

ПРЕДИЗВИКАТЕЛСТВОТО – НАЦИОНАЛНА ПРОГРАМА „ОБУЧЕНИЕ ЗА ИТ КАРИЕРА“

Мухарем Моллов, Генчо Стоицов

Пловдивски университет „Паисий Хилендарски“ – Пловдив

Резюме. В статията се коментира обучението по програмиране за ученици от средната образователна степен, които участват в придобиването на професионална квалификация по професия „Приложен програмист“. Анализирани са европейски и международни препоръки за разработване на учебни програми в областта на компютърните науки и е направено сравнение с учебния план и учебните програми по модули за професия „Приложен програмист“. Коментират се целите на обучението, методите и инструментите на преподаване, подготовката на учители.

Ключови думи: образование; ИТ кариера; приложен програмист; програмист

1. Въведение

Образованието, както в национален, така и в световен мащаб е силно повлияно от високите темпове на развитие на икономиката, информационните и комуникационните технологии (ИКТ). То е изправено пред новите предизвикателства и възможности да бъде адекватно на съвременните технологични изисквания и обществени очаквания. Особена важност придобива въпросът за възможността ефективно и качествено да се подготвят млади специалисти в областта на високите технологии, в частност в областта на приложното програмиране. Налага се разработване и интегриране на гъвкави педагогически и методически подходи на преподаване на съвременни технологии, които да отговарят на потребностите както на обучаемите, така и на индустрията в обобщение. Според (Hristov & Krushkov, 2015), (Hristov & Krushkov, 2015a) процесът на обновяване и актуализиране на образователното съдържание, усъвършенстването на методологията, методите и средствата на преподаване чрез внедряване на съвременни технологии за създаване на софтуер в учебните програми протича бавно в сравнение с революционното развитие на технологията. Според цитираните източници методологията за подготовка на специалисти с висше образование за софтуерното производство не е достатъчно ефективна. Обучението по информатика е на високо ниво в университетите, но подготовката на студентите в направление „Софтуерно инженерство“ изостава. Направени са препоръки към съдържанието на учебните планове,

броя на застъпените теми, липсата на подходи и модели за определяне на междудисциплинарни връзки (Hristov, 2016).

Съвсем логично е проблемът в средното образование да стои още по комплицирано, като се отчитат спецификите на обучаваните по отношение на тяхното ниво на знания и компетенции, възрастови, психологически и други фактори, влияещи на формирането им като млади специалисти със средно образование. Като пояснение трябва да се добави, че висшето и средното образование трябва да са в синхрон: учениците, продължаващи обучението си в университет, да могат да използват натрупаните знания и умения от средното образование за последващо усъвършенстване, както и подготовката на педагогически специалисти за средното образование в университета трябва да гарантира качеството на обучение на ученици, така че спиралата от знания и компетенции да е развиваща се.

В отговор на тези предизвикателства, на базата на Стратегията за ефективно прилагане на информационни и комуникационни технологии в образованието и науката на Република България за периода 2014 – 2020 г.¹⁾ и Концепцията за насърчаване обучението на софтуерни специалисти от 2015 г. на МОН²⁾, е създадена Национална програма „Обучение за ИТ кариера“³⁾, съобразена с поставените цели в Концепцията и подписаното споразумение с представителите на ИТ сектора в България, чиято цел е разширяване базата от ученици, които са придобили професионална квалификация по професия от професионално направление „Компютърни науки“ и/или продължават образованието си във висши училища в професионално направление „Информатика и компютърни науки“.

За успешна реализация на практическото обучение на учениците е важно да се синхронизира целият процес с европейските и световните тенденции в областта на компютърните науки и в частност със софтуерното инженерство.

2. Европейски и международни тенденции

Една от водещите организации в областта на разработването на препоръки за учебни програми, свързани с компютърните науки, е ACM (Association for Computing Machinery). Тя се стреми да поддържа актуално нивото на препоръките, съобразно съвременното технологично развитие. ACM класифицира компютърните науки в няколко направления⁸⁾: компютърно инженерство, информатика, информационни системи, информационни технологии, софтуерно инженерство. В редица документи дава указания към различните направления за квалификации, изготвя компетентностен модел по информационни технологии, създава програми и стандарти за обучение в начална, прогимназиална и гимназиална степен на образование.

В доклад на тема „Informatics Education in Europe: Are We All In the Same Boat?“⁴⁾ от Комитета по европейско образование в компютърната сфера (The

Committee on European Computing Education, CECE), създаден от организациите ACM Europe и Informatics Europe, е направено проучване за актуалното състояние на образованието по информатика, цифрова грамотност и обучението на учители в Европа и Израел (общо 55 административни области). В него, след разясняване на разликата между двете концепции – информатика и цифрова грамотност, е предоставена богата информация за състоянието на образованието по тези направления, както и за обучението на учители, които да ги преподават.

Според доклада информатиката е отделна научна дисциплина, характеризира се със собствени фундаментални концепции, методи, знания, практики и отворени въпроси в области като: данни, информация и представяне; алгоритми и програмиране; шаблони и параметризиране; абстракция и концептуално моделиране; устройства, мрежа и интернет; изчисление и комуникация; дизайн и взаимодействие; сигурност, неприкосновеност на личния живот и етика; обществено въздействие.

Цифровата (дигитална) грамотност покрива свободното използване на стандартни софтуерни инструменти и интернет, основни практически умения, които всеки гражданин трябва да познава, за да живее в XXI век.

Като най-голямо предизвикателство е посочено въвеждане на информатиката като предмет на всички образователни нива: начално, средно и висше образование. Предложената стратегия включва специализация и интеграция. Изпълнението на тази стратегия води до трудни задачи. Във висшето образование всички останали учебни програми трябва да разглеждат и усвояват съответните аспекти на информатиката. В средното и основното образование информатиката трябва да се развива както като индивидуален и задължителен предмет, така и да се интегрира в останалите предмети. Тези промени представляват голямо предизвикателство за всички образователни системи. Препоръките са за специализирано изучаване във всички степени на образованието с интегриране в други учебни дисциплини на ниво развитие на компетенциите на обучаемите. За изпълнението на подобни препоръки е необходимо разработване на прецизни учебни програми и ефективни учебни материали, които постепенно да приспособяват когнитивното развитие на учениците и да развиват подходящи знания и умения. Допълнително задължение е подготовката на учители за преподаване на всички нива, оказване на подкрепа на същите чрез осигуряване на адекватна квалификация и материална база за ефективно изпълняване на тяхната работа.

Цифровата грамотност, според доклада на CECE, трябва да се преподава от ранните етапи на обучението, където обучаемите се запознават с основните аспекти до момента. Всички граждани трябва да бъдат дигитално грамотни, тъй като днес много (ако не всички) социални и професионални услуги, които използват, са онлайн дейности.

От изложението на доклада става ясно, че трите основни направления, на които трябва да се обърне внимание, са учебните програми, методите и инструментите на преподаване и подготовката на учители.

3. Националната програма „Обучение за ИТ кариера“

Анализът на АСМ препоръките за различни учебни направления⁸⁾ показва, че учебната програма на националната инициатива „Обучение за ИТ кариера“ най-много се доближава до тази на софтуерното инженерство. Очакванията към софтуерното инженерство са свързани с осигуряване на възможност за подготовка на специалисти, които да познават добре всички аспекти на разработването и поддържането на надежден и ефективен софтуер за цялата област на компютърните приложения. Световният и националният опит в тази област дават много полезни практики, които трябва да бъдат анализирани, осъвременени и успешно вградени на местата, където ще бъдат методически ефективни в процеса на обучение.

Принос със своята практика в обучението дава първата Частна професионална гимназия по дигитални науки в България – „СофтУни – Светлина“, която работи по програма „Приложен програмист“, следвайки отблизо съвременното разбиране за понятието софтуерно инженерство.

Учебният план предвижда тригодишно обучение по професията „Приложен програмист“. Разработени са учебни програми^{5),6),7)}, покриващи изучаваните модули от таблица 1.

Таблица 1

Модул	Присъствени часове	Часове за самоподготовка	Часове за изпит	Часове общ хорариум
I година – X клас				
Увод в програмирането	50	18	4	72
Програмиране	66	18	6	90
Увод в обектно ориентираното програмиране (ООП)	32	36	4	72
Увод в алгоритмите и структурите от данни	66	18	6	90
II година – XI клас				
ООП	94	20	6	120
База данни	64	20	6	90

Разработка на софтуер	94	44	6	144
Въведение в операционни системи и вградени системи	46	22	4	72
III година – XII клас				
Алгоритми и структура от данни	67	72	5	144
Функционално програмиране	28	28	2	58
Интернет програмиране	75	90	5	170
Софтуерно инженерство	67	72	5	144

Какъв профил на обучавания се стремим да изградим?

За по-пълнен и точен отговор на този въпрос е необходимо детайлно запознаване със същността на обучението по професията „Приложен програмист“ във всичките ѝ аспекти:

- педагогически, професионални, технологически, психологически (общуване и личностни), касаещи изграждане на качества на личността, спомагащи за успешното ѝ професионално израстване, т.нар. „меки умения“ (soft skills);
- проучване на световния и националния опит в това обучение – училищно, университетско, в професионални академии (национални и световни лидери в това направление).

Според Наков⁹⁾ желаният млад специалист по професията „Приложен програмист“ трябва да придобие основни знания и умения в следните направления:

- писане на код (coding skills) – 20%. Включва: работа с променливи, проверки, цикли; ползване на функции (методи), обекти; работа с данни: масиви, списъци, речници и други структури.
- алгоритмично (инженерно, математическо) мислене – 30%. Способност да се анализират задачи и да съставят решения чрез разбиране на проблема на стъпки (алгоритъм). Изгражда се чрез решаване на задачи по програмиране (1000+). Отнема половин-една година. Необходими курсове: *Programming Basics, Technology Fundamentals, Data Structures, Algorithms*.
- фундаментални знания за професията – 25%. Включва: обектно ориентирано програмиране, функционално програмиране (ФП), асинхронно програмиране и паралелност, бази данни, SQL, документни бази (напр. MySQL, MongoDB), интернет технологии (HTTP, JS front-end, DOM, AJAX, REST, ...), софтуерно инженерство (сурс контрол, процеси, архитектури).
- езици за програмиране и софтуерни технологии – 25%. Езиците за програмиране и технологиите винаги са в комбинация (technology stack). Примерен минимум за Junior. NET Developer:

C# + .NET Core + VisualStudio + бази данни + SQL Server + SQL + Entity Framework + ASP.NET MVC + HTML + CSS + JavaScript + AJAX + REST + XML + ООП + ФП + алгоритмично мислене + Git + софтуерно инженерство + английски език + личностни качества

Допълнения към този профил прави Йовчева (Yovcheva, 2018) при проучване на ролята на алгоритмичното програмиране за подготовка на висококвалифицирани специалисти за ИТ сектора. Направените заключения за качества, които развиват обучаемите в това направление са следните.

- Алгоритмичното (състезателно) програмиране, изисква максимално съсредоточаване при решаване на дадена задача. То подготвя обучаемия за моментална реакция при кризисни ситуации и изгражда умения за бързо и ефективно решаване на възникналите проблеми.
- Обучението в състезателно програмиране създава навици за писане предимно на високоефективен код, като се отчита ефективност и във време, и в памет.
- Учениците, обучавани в това направление, привикват на строга дисциплинираност и самоконтрол, което ги прави едни от най-надеждните членове на даден ИТ колектив.
- Умението да се откриват и отстраняват логически грешки в кода, е едно от най-впечатляващите следствия от подготовката на състезателните програмисти.
- Самообучението е важна част от подготовката на състезателите и това допринася за развитието на уменията им за проучване и изучаване на конкретна проблематика.
- Прецизната и задълбочена подготовка в областта на програмирането води и до отлична представа за общата организация на работа на съвременните компютърни системи като съвкупност от операционна система, различни видове програмни интерфейси и потребителския софтуер.

4. Методика и средства за обучение

Използваната методика е съобразена както с видовете дейности, които се извършват по време на обучението, така също и с конкретния курс, което понякога изисква различна организация на обучението. Използват се подходящи технологии и средства за обучение като:

– среди за програмиране (dotnetfiddle.net, VisualStudio). Първата е удобна за писане на несложен код, който може да се тества на всяка машина, имаща достъп до интернет (нашите ученици минаха целия първи учебен курс „Увод в програмирането“ в тази среда). Visual Studio осигурява среда за разработка на цялостни решения на професионално ниво, като предлага Use refactoring

and Intelli Sense, Completion lists, Automatic code generation, Generate event handlers, Debugcode.

- хранилища за данни (Git, GitHub). Предлагат възможност за неограничено съхранение на цели проекти, пакейки кода на версиите с възможност за разклоняване и достъп до всеки момент в историята на развитието на проекта. Позволява достъп до огромни масиви от споделени ресурси.
- електронни системи за управление на обучението с различни видове ресурси (G Suite for Education Google Classroom; Moodle; порталът за е-обучение по специалност „Приложен програмист“ на сайта на МОН, <https://it-kariera.mon.bg/e-learning>). Предлага се цялостно онлайн обучение – обучителни материали под формата на презентации за новите теми; документи със задачи; ръководства за решаване на задачи; линкове към материали, видеа, онлайн книги по темата.

Използването на такава виртуална среда е съвсем естествен етап в приложението на ИКТ в образованието и още повече в обучението по компютърни науки, където може максимално да се използват предимствата на този подход. Това е пътят за кариерно развитие на младите хора, за придобиването на необходимите знания, умения, компетенции и норми на поведение (Angelova & Staribratov, 2010).

Представяват интерес изследвания в областта на използването на VLE (Virtual Learning Environment) и LMS (Learning Management System) на (Cavus, Uzunboylu & Ibrahim, 2006), които препоръчат колаборативен подход на работа за повишаване на ефективността от обучението.

– Автоматизирани оценяващи системи (Judge systems) на МОН и „СофтУни“. Системата на МОН дава обратна информация за всички неуспешни тестове, което улеснява обучаемите, т.нар. сива кутия (graybox). Подходяща е за начинаещи. Системата на „СофтУни“ не връща информация за това каква грешка е настъпила, а само, че някои тестове не са минали успешно, третира скритите тестове като черна кутия (blackbox). Доближава се повече до професионалното програмиране. Препоръчва се съвместно използване и на двете. Друг аспект на използването на оценяващите системи е изпитването. Осъществява се в системата на МОН. Този формат на изпитване позволява реално оценяване на знанията и уменията на обучаемия. Това позволява на учителя да се съсредоточи върху качеството на упражненията и процеса на преподаване.

Възможно е използването и на други уеб базирани или локално инсталирани подобни системи. Основни приложения на Judge системите са: онлайн платформи за състезателно програмиране за натрупване и споделяне на опит, подобно на това по време на състезание; образование; онлайн компилатори; набиране на персонал; в помощ на други състезания и обучителни курсове и други. По-голямата част от тези системи са отворени за потребители по

целия свят, които желаят да подобрят своето алгоритмично мислене и умения за програмиране.

– По време на изпита обучаемите решават изпитни задачи, подобни на разгледаните във всеки модул. Тестват решенията в оценяващата система, при което получават веднага обратна информация за резултатите от проверката, а това е допълнителен мотивационен подход. Друг положителен елемент по време на изпит е възможността за подаване на множество решения от обучаемите, което позволява нанасянето на подобрения в реализирания код и развитие на специализирани творчески умения.

– Използване на независими източници (сайтове – w3schools.com, mysqltutorial.org; видео лекции; презентации; форуми – StackOverflow; обратна връзка с обучаемите – sli.do).

Като използвани подходи и методи може да се посочат:

– решаване на задачи – ключов елемент в обучението както по време на часовете за упражнение, така и в тези за нови знания. Принципът е учене чрез правене, при който минимум 50% от заниманията са предвидени за писане на код. Всяко ново знание се подкрепя с примери на практика под формата на лабораторни упражнения. Структурата на всяка задача включва задължителни и опционални елементи:

- заглавие на задачата;
- текстово описание на заданието;
- примери с базови тестови случаи, включващи входни данни и изходни данни под формата на текст. Ако условието е по-сложно, се включват и обяснения за решаването на задачата при конкретен тестов случай. Представява своеобразно кратко ръководство за решаване на тестовия случай и е особено важно за начинаещите;
- допълнителни указания, ако са необходими;
- подсказки, ако са необходими, особено полезни за случаите, в които задачата изисква по-задълбочен анализ или е творческа съгласно таксономията на Блум.

– проектно базирано обучение – включва работа в екип при разработването на софтуер, симулирайки реална ситуация от ИТ сектора (Ghezzi & Mandrioli, 2006; Cavus, Uzunboyly & Ibrahim, 2006). В подобни ситуации учениците не само създават, но и менажират самия процес. Обръща се внимание на така наречените меки умения. Подобрява се екипната работа.

– игровизация, игров подход – доказан високоефективен метод, повишаващ мотивацията на обучаемите. Играта е интересен казус. Усещането у обучаемите е, че правят нещо значимо, към което преди са проявявали респект и са го смятали за нещо непостижимо. При обучението се използват две разновидности: решаване на конкретна задача чрез писане на код, режисирано от учителя под формата на игра, например с използването на

състезателен елемент; разработка и програмиране на елементарна игрова логика под формата на задача. Например подходящи за обучаемите в първата година са игри като „Познай числото“ (Teodosiev & Teodosieva, 2018), където се прилага алгоритъм за двоично търсене в подредена съвкупност; „Крави и бикове“ – логическа игра, генерираща четирицифрено число (без повтарящи се цифри), а играчът да прави на всяка стъпка предположение за генерираното число, което трябва да познае с минимален брой опити, и др.

– дебъгване, тестване на код и знания – за целите се използват автоматизирани системи. Дебъгването е дейност при разработването на софтуер, която анализира и премахва причините за грешки, а тестването може да се разглежда в два основни аспекта: тестване на качеството на софтуера и проверка на знанията и уменията на обучаемите. Тестването на кода е един от методите за развитие на алгоритмично мислене. То е основна дейност на програмиста. Обучаемите трябва да усвоят неговите основни принципи, етапи, цели, видове, методи за тестване. Да се запознаят с възможностите на автоматизираното тестване, както и на практика да използват такива системи. Например Selenium е javascript базиран софтуер за тестване на уеб приложения, работи с тестове, написани на C#, PHP, Java, Ruby, Python, Perl, Javascript.

5. Подготовка на учители

Програмата включва обучение и на преподаватели. Полезни са курсовете на Софтуерния университет, провеждани безплатно за учители във всичките направления. Според световните изследвания, правени за ефективно обучение по компютърни науки, се акцентира на високата степен на компетентност на учителите, с която не трябва да се прави компромис. Учителите, които желаят да преподават по това направление, трябва да са в синхрон с твърде динамично изменящите се нови технологии, което усложнява процеса на обучение. Преподаването по тази програма по отношение на учебно съдържание се доближава до тези в университетите. Авторите на (Hazar, Gal-Ezer & Blum, 2008) препоръчват обучението на учители за подобни цели да се извършва от специалисти с минимум бакалавърска степен по компютърни науки и актуални познания в технологичната област.

6. Заключение

Обучението по професия „Приложен програмист“ цели популяризиране и масовост сред учениците от средните училища, които не се обучават по тази професия. Срокът и формата на обучение предвиждат извънредни занимания в извънучебно време в нелека научно-приложна област, където се изисква до-

бра подготовка по математика и други предмети. Това усложнява изпълнението на целта за масовост. Въпреки това обучаваната от нас група е съставена от 10 – 13 обучаеми при максимален избор от 26 ученици, което е един необичайно висок процент (приблизително 50%). В някои области учениците трудно могат да се придвижват до регионалните центрове, в които да се обучават в по-големи групи, но това не е проблем, защото формата на обучение изисква нескъпо оборудване, интернет свързаност и възможност за обучение в същите училища при наличие на специалисти. За мотивация на децата важни фактори са: престижността на професията; желанието за социален статус на специалистите в тази област; увлекателността към новите технологии. Целта е твърде амбициозна и за постигането ѝ трябва да се подходи спираловидно: на всеки етап да се подобрява преподаването на база статистика от постигнатото и натрупаният опит от преподавателите; да се отчита развитието на технологиите и на всеки 2 – 3 итерации да се осъвременява учебното съдържание.

БЕЛЕЖКИ

1. Ministry of Education and Science, Strategy papers, <http://www.strategy.bg/StrategicDocuments/View.aspx?Id=904>, (last visited 01.09.2020)
2. Ministry of Education and Science, Promoting training of software specialists, <http://www.strategy.bg/FileHandler.ashx?fileId=6315>, (last visited 01.09.2020)
3. Ministry of Education and Science, National IT Career Training Program, https://www.mon.bg/upload/11730/NacPr11_IT_kariera.pdf, (last visited 01.09.2020)
4. Informatics Europe, Informatics Education in Europe: Are We All In The Same Boat?, <https://www.informatics-europe.org/component/phocadownload/category/10-reports.html?download=60:cece-report>, (last visited 01.09.2020)
5. Ministry of Education and Science, Schedule of training first year, https://it-kariera.mon.bg/e-learning/pluginfile.php/92/mod_forum/attachment/29/Grafik-1%20god_2019_2020.pdf?forcedownload=1, (last visited 01.09.2020)
6. Ministry of Education and Science, Schedule of training second year, https://it-kariera.mon.bg/e-learning/pluginfile.php/92/mod_forum/attachment/59/Grafik-2%20god_2019_2020.pdf?forcedownload=1, (last visited 01.09.2020)
7. Ministry of Education and Science, Schedule of training third year, https://it-kariera.mon.bg/e-learning/pluginfile.php/92/mod_forum/attachment/60/Grafik-3%20god_2019_2020_new.pdf?forcedownload=1, (last visited 01.09.2020)
8. ACM recommendations, <https://www.acm.org/education/curricula-recommendations>, (last visited 0.10.2020)
9. Nakov, Sv. How to become a programmer?, <https://www.slideshare.net/nakov/ss-128219305>, (last visited 0.10.2020)

ЛИТЕРАТУРА

- Христов, Хр. & Крушков, Хр. (2015). Състоянието на обучението по информатика и софтуерни технологии. *Образование и технологии*, 6, 388 – 394.
- Христов, Хр. & Крушков, Хр. (2015a). Обучението по информатика и софтуерни технологии: развитие и перспективи. *Образование и технологии*, 6, 394 – 401.
- Христов, Хр. (2016). Методика на преподаване на съвременни технологии за създаване на софтуер. *Дисертация, Автореферат*. Пловдив: ПУ „Паисий Хилендарски“. Hristov, H. (2016).
- Йовчева, Б. (2018). Ролята на алгоритмичното програмиране за подготовката на висококвалифицирани специалисти за ИТ сектора. *Образованието и изследванията в информационното общество, XI Национална конференция*, 162 – 169.
- Теодосиев, Т. & Теодосиева, Г. (2018). Преподаване на програмиране чрез разработка на игри. *MATTEX 2018. Conference proceeding*, стр. 265 – 272.

REFERENCES

- Hristov, H. & Krushkov, H. (2015). Education in Informatics and Software engineering: State of the art. *Education and technology*, 6, 388 – 394.
- Hristov, H. & Krushkov, H. (2015a). Education in Informatics and Software engineering: Development and future directions. *Education and technology*, 6, 394 – 401.
- Methods of teaching advanced software creation technologies. *PhD thesis*. Plovdiv: Plovdiv University Press “Paisii Hilendarski”.]
- Angelova, E. & Staribratov, I. (2010). On a Virtual Learning Environment. *REMIA 2010. Anniversary International Conference*, pp. 403 – 410.
- Cavus, N., Uzunboyly, H. & Ibrahim D. (2006). The Effectiveness of Using Learning Management Systems and Collaborative Toolin Web-Based Teaching of Programming Languages. *International Symposium and Education on Electrical, Electronic, and Computer Engineering (ISEECE)*.
- Ghezzi, C. & Mandrioli, D. (2006). The challenges of Software Engineering Education. *Software Engineering Education in the Modern Age, International Conference on Software Engineering*, pp. 115 – 127.
- Yovcheva, B. (2018). The role of algorithmic programming in the training of highly skilled IT professionals. *Education and researches in the information society, XI National Conference*, pp. 162 – 169.
- Teodosiev, T. & Teodosieva, G. (2018). Teaching programming through games development. *MATTEX 2018. Conference proceeding*, pp. 265 – 272.

Hazar, O., Gal-Ezer, J., & Blum, L. (2008). A Model for High School Computer Science Education: The Four Key Elements that Make It!. *SIGCSE '08, 39th SIGCSE technical symposium on Computer science education*.

THE CHALLENGE – NATIONAL PROGRAM “TRAINING IN IT CAREER”

Abstract. The article comments the education of secondary school students during their participation in a computer programming course for obtaining the professional qualification “Application programmer”. European and international recommendations for the elaboration of learning programs for students in the field of Computer Science are analyzed and their comparison with the curriculum and learning programs for the occupation of “Application programmer” is realized. The aims of the presented teaching, the methods and the instruments of the process during the course and the preparation of teachers in the field of Computer sciences are described.

Keywords: education; IT career; application programmer; computer science

✉ **Mr. Muharem Mollov, PhD student**
Dr. Gencho Stoitsov, Assoc. Prof.

Faculty of Mathematics and Informatics
University of Plovdiv “Paisii Hilendarski”

24, TzarAsen St.

4000 Plovdiv, Bulgaria

E-mail: muharem.mollov@uni-plovdiv.bg
stoitsov@uni-plovdiv.bg