

ЛОГИЧЕСКИ ОПЕРАТОРИ И РАЗКЛОНЕНИ АЛГОРИТМИ В ОБУЧЕНИЕТО ПО КОМПЮТЪРНО МОДЕЛИРАНЕ

Румяна Йорданова Папанчева

Университет „Проф. д-р Асен Златаров“, Бургас

LOGICAL COMMANDS AND CONDITIONAL ALGORITHMS IN TEACHING COMPUTER SCIENCE AT PRIMARY SCHOOL

Rumyana Yordanova Papancheva

University “Prof. Dr Asen Zlatarov”, Burgas

Abstract: The aim of the paper is to present a methodological guidelines in teaching some computer science concepts in Primary school. In particular, conditional operators are considered, and conditional algorithms are constructed. The methodology aims in helping Primary school teacher to teach content from Computer Science in appropriate way, taking in account the age and the level of knowledge of primary school children.

Keywords: teaching programming at Primary school, conditional algorithms

Преподаването на предмета Компютърно моделиране, съгласно нормативната уредба у нас, е предоставено на начални учители. Възникват проблеми от факта, че тези учители нямат качествени и систематизирани базови знания от областта на информатиката. Един от начините за преодоляване на този проблем е разработването на подробни методически указания за учителя, като част от учебните ресурси, обезпечаващи преподаването на предмета в училище.

Целта на статията е да систематизира методически една от темите, заложи в учебното съдържание в четвърти клас – работа с условни команди и създаването на разклонени алгоритми. Обобщена е темата на база на ресурсите, разработени към учебника по Компютърно моделиране за 4 клас на издателство Изкуства [1].

В хода на обучение по Компютърно моделиране се формират основни знания от информатиката. След усвояването на последователни и на повтарящи се действия, следва запознаване с условните действия [1]. Условие може да се включи както в алгоритъм с последователни действия (блок за разклонение), така и в алгоритъм с повтарящи се действия (цикъл с условие).

В обучението в четвърти клас учениците се запознават с използването на блок за разклонение. Блоковете за разклонение са два вида:

1. Блок „Ако – Тогава“ (If – Then)
2. Блок „Ако – Тогава – Иначе“ (If – Then – Else)

Блокът „Ако – Тогава“ има следния формат:

Ако Условие, Тогава Действие.

Условието може да бъде изпълнено или не. Условието се задава чрез твърдение, което може да бъде вярно или невярно. Ако условието е изпълнено, казваме, че е вярно (на английски език – True). Ако условието не е изпълнено, казваме, че условието е Лъжа (на английски език False). Когато условието е Истина се изпълнява действието, записано в командата. Ако условието не е изпълнено, т.е. е Лъжа, то записаното действие не се изпълнява.

Блокът „Ако – То – Иначе“ има следния формат:

Ако Условие, То Действие 1, Иначе Действие 2

Когато условието е Истина се изпълнява записаното в командата Действие 1. Ако условието не е изпълнено, т.е. е Лъжа, то Действие 1 не се изпълнява, но се изпълнява Действие 2.

И в двата случая, ако блокът за разклонение е част от програмен код, след като се изпълни условната команда се изпълнява следващата команда в кода.

Формиране на разбиране за условен оператор

При работа с малки ученици начин на формиране на разбиране за условен оператор е използването на игрови подход. Например, урокът може да започне с условна команда, която издава учителят. Напри-

мер, учителят е приготвил изненада, която ще даде, ако учениците изпълнят някакво условие: „Ако запазят тишина за 20 секунди, тогава ще Ви дам карти и ще играем с тях през часа“.

Учителят съобщава за изненадата и следи реакцията на класа. Ако учениците не запазят тишина, не дава карти. Изчаква и отново задава командата. Повтаря докато постигне желаната тишина за 20 секунди. След това изважда комплект карти и го поставя пред учениците.

Учителят обяснява действията си. Той/Тя е задал/а условно действие. Условието е тишината за 20 секунди. Ако има тишина, условието е изпълнено и картите са на масата. Ако няма тишина, няма карти. За по-добра нагледност, учителят може да използва две флагчета – червено и зелено. Зеленият флаг се вдига, когато условието е изпълнено, т.е. е Истина. Червеният флаг се вдига, когато условието не е изпълнено, т.е. е Лъжа. Повтаря отново командата. Ако класът запази тишина, вдига зеленото знаме. Ако не – вдига червеното знаме. На този етап при вдигнато зелено знаме има следващо действие – учителят дава картите. Ако е вдигнат червения флаг – няма действие.

Друг пример – учителят хвърля зар и задава следната команда: Ако се падне четен брой точки, тогава момчетата да станат прави. Учителят хвърля зара. Нека се паднат 4 точки. 4 е четно число. Следователно условието на командата е изпълнено. Тогава трябва да се изпълни действието – момчетата трябва да станат прави. Ако се паднат 3 точки – условието не е изпълнено – всички ученици остават седнали по местата си.

Друг пример – игра с карти. Учителят осигурява по едно тесте карти за двама ученици. Играе се с картите от 2 до 10 в четирите бои. Примерна игра: Картите се разбъркват. Слагат се на купчина по средата. Първият играч обръща най-горната карта. Ако е червена, тогава той/тя печели една точка. Играе следващия играч. Правят се по 10 хода и играта спира. Печели играчът, който има повече точки.

След като се формира разбиране за командата „Ако ... тогава“, се преминава към формирането на разбиране за по-сложния вариант: „Ако тогава, иначе“

Примерна игра: учителят стартира отново играта със зар, но този път въвежда нови правила. Хвърля зара. Ако се падне четно число, тогава стават момчетата, иначе – стават момчетата. Преминава към изпълнение на командата. Учителят хвърля зара – нека се падне 3. 3 не е четно число. Следователно условието е Лъжа и учителят вдига червения флаг. В този случай трябва да станат момчетата. Обяснява се отново действието на командата „Ако – Тогава – Иначе“. Хвърля се отново зара няколко пъти, докато всички ученици разберат смисъла на условната команда от този вид.

Друг пример – игра с картите, като този път се въвежда по-сложно условие. Например: Първият играч тегли най-горната карта. Ако е червена, тогава играчът печели една точка, иначе – точка печели другият играч.

В процеса на формиране на алгоритмично мислене важен момент е умението за работа с блок-схеми. Блок-схемата се състои от блокове, които графично визуализират логиката на алгоритъма. След като се направи блок-схема на алгоритъма, той може да се превърне в код на произволен език за програмиране. Блок-схемата е и универсален език между хора, които или не знаят езици за програмиране, или знаят различни езици за програмиране.

Въвеждат се блокът за елементарно действие с форма на правоъгълник и блок за логическо условие с форма на ромб. От блока за логическо условие излизат два изхода – ДА и НЕ, които определят хода на действията в алгоритъма според това дали условието е изпълнено (т.е. е Истина) или не е изпълнено (т.е. е Лъжа). В резултат се създават т. нар. разклонени алгоритми.

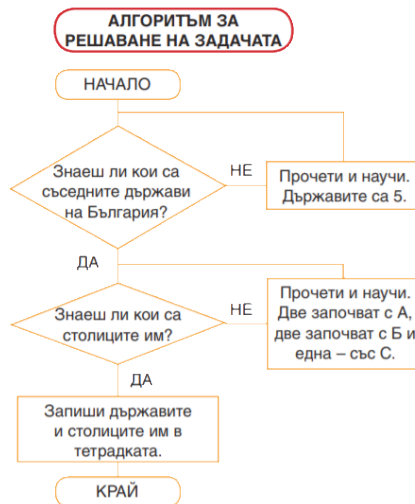
След като се формира знание за условна команда, може да се премине към създаване на разклонен алгоритъм. Примерна игра, която може да се използва за тази цел е следната: Учителят задава въпрос. Всеки ученик отговаря на въпроса за себе си и ако отговорът е ДА, тогава става прав, ако отговорът е НЕ, тогава остава седнал.

Примерен въпрос: Знаете ли кои са съседните държави на България? – стават учениците, които знаят. Учителят дава думата на избран ученик, за да назове държавите (Македония, Сърбия, Турция, Гърция, Румъния). Всички ученици сядат по местата си.

Учителят поставя нов въпрос: Знаете ли кои са столиците на съседните ни държави? – стават учениците, които знаят столиците. Учителят посочва един от правите ученици, да назове столиците на петте държави.

Учителят формулира следната задача – да се запишат съседните държави на България и техните столици. Дискутира се върху алгоритъма за решаване на задачата. За да може един човек да реши задачата, той първо трябва да си отговори на въпроса дали знае кои са държавите. Ако знае държавите, трябва да

си отговори на въпроса дали знае техните столици. Ако и на двата въпроса отговори положително, тогава може да пристъпи към изпълнение на задачата – да запише държавите и столиците им. Но ако на някой от въпросите отговори отрицателно, тогава ще трябва да направи допълнителни действия – да намери информация, за да попълни знанията си.



Фигура 1. Блок схема на разклонен алгоритъм

Алгоритъмът за решаване на поставената задача е представен чрез схема (Фиг. 1). Учителят запознава учениците с понятието „разклонен алгоритъм“. Разклоненият алгоритъм е като кръстопът. Според определено условие, избираме един или друг път, за да продължим напред. Задаваме въпрос. Ако отговорът е ДА изпълняваме едни действия, в противен случай – други действия.

Учителят обяснява схемата. Въвежда понятието „блок-схема“. Блок-схемата описва логиката на алгоритъма и начина на неговото изпълнение. Запознава учениците с основните блокове в схемата. Алгоритъма започва с блок „Начало“ и завършва с блок „Край“. Другите основни блокове са с форма на правоъгълник и с форма на диамант (ромб). В правоъгълника записваме елементарни действия. В блока с форма на ромб записваме условие. Условието е под формата на въпрос, който има отговор Да или Не.

Записан с думи, алгоритъмът от схемата е следният:

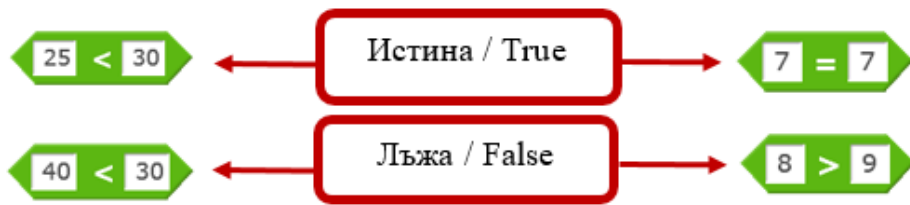
1. Начало
2. Ако знаеш кои са съседните държави на България, тогава премини към точка 2. Иначе, прочети и ги научи. Премини към точка 1.
3. Ако знаеш столиците на съседните държави, тогава ги запиши в тетрадката. Иначе, прочети и ги научи. Премини към точка 2.
4. Край

Може да се постави обратната задача – по даден алгоритъм, описан чрез блок-схема, да се разбере неговото действие.

Създаване на разклонен алгоритъм в среда за програмиране

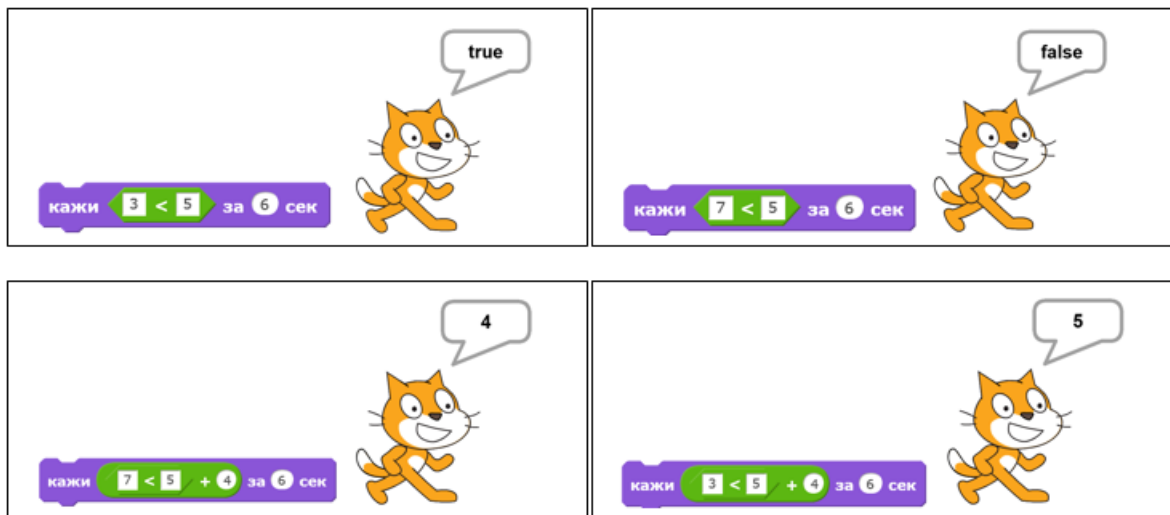
След формиране на знания за условните оператори и разклонените алгоритми се преминава към работа в конкретна среда за програмиране. При първи стъпки в програмирането обикновено се използват визуални среди за програмиране като Scratch, Code.org и др.

В програмата по Компютърно моделиране в 4 клас е предвидено учениците да се запознаят със средата Scratch. За създаване на разклонени алгоритми се използва блокът за разклонение „Ако ... то ...“. За да имаме условие, определящо избора на посока на действията, използваме оператори, генериращи условия за истина или лъжа. Такива оператори са например, операторите за сравняване, които обикновено използваме при работа с числа. В групата с командни блокове Оператори има блокове за аритметични отношения или блокове за сравняване на числа – блоковете „по-малко“, „по-голямо“ и „равно“. При изпълнение на командите за сравнение се получава резултат „Истина“ или „Лъжа“. Ако равенството или неравенството е вярно, то стойността на операцията е „Истина“ или „True“. Ако е невярно числово равенство или неравенство, тогава стойността на операцията е „Лъжа“ или „False“.



Фигура 2. Блокове за сравняване

Блокът за сравняване може да се вложи във всеки друг блок. Ако се вложи в команди като Кажи ..., Мисли ..., тогава резултатът е текст – True или False. Ако се вложи на мястото на число, тогава Истина се пресмята като 1, а Лъжа се пресмята като 0, т.е. Истина = 1, Лъжа = 0. Например при събиране на Истина с 4, резултатът ще е 5. При събиране на Лъжа с 4, резултатът ще е 4 (фиг. 3).



Фигура 3. Стойност на команда за сравняване

Най-често блоковете за сравняване се влагат в блоковете за разклонение. Те задават условието на командния блок за разклонение. Според това дали условието е истина или лъжа се изпълнява едно или друго действие. В блоковете за разклонение могат да се влагат само команди, чийто резултат е Истина или Лъжа (формата на тези блокове е със заострени краища). В условието на блок за разклонение не може да се вложи, например, блок за аритметично действие.



Фигура 4. Блок за разклонение в Scratch

Друга важна част от блоковете в групата Оператори са блоковете за логически действия. Често условията, от които зависят действията ни не са свързани само с едно ограничение, а са комбинация от отделни условия. Например, условието може да изисква да се случат две неща едновременно – числото да е по-малко от 10 и да е четно. (В този случай имаме логическата операция Конюнкция или логическото И). Може да имаме условие, което да е вярно при две възможности – например, когато числото е 4 или когато числото е 5. (В този случай имаме логическа операция Дизюнкция или логическо ИЛИ). Или пък искаме числото да не е 4 (В този случай имаме логическа операция Отрицание, или логическо НЕ). В тези случаи използваме логическите операции – И, ИЛИ, НЕ.

Резултатът от блоковете за логически операции е също Истина или Лъжа, както резултатът от блоковете за аритметични отношения. И всичко, казано за блоковете за аритметични отношения важи и за блоковете за логически операции – те имат същата форма и могат да се влагат във всички блокове (в правоъгълни, в кръгли и в ромбоидни места за попълване на командни блокове).

Преди да създават самостоятелно разклонени алгоритми, учениците трябва да се научат да разбират логиката и изпълнението на предварително създадени такива. По предварително създадени компютърни проекти, учителят може да постави водещи въпроси, за да насочи учениците към важните моменти от логиката на програмата, за да могат те сами да разберат нейното съдържание и да предвидят резултата от изпълнението ѝ.

Примерни въпроси към компютърен проект, за който учениците трябва да разберат как работи, каква е логиката му, как е реализирана са например [1]:

1. Колко декора има в проекта? Кои са те?
2. Колко герои (спрайтове) има в проекта?
3. Как се сменят декорите?
4. Какво се случва при щракване на зеленото знаме?
5. Какво е програмирана да прави Героят при смяна на декора?
6. Как се стартира нова задача?

Въпросите може да се поставят към групи от ученици. Когато работят по двойки или в екипи учениците се учат взаимно и успяват да си обяснят логиката на написания код по достъпен и разбираем за тях начин. В случая ролята учителя може да бъде само насочваща и контролираща. Важно е учениците сами да достигнат до разбирането на кода на разклонения алгоритъм, за да могат в бъдеще да създадат самостоятелно такъв алгоритъм.

Обучението по Компютърно моделиране поставя много предизвикателства пред учители и ученици. Много от конструктите, които се изучават изискват поне минимални знания по информатика, свързани с информация, оператори, структура от данни. Това очертава голямата нужда от допълнителни ресурси и практически насоки за успешното реализиране на поставената по предмета Компютърно моделиран цел за формиране на дигитални компетенции и развиване на алгоритмично и критическо мислене.

References:

1. Papancheva, R., T. Glushkova, *Kompyutarno modelirane za 4 klas, Izkustva*, Sofia, 2019.