



DOI: 10.54664/ZCTU5237

ПРИЛОЖЕНИЕ НА OBJECTS-EARLY ПОДХОД В УЧИЛИЩНИЯ КУРС ПО ИНФОРМАТИКА

Габриела Чотова

APPLICATION OF THE OBJECTS-EARLY APPROACH TO THE SCHOOL COURSE ON COMPUTER SCIENCE

Gabriela Chotova

Abstract: *Raising the level of digital education is one of the main challenges facing the education system now and in the future. The use of modern technical means in the learning process is an important prerequisite for achieving this ambitious goal. However, this must go hand in hand with the introduction of new scientific methods and teaching approaches. This article presents different approaches which the author has applied in her practice with high school students. It proves the benefit of their implementation in the educational process.*

Keywords: *innovations, approach, computer science, methodology, programming, didactic tools.*

ВЪВЕДЕНИЕ

Развитието на образованието и промяната на учебните планове през последните няколко години изправи учителите пред много предизвикателства. Драстични са промените в предметите, свързани с дигиталната компетентност.

Едно от основните предизвикателства е по предмета „Информатика“. Акцент в обучението тук вече е използването на визуално програмиране за създаване на несложни приложни софтуерни продукти със съвременен интерфейс на управление [3]. Това налага коренна промяна и в подхода на преподаване. Не е достатъчна само замяната на използвания процедурен език за програмиране с чист обектноориентиран. Нужна е мащабна промяна в цялостната методика по отношение на учебния материал, методите на преподаване и оценяване, системата от задачи в учебния курс.

Водени от желанието да бъдем полезни на учениците и на учителите в тази статия представяме своя подход за преподаване на предмета „Информатика“, вариант на класическия модел на въведение в компютърните научни чрез програмиране с ранно въвеждане на обектноориентираните концепции.

ИЗЛОЖЕНИЕ

Въпреки стриктното изпълнение на заложените в наличните учебници изисквания и прилагането на всички методически препоръки очакваните резултати от обучението през последните години не бяха достигнати в нужната степен. Усвояването на основните концепции в програмира-

нето не беше достатъчно задълбочено. Много понятия се оказаха неразбрани в тяхната пълнота. Това ни мотивира да предложим промени в педагогическия подход и да споделим опита си от прилагането му. Акцентира се на:

1. Създаване на подходящо учебно съдържание.
2. Структуриране на учебното съдържание съгласно училищната методология.
3. Осигуряване на междупредметни връзки с други учебни предмети.
4. Създаване на система от учебни задачи за основните теми, свързани както с основни алгоритми, така и с решаване на проблеми чрез търсене.

Подходът, който предлагаме, е вариант на objects-first стратегията [8]. Обектите се въвеждат рано (objects-early), но без да се подценяват процедурните конструкции. Практиката ни показва, че в обучението по информатика най-удачно е да се комбинират идеи от няколко подхода. Нашият вариант залага на допълнителна мотивация на учениците чрез:

- използване на съвременна среда за разработка (актуална версия на Visual Studio и Eclipse IDE) и изграждане на приложения с графичен интерфейс (WPF, Windows Forms и Java JFrame);
- управление на физически системи [3].

Преди да преминем към по-задълбочено разглеждане на подхода, който предлагаме, накратко ще дадем информация за ситуацията в училищния курс по информатика към момента.

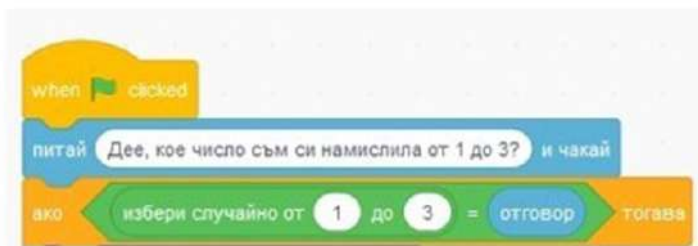
Стратегията, по която е организирано учебното съдържание, е повлияна силно от programming-first подхода към въведение в информатиката и компютърните науки. В основата на този подход е идеята, че програмирането е най-подходящото средство за навлизане в сложната материя и трябва да се включи на ранен етап от обучението. И това може да се случи още в детската градина, като се използват различни дидактически инструменти за блоково програмиране, например Bee-Bot [1]. Така учениците ще се докоснат рано до ключови знания и умения.

Подходът, който предлагаме, не се отклонява от тази стратегия, но залага на objects-first модел – акцентира върху програмирането в началото на компютърното обучение, но включва възможно най-рано прилагането на принципите на обектноориентирания дизайн и програмиране. Основни понятия са „обект“ и „клас“. Ранно се въвеждат и ключовите механизми „наследяване“ и „полиморфизъм“. За да могат обучаемите да експериментират на този ранен етап, те трябва да използват готови или полуготови интерактивни програми. Традиционните процедурни конструкции се въвеждат паралелно на обектните като допълващи основните идеи [3].

Нашата имплементация на object-first модела е силно повлияна и от роботиката. В последните години интерес представлява усвояването на компютърни концепции чрез изучаване на изкуствен интелект. Професорът по компютърни науки Тара Хловски от Университета на Британска Колумбия смята, че той е подходящ за ученици от всички възрасти, стига да е представен по подходящ начин. Тя залага на роботиката за това. Нейната организация с нестопанска цел Iridescent се стреми всички деца да се докоснат до роботиката и чрез нея да изучават основните концепции на математиката, информатиката, физиката и т.н. [2][7].

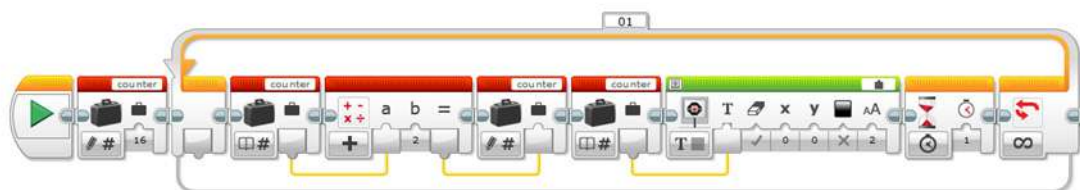
Приложението на тези идеи в нашия подход се изразява във включването в учебния курс на базови елементи от роботиката чрез използването на LEGO-роботика.

В много училища в европейските страни (Германия, Испания, Великобритания, Португалия, Нидерландия) за учениците в гимназиален етап се въвеждат на базово ниво изучаването на елементи от машинното учене. Разработиха се програмни среди с потребителски интерфейс, който намалява необходимостта от познаването на езици за програмиране [2][9]. Тази среда позволява на децата да разработват програми в Scratch (фиг. 1.) – платформа, която изучават учениците от III и IV клас в българските училища. Започвайки да програмират в Scratch, учениците ще бъдат готови в бъдеще да придобиват знания и умения за програмиране на C#, Java, Python и други езици за програмиране.



Фигура 1. Програмиране в Scratch

След демонстрацията на различни видове готов софтуер и използването на програмируеми, интерактивни ресурси учебните часове следват предвидените в учебната програма [6] теми. Направената пропедевтика помага на учениците да добият представа защо е нужно да се изучават процедурните елементи (условен оператор, циклични алгоритми, функции). Отново се връщаме към демонстрациите при изучаване на понятието „алгоритъм“ и видовете алгоритми, където включваме LEGO-робот (фиг. 2.).



Фигура 2. Програмиране на LEGO-робот

Преплитането на идеи от няколко подхода е най-подходящият метод при преподаването на програмиране.

В [5] много подробно и структурирано се прави анализ на причините за бавното навлизане на обектноориентираното програмиране в средното училище и е представен подход за смяна на стила на преподаване.

Обектноориентираното проектиране и програмиране (ООП) поставиха началото на нова ера в създаването на софтуер. Успешното им съчетаване с трите основни парадигми – процедурна, логическа и функционална, както и СУБД е още едно свидетелство за това.

Реалните предимства на ООП се проявяват при по-големите софтуерни проекти. При по-малки задания, и особено при решаване на основни задачи, тези предимства не могат да бъдат забелязани. В училище също няма време за големи проекти. Това е и главната причина за бавното навлизане на ООП в учебната практика. Набляга се на стандартни типове данни, операции с тях, библиотечни функции, управляващи конструкции и структури от данни. На това ниво няма особено значение с какъв език и в какъв стил ще бъдат практикувани [5].

В последните години педагогически експерименти, представени в различни литературни източници, доказват огромната полза от активното обучение (active learning). То включва различни подходи, обединени от едно – поставянето на обучаемия в центъра на образователния процес като активна единица. При него ученикът не е пасивен регистратор на лекторските идеи, а активен участник в обучението, което, от своя страна, се превръща в обмен на идеи.

Подходите включват техники като участие в различни видове образователни игри, експериментиране със симулационни модели, използване на компютърни системи за електронно обучение, проектно базирано обучение, работа в екип и др. Въпреки че се прилага в различни области, най-голям дял то има в природните науки, математиката и информатиката. Бурното развитие на ИКТ, интернет и социалните мрежи допринесе изключително за разширяване на възможностите за активно обучение [4].

Ще дадем няколко примера за училища и университети, които използват създаден образователен софтуер за часовете по обектноориентирано програмиране, доказвайки плавното преминаване от структурно програмиране към обектноориентирано програмиране.

В университета „Карнеги Мелън“, САЩ, за часовете по обектно-ориентирано програмиране е създаден образователен софтуер “Alice”, чрез който студентите създават сами виртуални реалности, игри и анимации, като между другото научават и структурата, функциите и алгоритмите за създаването на едно истинско приложение. Друг подобен продукт е “Greenfoot” – проект на университета в Кент, Англия. Представява интегрирана образователна среда, чиято цел е да обучи потребителите си на обектноориентирано програмиране на Java. Greenfoot комбинира програмирането на Java с графичен и интерактивен резултат, който се визуализира на екрана. Интерфейсът му е по-олекотен от този на “Alice”, като успоредно със създаването на обекти на сцената обучаемият може да следи и автоматичното генериране на код. Двата продукта са високо оценени от Оракъл. Началното обучение по програмиране в тяхната Оракъл академия се провежда с използването на тези две системи [4].

С развитието на технологиите и промяната на поколенията се налага прилагането на различни подходи за обучение. Например т.нар. „учебно скеле“ (instructional scaffolding), екипно обучение (group-based learning), технологично поддържано обучение (Technology Enhanced Learning), проектно базирано обучение, активно обучение с елементи на студентско изследване (inquiry based learning), колаборативно обучение, смесено обучение (blended learning), „подгрыващо обучение“ (Just-in-Time Teaching) [4].

Редица проучвания доказват, че употребата на подходящи технологични средства, комбинирани с правилните педагогически подходи, могат да стимулират ученето и да насърчат учениците да повишат компетентностите си и извън учебния план.

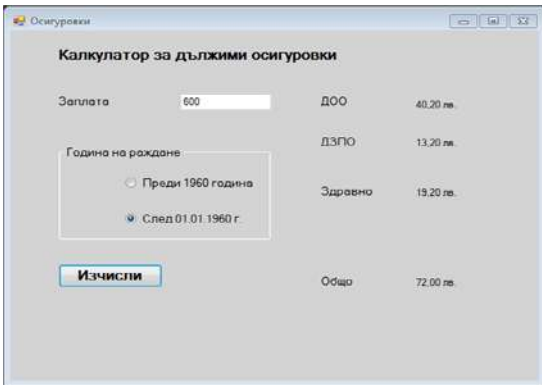


Фигура 3. Конус на познанието

Международно признатият Едгар Дейл, който е работил в Университета на Охайо, е доказал ползата от използването на аудио-визуални материали в обучението. Неговата най-известна концепция е „конус на познанието“. Чрез нея той чудесно илюстрира графично връзката между начина на представяне на учебното съдържание и степента на усвояването му. На фигура 3. е изобразен конусът на познанието.

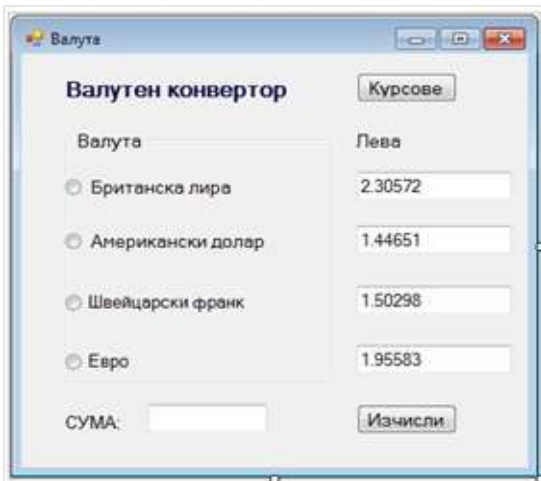
Важна част от нашия подход са примерите и задачите, които развиват уменията и затвърждават знанията на учениците. След въвеждането на всяко от заложените в учебната програма понятия е необходимо опериране с него. Предпочитаните от нас задачи са с приложен характер: създаване на различни видове калкулатори – графичен, геометричен, изчисляване на индекс на

телесната маса; приложение за наемане на автомобили; игра на морски шах и други. Опитът ни сочи, че силно положително въздействие оказва създаването на игри [3].



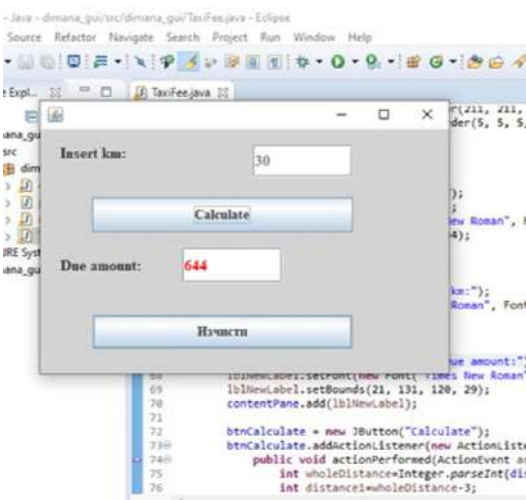
Фигура 4. Осигуровки

Следните няколко фигури демонстрират проекти, създадени от ученици, базирани на реалния живот. Езиците за програмиране, които сме използвали, са Java (JFrame) и C# (Form app). На фиг. 4. е показана главната форма на създадения от учениците софтуер за пресмятане на дължимите осигуровки в зависимост от въведената година в текстовата кутия.



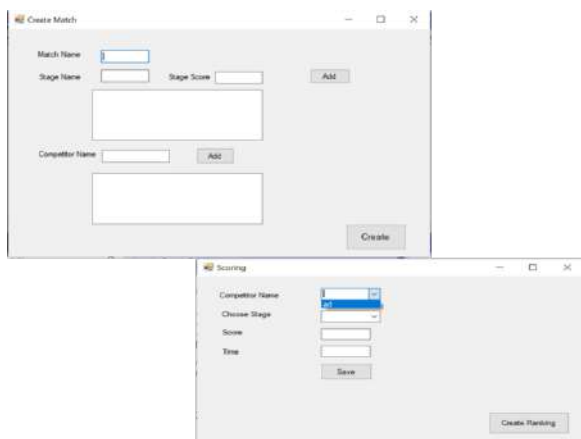
Фигура 5. Валута

Валутният калкулатор на фиг. 5. предизвика голям интерес и учениците създадоха и калкулатор за мерните единици, но за избирането им те използваха падащо меню, а не контролата радиобутон.



Фигура 6. Реализиране на аритметична прогресия

Създадохме и таксиметър на фиг. 6. и чрез това приложението на демонстрираме „Аритметична прогресия“.



Фигура 7. Създаване на мач. Реализиране на наследяване между формите.

Фигура 7. Създаване на мач. Реализиране на наследяване между формите

Проектът на фиг. 7. демонстрира много познания по предмета „Информатика“ от обектноориентираното програмиране. Едно от най-важните е „наследяване“.

Тези проекти са само част от работата на учениците. Някои са създадени по техни идеи, което е една от основните идеи на подхода. Стараем се учениците да се чувстват свободни да създават, да експериментират и да не се притесняват да сгрешат.

ЗАКЛЮЧЕНИЕ И БЪДЕЩА РАБОТА

Наблюденията ни сочат, че преподаването само на суха теория повишава нивото на разсеяност у учениците. Затова учителят трябва да наблегне на подбора на примерите и задачите, които дават на теорията реални очертания. Един добре подбран и атрактивно представен пример ще бъде по-лесно запомнен и усвоен, отколкото правилата и дефиниции, които ще бъдат зазубрени за момента и скоро забравени. Добра практика, която препоръчваме, е да се провокират учениците да създадат собствени примери, подобни на разгледаните от учителя. Опитът ни показва, че стимулирането и насърчаването на учениците да бъдат активни в учебните часове е най-трудната задача за реализиране.

Предстои обобщаване на резултатите от провеждания вече трета година педагогически експеримент за ефективността от прилагането на описания по-горе подход.

ЛИТЕРАТУРА

[1.] Ангелов, А., Момчева, Г. Програмируеми играчки в предучилищна и начална училищна възраст. Научно-практически форум „Иновации в обучението и познавателното развитие“, Бургас, 2013.

[2.] Глушкова, Т., Стоянов, Ст. Изкуствен интелект в училище – предизвикателство, необходимост, излюзия и реалност, списание „Математика+“, издателство „Архимед, брой 1, 2021.

[3.] Дончев, Ив., Чотова, Г. Реализация на objects-early подход в училищния курс по информатика, Списание „Математика, компютърни науки и образование“, 2019 / Том 2 / Брой 2. (Страници: 93-99).

[4.] Крушкова, Маринова. Методика за активно обучение по програмиране чрез използване на информационни и комуникационни технологии, Пловдив, 2014.

[5.] Крушков, Хр. От структурно към обектно-ориентирано програмиране. Списание „Математика и информатика“ – Аз Буки, Volume 59, Number 4, 2016.

[6.] МОН (2017), Учебна програма по Компютърно моделиране за IV клас. // МОН (2017), Учебна програма по Компютърно моделиране за IV клас, https://www.mon.bg/upload/13767/UP9_KM_ZP_4kl.pdf

[7.] Tara chklowski, Help communities solve real-world problems AI: become a technovation mentor!, AI Matters 5(3):12-14, DOI:10.1145/3362077.3362080, 2019

[8.] Computing Curricula 2001: Computer Science: Report of The Joint Task Force on Computing Curricula, IEEE Computer Society and The Association for Computing Machinery, December, 2001.

[9.] Програмна среда с интерфейс за разработка, който намалява необходимостта от познаването на езици за програмиране URL: <https://machinelearningforkids.co.uk/>

ИНФОРМАЦИЯ ЗА АВТОР

Габриела Чотова – докторант, Факултет „Математика и информатика“, Великотърновски университет „Св. св. Кирил и Методий“, e-mail: gabriela.chotova@gmail.com

ABOUT THE AUTHOR

Gabriela Chotova, PhD student, Faculty of Mathematics and Informatics, Department of Information Technologies, ”St. Cyril and St. Methodius” University of Veliko Tarnovo, E-mail: gabriela.chotova@gmail.com.