FRI-ONLINE-1-CCT1-01

LABVIEW AS A TOOL FOR LEARNING RECURSION[1]

Prof. Tsvetozar Georgiev, PhD
Department of Computer Systems and Technologies,
"Angel Kanchev" University of Ruse
Phone: 082 888 711
E-mail: TGeorgiev@ecs.uni-ruse.bg

Abstract: *The report describes an approach to studying recursion through its visualization. Various program implementations are considered and the possibility for using the LabVIEW environment in the process of training students in the discipline Synthesis and Analysis of Algorithms is analysed. In particular, attention is paid to the resources that LabVIEW has for the development, visualisation and step-by-step interactive management by the user of the implementation of recursive functions. Exemplary realisations of direct and indirect recursion are described (calculation of factorial, summation of n consecutive numbers, calculation of Fibonacci numbers, finding the greatest common divisor (GCD), recursive binary search, etc.). The conclusion states that the LabVIEW environment is suitable both for demonstrations and to learn the basic principles of recursion.*

Keywords: *Education, Algorithms, Recursion, Animation, LabVIEW*
JEL Codes: *C60, C61, C80, I23*

ВЪВЕДЕНИЕ

Една от сложните материи за усвояване от студентите при изучаване на алгоритми е работата с рекурсия (Dann, W., Cooper, S., Pausch, R., 2001). Основните проблеми са свързани с факта, че тя е неявен цикъл, работата на който обучаемите си представят трудно.

С цел по-добро представяне на тази материя, в учебния процес много често се използват различни методи за описание и онагледяване на стъпките при реализацията на различни рекурсивни алгоритми:

- текстово описание;
- графично представяне (фиг.1а);
- изпълнение в среда за програмиране (фиг.1б);
- анимация на стъпките на рекурсията (Visualizing Algorithms, 2021).

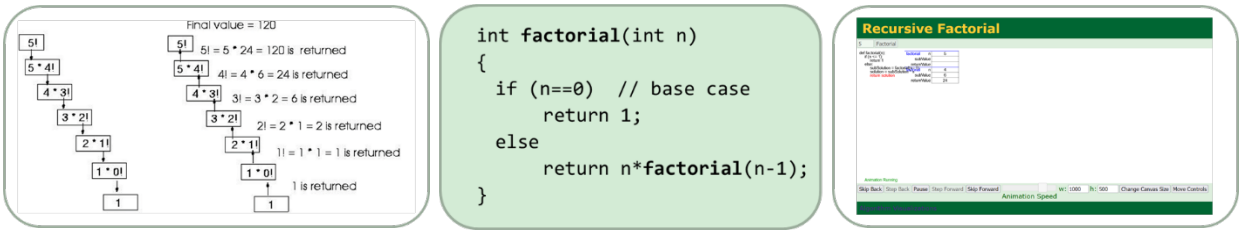
Всеки от тези методи има своите предимства и недостатъци. Предимство на текстовото описание е възможността точно и последователно да бъдат описани отделните стъпки на даден рекурсивен алгоритъм, но сериозен недостатък е трудното възприемане от студентите (Wilcocks, D., Sanders, I., 1994) .

При графичното представяне имаме онагледяване на рекурсивния процес, което прави възприемането по-лесно, но като недостатък може да се отчете, че то най-често показва едно статично състояние и също е трудно за възприемане (Sambamurthy, N., Edgcomb, A., Vahid, F., 2019).

Предимство на изпълнението в средата за програмиране е възможността на обучаемите да проследят постъпковото изпълнение на даден рекурсивен алгоритъм и да получат реален резултат от неговата работа. Недостатък е, че извикването на копията на рекурсивната функция и работата със стека остават невидими за обучаемите.

[1] Докладът е представен на заседание на секция 3.2 на 29 октомври 2021 с оригинално заглавие LABVIEW AS A TOOL FOR LEARNING RECURSION

При анимационното представяне на рекурсия потребителите имат възможност да проследят извикването на всяко копие на рекурсивната функция, записването му в стека, достигането до крайното условие и изплуването от рекурсията (Velazquez-Iturbide, J., Perez-Carrasco, A., 2010). Като недостатък може да се посочи, че създаването на такива анимирани приложения е по-сложно от останалите методи и изисква повече време и ресурси.



а) б) в)

Фиг. 1. Представяне на рекурсивни алгоритми:
а) графично; б) в среда за програмиране; в) анимирано (Recursive Factorial, 2021)

Много често се налага използване на смесен подход, при който едновременно се прилагат няколко метода.

В доклада е описан един нов подход за обучение на студенти за работа с рекурсивни алгоритми, като се използват възможностите на средата за програмиране LabVIEW. Този подход съчетава възможностите на третия и четвъртия метод за онагледяване – изпълнение в среда за програмиране и едновременно с това анимация на основните стъпки.

ИЗЛОЖЕНИЕ

Среда за програмиране LabVIEW

Някои от основните характеристики на LabVIEW, по които тя се различава от традиционните среди за програмиране, са следните (Larsen, R., 2014):

- програмите, разработени на LabVIEW, се наричат виртуални инструменти;
- програмите се състоят от преден панел (потребителски интерфейс) и блокова диаграма (програмна реализация);
- за всяка програма трябва да се създаде икона и конектор (входно-изходни връзки);
- използва графичен език за програмиране G;
- програмира се потокът на данните;
- всеки виртуален инструмент може да извика друг такъв посредством неговата икона и конектор;
- притежава мощни инструменти за откриване на грешки;
- притежава многобройни вградени функции (програмни, математически, статистически, за комуникация и др.).

Една от ключовите характеристики, които правят използването на LabVIEW подходяща среда за демонстрация на рекурсивни алгоритми, е, че изпълнението на дадена блокова диаграма може да бъде анимирано чрез използване на бутон Highlight Execution. Този начин най-често се прилага при постъпково изпълнение с цел проследяване потока на данни в блоковата диаграма (Bishop, P., 2015). За разрешаване на този режим трябва да се щракне върху бутон Step Into или Step Over. След изпълнението на тази операция първият елемент от блоковата диаграма започва да мига, което показва, че е готов да изпълни операцията. След това може да се щракне отново върху Step Into или Step Over, за да се изпълни дадената операция и да се премине към следващия елемент. Ако следващият елемент е структура или виртуален инструмент, може да се щракне върху бутон Step Over. По този начин елементът ще изпълни операцията, без да се проследява постъпково неговото изпълнение. Ако се щракне върху бутон Step Into, това ще предизвика постъпково изпълнение на елемента, представляващ структура или виртуален инструмент. За прекратяване на изпълнението на

блоковата диаграма или за завършване на постъпковото изпълнение трябва да се щракне върху бутон Step Out.

Средата за програмиране LabVIEW притежава и някои характерни особености при работа с рекурсия, които я правят подходяща за използване в процеса на обучение:

- поддържа рекурсивно изпълнение на програми (Yang, Y., 2014);
- дава възможност за постъпково изпълнение, анимиране и проследяване на последователността на предаване и получаване на данни от отделните функционални блокове и между копията на рекурсивната програма:
  - при извикване всяко ново копие на рекурсивната програма автоматично се визуализира на екрана с преден панел и блокова диаграма;
  - при завършване на изпълнението на действието на дадено копие на рекурсивната програма, неговата визуализация автоматично се затваря.

Реализирани рекурсивни алгоритми

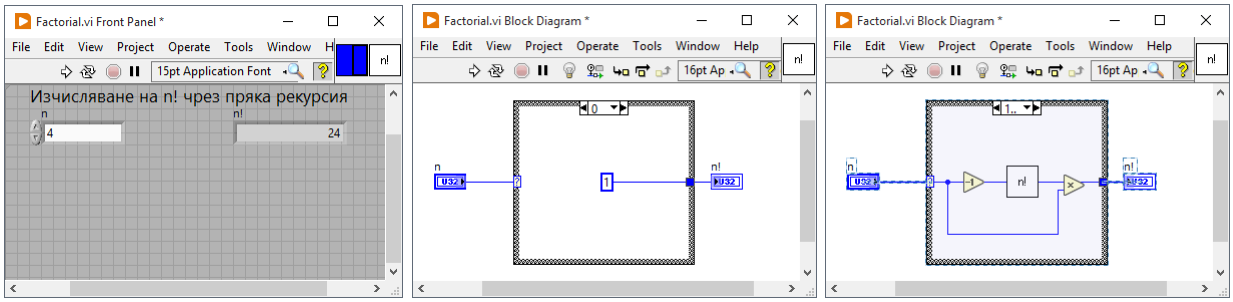
В много от книгите и учебниците за алгоритми са описани различни преки и косвени рекурсивни реализации (Welss, A., 2003) посредством използване на първия и втория метод – текстово описание и графично представяне (Drozdek, A., 2012). Някои от най-често разглежданите рекурсивни алгоритми са за изчисляване на факториел (Fellah, A., Bandi, A., 2018), изчисляване числата на Фибоначи (Pal, D., Halder, S., 2018), намиране на най-голям общ делител на две числа (Jain, H., 2017), сумиране на елементите на целочислен масив (Goodrich, M., Tamassia, R., Mount, D., 2011) и други.

За целите на обучението на студентите по дисциплината „Синтез и анализ на алгоритми“ са реализирани няколко различни алгоритъма за пряка и косвена рекурсия.

За пряка рекурсия тези алгоритми са:

- сумиране на последователни числа;
- повдигане на число на степен;
- изчисляване на факториел;
- намиране на най-голям общ делител на две числа;
- изчисляване числата на Фибоначи;
- двоично търсене;
- обръщане на реда на цифрите на дадено число.

Един от многото реализирани примери е за изчисляване на факториел (фиг.2). Програмата се състои от преден панел (фиг.2а), в който обучаемият може да зададе стойност на променливата n за изчисляване на n! и да види крайният резултат от работата на алгоритъма. Блоковата схема (фиг.2б и 2в) представлява програмният код, реализиран на езика G. Крайното условие за излизане от рекурсията n=0 се задава в case блок (фиг.2б) и тогава стойността, която се връща е 1. Преките рекурсивни извиквания на програмата се реализират посредством включване на собствената икона (n!) на програмата в case блока, когато n>0 (фиг.2в). При изплуване от рекурсията до началната програма, която инициира рекурсивните извиквания, на предния панел се извежда резултатът от изчисленията.



а) б) в)

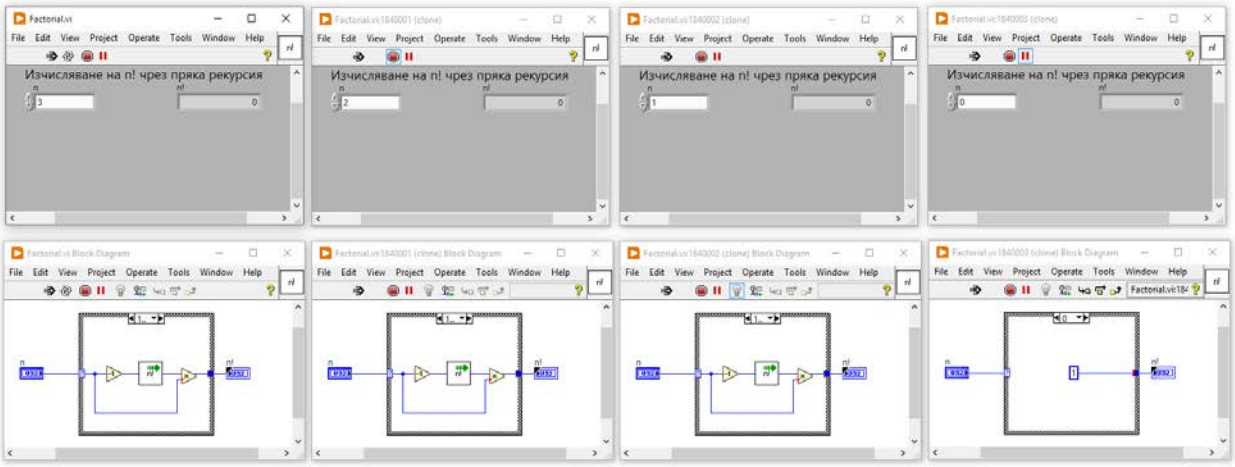
Фиг. 2. Реализация на пряка рекурсия за изчисляване на n! в LabVIEW:
а) преден панел; б) задаване на крайно условие n=0 за излизане от рекурсията;

в) рекурсивно извикване на програмата за изчисляване на n! при n>0

Алгоритъмът за работа с приложението за изчисляване на n! с пряка рекурсия е следният (фиг.3):

1) от предния панел на програмата се задава стойност на n;
2) от блоковата диаграма се стартира постъпково изпълнение на програмата (Highlight Execution);
3) последователно се щраква се върху бутон Start Single Stepping / Step Into … ;
4) при достигане до постъпково изпълнение на рекурсивната програма (Step Into subVI "Factorial.vi") на екрана ще се появи новото копие на програмата с преден панел и блокова диаграма;
5) продължава се работата с блоковата диаграма на новото копие, като се изпълняват последователно стъпки от 3 до 5;
6) при достигане до крайното условие n=0 на последното копие трябва да се щракне върху бутон Finish VI …, след което копието на програмата се затваря и изчислената текуща стойност за n! се предава на копието от по-горно ниво (изплуване от рекурсията);
7) отново се извършва постъпково изпълнение, докато се изчисли новата стойност на n!;
8) щраква се върху бутона Finish VI …, след което копието на програмата се затваря и изчислената текуща стойност за n! се предава на копието от по-горно ниво (изплуване от рекурсията);
9) стъпки 7 и 8 се повтарят, докато се достигне до началната програма и на нейния преден панел се получи крайният резултат от работата.

Тази последователност от действия може да се използва и при другите реализирани примери за пряка рекурсия.



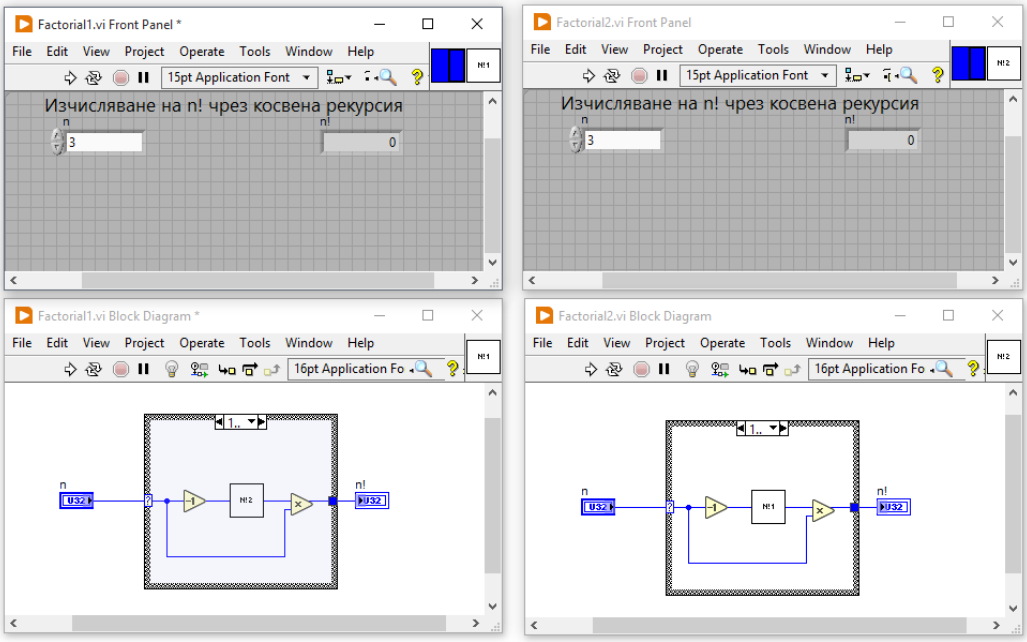
а) б) в) г)

Фиг. 3. Постъпково изпълнение на пряка рекурсия за изчисляване на 3! в LabVIEW:
а) начална програма; б) първо копие на програмата; в) второ копие; г) трето копие

За косвена рекурсия реализираните алгоритми са:

- изчисляване на факториел;
- изчисляване на стойностите на две взаимно зависими функции.

Един от примерите за косвена рекурсия е за изчисляване на факториел. За целта бяха реализирани две програми – Factorial1 и Factorial2 (фиг.4), които се извикват взаимно една друга (посредством техните икони №1 и №2 в блоковата диаграма на другата програма) с намаляваща стойност на променливата n до достигане до крайното условие n=0. Изпълнението на рекурсията може да започне както от първата, така и от втората програма, като резултатът от работата се извежда на предния панел на програмата, от която е започнало изпълнението.

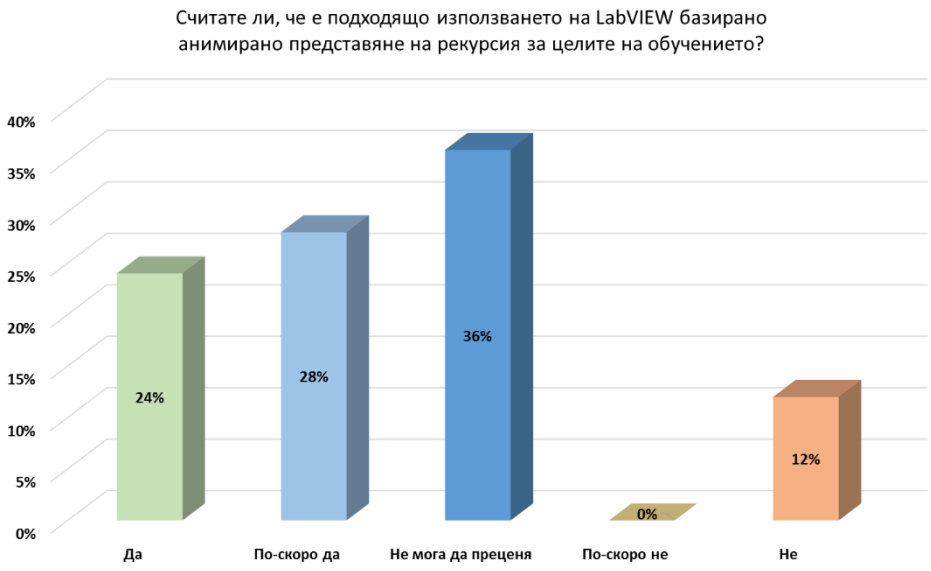


а) б)

Фиг. 4. Реализация на косвена рекурсия за изчисляване на n! в LabVIEW:
а) програмата Factorial1 извиква програмата Factorial2;
б) програмата Factorial2 извиква програмата Factorial1

Резултати от проучване мнението на студенти

За да се прецени доколко използването на LabVIEW за анимирано представяне на рекурсия е удачно за използване в учебния процес, беше проведено анонимно анкетно проучване сред 38 студенти, които изучават дисциплината „Синтез и анализ на алгоритми“. На въпроса „Считате ли, че е подходящо използването на LabVIEW базирано анимирано представяне на рекурсия за целите на обучението?“ повече от половината студенти отговориха положително (24% с „Да“ и 28% с „По-скоро да“), 36% отговориха, че не могат да преценят, а останалите 12% посочиха отговор „Не“ (фиг.5). Получените резултати са обнадеждаващи предвид факта, че студентите до този момент не бяха работили с LabVIEW.



Фиг. 5. Резултати от отговора на въпроса от анкетата

ЗАКЛЮЧЕНИЕ

Средата LabVIEW осигурява много добри възможности за разработване на преки и косвени рекурсивни алгоритми. Тя успешно може да бъде използвана за интерактивно управление и анимирана визуализация на тяхната работа.

Над 50% от анкетираните студенти считат, че LabVIEW базирано анимирано представяне на рекурсия е подходящо да се използва за целите на обучението. Това ще доведе до затвърждаване на техните знания и повишаване на качеството на обучението по дисциплината „Синтез и анализ на алгоритми“.

REFERENCES

Bishop, P. (2015). Learning with Labview. Pearson Education Inc.

Dann, W., Cooper, S., Pausch, R. (2001). Using visualization to teach novices recursion. *In Proceedings of the 6th annual conference on Innovation and technology in computer science education*,109-112.

Drozdek, A. (2012). Data Structures and algorithms in C++. Cengage Learning.

Fellah, A., Bandi, A. (2018). The essence of recursion: Reduction, delegation, and visualization. *Journal of Computing Sciences in Colleges*, 33(5), 115-123.

Goodrich, M., Tamassia, R., Mount, D. (2011). Data structures and algorithms in C++. John Wiley & Sons.

Jain, H. (2017). Problems Solving in Data Structures & Algorithms Using C++. Createspace Independent Publishing Platform.

Larsen, R. (2014). LabVIEW for Engineers. Pearson Education Inc.

Pal, D., Halder, S. (2018). Data structures and algorithms with C. Alpha Science International Ltd.

Sambamurthy, N., Edgcomb, A., Vahid, F. (2019). Animations for Learning: Design Philosophy and Student Usage in Interactive Textbooks. *In 2019 ASEE Annual Conference & Exposition*.

Velazquez-Iturbide, J., Perez-Carrasco, A. (2010). InfoVis interaction techniques in animation of recursive programs. *Algorithms*, 3(1), 76-91.

Welss, A. (2003). Data structures and problem solving using C++. Pearson Education International.

Wilcocks, D., Sanders, I. (1994). Animating recursion as an aid to instruction. *Computers & Education, 23(3)*, 221-226.

Yang, Y. (2014). LabVIEW graphical programming cookbook. Packt Publishing Ltd.

Recursive Factorial, https://www.cs.usfca.edu/~galles/visualization/RecFact.html, последно посетен на 16.09.2021.

Visualizing Algorithms, https://bost.ocks.org/mike/algorithms/, последно посетен на 24.09.2021.