отношения започват да се развиват от най-ранна училищна възраст [3, 4]. Съществена характеристика на компетентността представлява трайността на знанията и уменията, получени след прилагането им в нови ситуации [5]. Според Гъров и екип в [6, 7, 8] рефлексията, самооценяването и взаимното оценяване между учащите е важен инструмент в изграждането на компетентностите у обучаемите. В разглеждания модул „Увод в програмирането“ самооценяването е подпомогнато от предоставената автоматизирана оценяваща система (Judge system) към сайта за обучение по програмата. Възможно е използването на други нейни алтернативи.

Модул „Увод в програмирането“

Първият модул, по който се обучават учениците от 10 клас по НПОИТК е „Увод в програмирането“. В края на модула обучаемите придобиват следните знания, умения и компетентности: познават основни

оператори, конструкции за проверка и цикъл, функции/методи в програмирането; Създават самостоятелно програмен код с ниска степен на сложност; съставят и прилагат основни алгоритми при решаването на задачи с ниска степен на сложност; използват логически и алгоритмичен подход при изпълнение на математически задания, свързани с програмирането; познават работни среди за разработка (IDE), компилация и интерпретация. Практическите измерители на тези компетентности са: писане на програмен код на начално ниво, работа със среда за разработка (IDE), използване на променливи и данни, оператори и изрази, работа с конзолата (четене на входни данни и печатане на резултати), използване на условни конструкции (if, if-else) и цикли (for, while, do-while).

Използвани подходи и методи са: учене чрез действие и преживяване; интерактивни методи и техники; активно общуване и осмисляне на собствения опит. За разработване на учебните материали, както и за методиките на преподаване е използван опит от Образцова математическа гимназия „Академик Кирил Попов“ – Пловдив [9, 10] и професионалисти от ИТ бранша, активни членове на БАИТ и БАСКОМ. Прилагането на спираловидния подход на преподаване чрез серия от интересни задачи с повишаваща се сложност спомага за развитието на компетентността и мотивацията на учениците. Самото преживяване доставя такава наслада, че обучаемия е готов да положи сериозни усилия, за да го изпита отново [11, 12]. Чрез решаването на практически задачи за програмиране учениците придобиват: разбиране за основни команди, за създаване на алгоритъм, развиват трайни умения и компетенции да пишат и преизползват програмен код в различни конструкции [13].

Учебното съдържание включва следните раздели: Въведение в програмирането, Пресмятания, Оператори, Изрази, Условни конструкции (Проверки), Повторения (цикли), Подпрограми (функции/методи). По-сложните раздели като „Проверки“ и „Повторения“ са разгледани в градация (спираловидно) чрез подходящо структурирани подтеми. Например за раздел „Проверки“ подтемите са: „Логически изрази и проверки“; „Серия от проверки“; „Сложни проверки“, а за раздел „Повторения“ са: „Повторения“; „Повторения с различни стъпки“; „Повторения от по-висока сложност“.

Наборът от дидактически материали и средства за постигане на целите на обучението в уроците за нови знания са: лекции (презентации), като теоретичните знания се онагледяват задължително с писане на код; ръководства със задачи и постъпковите им решение; оценяващи системи (Judge systems); среда за програмиране (dotnetfiddle.net и Visual Studio); хранилище на решенията (GitHub); видеа; професионални форуми, посветени на софтуерното инженерство – Stack Overflow; ръководство за програмиране на C#; анкети; мисловни карти. Съгласно ЕКР се цели:

придобиване и развиване на набор от познавателни и практически умения, необходими за намиране на решения на задачи, изискващи писане на несложен алгоритъм; компетентности (способност) за самостоятелно решаване на такива задачи. Самите задачи имат следната структура: условие, подсказка, примери с входни и изходни данни за проверка на правилността на резултатите, бонус.

Всяка използвана задача е съобразена с определена когнитивна област (по Блум) или модификацията от Андерсън и Кратуол, съответстващи пряко на ЕКР. Например, ако се разгледа частния случай за задачите от подтемите на раздел „Проверки“, те могат да се класифицират по начина, показан в Таблица 1.

Таблица 1. Класификация спрямо когнитивната област

Типове задачи	Подтема	По Блум	ЕКР
1. Въвеждане на основни знания и понятия	Логически изрази и проверки	Знания, разбиране	запомняне, разбиране
2. Затвърдяване на знанията	Серия от проверки	Приложение, анализ	Приложение
3. Приложение в нова ситуация	Сложни проверки	Синтез, оценка	Оценяване, създаване

Примерни задачи от тези области могат да бъдат следните:

- Знания – задачата формира компетенции, свързани с познаването и самостоятелното прилагане на условни конструкции (if; if-else) при писането на несложен разклонен алгоритъм.

Задача. 1. Проверка за отлична оценка

Условие: Да се напише конзолна програма, която чете оценка (десетично число), въведена от потребителя, и отпечатва “Excellent!”, ако оценката е 5.50 или по-висока.

Решението може да бъде представено чрез програмния код за конзолно приложение от Фигура 1.

- Разбиране – този тип задачи е насочен към развиването на компетенции, подпомагащи самостоятелното разбиране на логиката на програмен код, включващ условна конструкция if и if-else; възможност за обясняване и проследяване последователността на изпълнение на разклонен алгоритъм; различаване на кратка и пълна форма на операт if; самостоятелно съставяне на елементарен разклонен алгоритъм; писане на фрагмент от програмен код, включващ условна конструкция в IDE.

```
namespace Excellent_Result
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            var grade = double.Parse(Console.ReadLine());
            if (grade >= 5.50)
            {
                Console.WriteLine("Excellent!");
            }
        }
    }
}
```

Фигура 1.

Задача 2. Еднакви думи

Условие: Да се напише програма, която чете две думи, въведени от потребителя, и проверява дали са еднакви. Да не се прави разлика между главни и малки думи. Да се изведе “yes” или “no”.

Подсказка: Използвайте if-else конструкция. Преди сравняване на думите ги обърнете в долен регистър: `word = word.ToLower()`.

- Приложение – примерната задача формира компетенции, свързани с прилагането на придобитите знания и умения за решаване на нови задачи от практиката, както и откриването и коригирането на грешки в код, включващ условна команда.

Задача 3. Лица на фигури

Условие: Да се напише програма, в която потребителят въвежда вида и размерите на геометрична фигура и пресмята лицето ѝ. Фигурите са четири вида: квадрат (square), правоъгълник (rectangle), кръг (circle) и триъгълник (triangle). На първия ред на входа се чете вида на фигурата (square, rectangle, circle или triangle). Ако фигурата е квадрат, на следващия ред се чете едно число – дължина на страната му. Ако фигурата е правоъгълник, на следващите два реда се четат две числа – дължините на страните му. Ако фигурата е кръг, на следващия ред се чете едно число – радиусът на кръга. Ако фигурата е триъгълник, на следващите два реда се четат две числа – дължината на страната му и дължината на височината към нея. Резултатът да се закръгли до 3 цифри след десетичната точка.

- Анализ и синтез – анализът включва разделянето на задачата на нейните основни независими подслучаи и решаването на всеки от тях. Синтезът представлява обединяващата логика на алгоритъма, включително съставянето на самата програма. Включените

компетенции са насочени към самостоятелен анализ на структурата на програма чрез разбиване на съставните ѝ части; способност за разделяне на сложен алгоритъм на подалгоритми, включващи условна команда; дебъгване на код; графично представяне на алгоритъм с блок-схеми, правилно структуриране на програмата; разработване на план за съставяне на програма (алгоритъм); интегриране на наученото в алгоритъм за решаване на даден проблем от различни области.

Задача 4. Време плюс 15 минути

Условие: Да се напише програма, която чете час и минути от 24-часово денонощие, въведени от потребителя, и изчислява колко ще е часът след 15 минути. Резултатът да се отпечата във формат hh:mm. Часовете винаги са между 0 и 23, а минутите винаги са между 0 и 59. Часовете се изписват с една или две цифри. Минутите се изписват винаги с по две цифри, с водеща нула когато е необходимо.

Подсказка: Добавете 15 минути и направете няколко проверки. Ако минутите надвишат 59, увеличете часовете с 1 и намалете минутите със 60. По аналогичен начин разгледайте случая, когато часовете надвишат 23. При печатането на минутите проверете за водеща нула.

Бонус: Разгледайте кода на програмата на ваш съученик. Направете анализ на алгоритъма му. Ако има грешки в кода ги поправете.

Резултати от обучението

От проведеното обучение в модул УП, се вижда че учениците развиват своите специфични компетентности: познават основни оператори, конструкции за проверка; създават самостоятелно програмен код с ниска и дори средна степен на сложност; съставят и прилагат основни алгоритми при решаването на задачи с ниска степен на сложност; използват логически и алгоритмичен подход при изпълнение на математически задания, свързани с програмирането; работят със среда за програмиране Visual Studio и .Net Fiddle. Успех при решаването на задачите е над 90%. Изпитните резултати са отлични – успех 100%.

Заклучение

От направения анализ на резултатите се установява, че учениците надграждат стъпаловидно и спираловидно своите специфични компетентности (знания, умения, опит, рефлексия и самооценка за способността самостоятелно да се справят със задачите от курса), изграждат логическото, аналитично, алгоритмично, творческо и критично мислене.

Така осъщественото обучение води до трайни знания и умения у обучаемите. Учениците имат позитивно отношение към учебния процес и са високо мотивирани от формата на обучение.

Литература

- [1] Gancheva, Z., Application of competency-based approach to customs control training. *Dialogue*, 2, 14-47, 2019
- [2] Teodorescu, T., Competence versus Competency. What is the difference? *Performance Improvement*, 45 (10), pp 27-30, 2006
- [3] Stoitsov, G., G. Stoitsova, Increasing the motivation of primary school pupils through the use of ICT in the educational process. *International Journal of Research – GRANTHAALAYAH*, 7(2), 207-213, 2019
- [4] Стоицов, Г., Стоицова, Г. Виртуален инструмент за подпомагане на обучението в началното училище. *Образование и наука – за личностно и обществено развитие*. Националната научна конференция, стр. 75-84, 2017
- [5] Бойкова., К. Приложение на компетентностно ориентирания подход в обучението с използване на информационни и комуникационни технологии, Автореферат, <http://procedures.uni-plovdiv.bg/docs/procedure/1452/1812050166443263637.pdf>
- [6] Garov, K., E. Todorova, Reflection in studying electronic tables school, Proceedings of the Anniversary International Conference „Research and Education in Mathematics, Informatics and their Applications“, Plovdiv, Bulgaria, 2010, p. 427-434, ISBN 978-954-423-648-9.
- [7] Garov, K., E. Todorova, Reflection and comprehension in information technology education, *Scientific works Plovdiv university „Paisii Hilendarski“*, vol. 39, book 3 – Mathematics, 2012, p. 17-30, ISSN 0204-5249.
- [8] Гъров, К., Е. Тодорова, Самооценката в обучението по информационни технологии, *Образование и технологии*, бр. 4, 2013, стр. 398-408, ISBN 978-954-423-621-2
- [9] Старибратов, И., Тодорова, В., (2009). Професионалното обучение по програмиране в ОМГ „Академик Кирил Попов“ Пловдив, *Сборник доклади на Национална конференция „Образованието в информационното общество“*, Пловдив, стр. 88-92, 12-13 май 2009.
- [10] Старибратов, И., Димитрова, Ц., „За ранната подготовка на учениците по информатика“, *Сборник доклади на Национална конференция „Образованието в информационното общество“ Пловдив*, 12-13 май стр.93-102, 2009.

- [11] Csikszentmihalyi, M., *The Psychology of Optimal Experience*. New York: Harper & Row, 1990.
- [12] Ianakiev, Y., *Flow and Creativity: Strategies, Methods and Tendencies in the Study of the State of Flow In the Creative Process*. Positive Psychology. Plovdiv University Publishing House, pp. 147-160, 2016.
- [13] Рахнев А., А. Малинова, Н. Павлов, Параметризирано изпитване в средата DisPeL. Международна конференция „From DeLC to VelSpace”, 26-28 март 2014, Пловдив, Proceedings, 263-272

**DEVELOPMENT OF COMPETENCIES DURING
THE EDUCATION IN THE MODULE
“INTRODUCTION IN PROGRAMMING” AS A PART
OF THE DEGREE “APPLICATION PROGRAMMER”
OF THE NATIONAL PROGRAM “EDUCATION FOR IT
CAREER” OF THE MINISTRY OF EDUCATION**

Muharem Mollov^{1,*}

¹ *Faculty of Mathematics and Informatics, University of Plovdiv Paisii Hilendarski, Bulgaria Blvd. 236, Bulgaria, muharem.mollov@uni-plovdiv.bg*

** Corresponding author: muharem.mollov@uni-plovdiv.bg*

Abstract. The article presents different variants of tasks used for the development of competencies during the education in module “Introduction in programming” as a part of the degree “Application Programmer” of the national program “Education for IT career” of the Ministry of Education. The specificity and the goals of the course, as well as the used tools and the expected competencies are described. A review of the chapter “Verification” which includes topics for the development of specific competencies according to the levels of the European Qualifications Framework (EQF) was made.