

# РАЗВИВАНЕ НА УМЕНИЯ ПО ПРОГРАМИРАНЕ ЧРЕЗ СИСТЕМА MATHEMATICA

Анна Малинова

## РЕЗЮМЕ

*Статията представя изследване на възможностите за изграждане на умения по програмиране чрез системата за компютърна алгебра Mathematica. Дадени са примери от лекционния курс и лабораторните упражнения по дисциплината „Информационни технологии в математиката”, свързани както с доразвиване на уменията в областта на процедурното програмиране, така и с въведение в неизучавани от студентите парадигми на програмиране като функционалното програмиране и програмирането, базирано на правила.*

**Ключови думи:** система Mathematica, умения по програмиране, процедурно програмиране, функционално програмиране, програмиране, базирано на правила.

Необходимостта от иновации в обучението по математика и природните науки отдавна е осъзнат факт и обект на многобройни изследвания и разработки. Иновациите в математическото образование могат да включват както нови дидактически подходи (GG, 2009), интегриране на знания по математика и информатика в специализираните курсове по математика в средното училище (RG, 2009), добри практики в обучението на ученици (SR, 2010) и на учители (RA, 2009), свързани с използване на информационни технологии, така и с използването на динамичен софтуер в обучението по математика (ЧЕХ, 2010), (VAR, 2009).

В този поток от дейности могат да се открият и редица български и международни проекти. В (ГР, 2008) и (GR, 2008) са представени проекти с участието на ФМИ към ПУ „П. Хилендарски”, а именно изграждане на междуфакултетен разпределен център за електронно обучение и изграждането на европейска виртуална лаборатория по математика. В (КЕН, 2010) се представят европейски проекти с българско участие, свързани с разработването, внедряването и разпространението на иновативни дидактически концепции и педагогически стратегии в математическото образование, основаващи се на нови образователни технологии, които са тясно свързани и със съвременните информационни технологии. Проектите

предвиждат дейности, свързани със следните три целеви групи: ученици, учители и студенти. Във връзка с третата група – студентите, се предвижда в учебното съдържание на университетите да се включат теми, отразяващи най-съвременните тенденции на ИКТ в математическото образование.

Настоящата статия представя изследване на възможностите за изграждане на умения по програмиране чрез системата за компютърна алгебра Mathematica (WRES, 2010). Проведен е експеримент, свързан с обучението на студенти първокурсници от специалност „Бизнес информационни технологии” по дисциплината „Информационни технологии в математиката” към ФМИ на ПУ „П. Хилендарски”. С обучението по тази дисциплина си поставяме следните задачи: 1) запознаване на студентите с основните принципи на системите за компютърно подпомагане на решаването на математически задачи; 2) запознаване с основните принципи за създаване и споделяне на документи с математически текст. Реализирането на първата задача е с помощта на средата Mathematica (ГОЧ, 2009). Крайната цел на този процес е даване на възможност за изграждане на Mathematica-базирани курсове по математическите и някои от икономическите дисциплини, които студентите ще изучават в по-горните курсове. Представени са примери от лекционния курс и лабораторните упражнения, свързани както с доразвиване на уменията в областта на процедурното програмиране, така и с въведение в неизучавани от студентите парадигми на програмиране като функционалното програмиране и програмирането, базирано на правила.

## СИСТЕМА МАТЕМАТИКА КАТО СРЕДА ЗА ПРОГРАМИРАНЕ

Система Mathematica предоставя мощен език за програмиране, включващ различни парадигми и методи от съвременните езици за програмиране, като добавя и нови идеи (WOL, 2003). Mathematica е език, базиран основно на системи от правила за трансформации и на техники за функционално, процедурно и обектно-ориентирано програмиране. Същевременно в Mathematica са интегрирани богати възможности за работа със символни низове, свързани не само с регулярни изрази, но и със собствен символен език за манипулиране на низове чрез шаблони.

Mathematica поддържа голям набор от структури от данни (ШВГ, 1995) или Mathematica обекти (функции, множества, списъци, масиви, таблици, матрици, вектори и др.), както и операции върху тези обекти (проверка за тип, селектиране, композиция и др.).

Системата Mathematica може да бъде разширявана с потребителски програми и пакети (MD, 2007), като същевременно дава богати възможности за работа с външни файлове и за импортиране и експортиране на данни и графика в различни формати (WOL, 2003).

## ПРОЦЕДУРНО ПРОГРАМИРАНЕ

Процедурното програмиране е стилът на програмиране, с който студентите са вече запознати от училищния курс по информатика, както и от изучаваната в първия триместър на първи курс дисциплина „Основи на компютърната информатика” и изучаваната паралелно през втори триместър дисциплина „Програмиране”. Mathematica поддържа всички стандартни конструкции, характерни за процедурното програмиране.

Лекционният курс прави обобщение на процедурното програмиране. Представят се средствата на Mathematica за дефиниране на променливи, присвояване на стойност, дефиниране на потребителски функции, условни разклонения, цикли, управление на потока на изпълнение, управление на видимостта и жизнения цикъл. Едновременно с това се показват и разискват и такива специфични за Mathematica подходи извън парадигмата на процедурното програмиране, като отложеното присвояване при дефинирането на променливи и функции (WOL, 2003).

На Фиг. 1 е показан пример от Упражнение 7 (Решаване на задачи от аналитичната геометрия. Координатни системи. Уравнения на права и равнина. Взаимни положения, разстояния. Криви от втора степен.). Представен е един от начините за решение на б) подточка на задачата:

*Да се намери уравнението на права, минаваща през две точки:*

*а) уравнение на права, минаваща през точките  $A1(1,2)$  и  $A2(3,-3)$ ;*

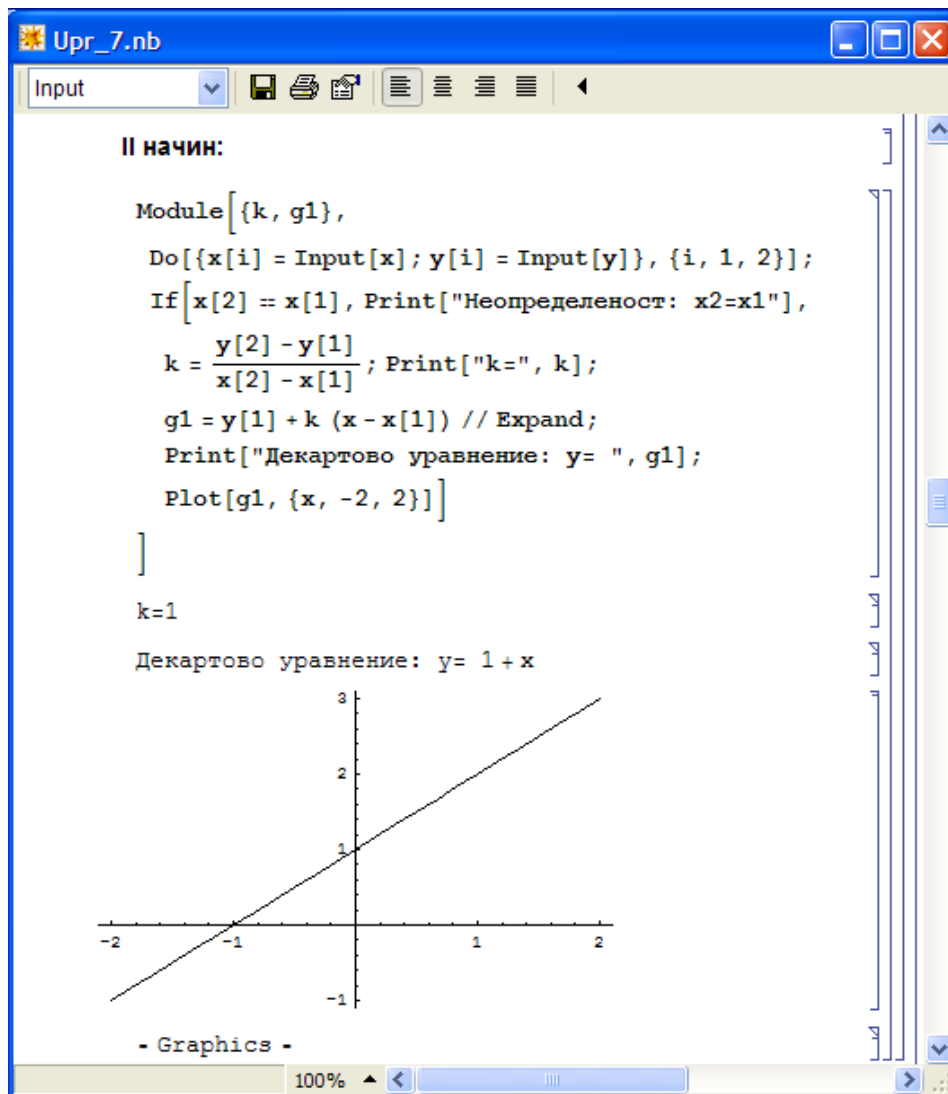
*б) декартово уравнение на права, минаваща през кои да е две зададени точки.*

## ФУНКЦИОНАЛНО ПРОГРАМИРАНЕ

Функционалното програмиране е може би най-важният стил на програмиране в Mathematica (RUS, 2009). В основата си програмирането в Mathematica представлява създаване на потребителски функции, които, приложени върху дадени аргументи, извършват изчисления и връщат резултат. Поради унифицираното представяне на всички обекти в Mathematica като изрази, в действителност функциите в Mathematica могат да оперират с произволни изрази, включително и други функции (WKG, 2005). Този функционален стил на програмиране отличава Mathematica от традиционните езици за програмиране и дава възможност за пълно възползване от изчислителната мощ на средата.

До момента на запознаване със системата Mathematica студентите не са изучавали функционално програмиране. В процеса на обучение по „Информационни технологии в математиката” се разглежда прилагането на основни принципи на функционалното програмиране и предимствата на този подход, без студентите да се запознават в детайли със специфичните за функционалното програмиране терминология и вътрешни механизми. Вместо

това се работи на по-високото ниво на абстракция, предлагано от Mathematica, и с Mathematica терминологията. Например, в курса се разглеждат: списъци, шаблони, функции за манипулиране на изрази (Map, Apply, Thread); многократно прилагане на функция върху израз – итеративни функции (Nest, NestList, Fold, FoldList); чисти (анонимни) функции (Function, #, &), но не се борави с понятия като примитивни изрази, комбинации, специални форми, термове, филтър на термовете, лямбда изрази, конструктори, селектори (ТОД, 2003).



**Фиг. 1.** Процедурно програмиране – пример за вход от клавиатурата, цикъл, условен оператор, локални променливи в модул

```

In[3]:= countPrimes[n_] := Length[Select[Range[n], PrimeQ]]
In[4]:= countPrimes[20]
Out[4]= 8

a) решение във функционален стил

In[1]:= sieve[n_] := Module[{broi, result, i},
  result = Range[n];
  result[[1]] = 0; i = 2;
  While[i^2 ≤ n,
    If[result[[i]] ≠ 0, For[j = 2 * i, j ≤ n, j += i, result[[j]] = 0]]; i++
  ];
  broi = 0;
  For[k = 1, k ≤ n, k++, If[result[[k]] > 0, broi++]];
  broi
]

In[2]:= sieve[20]
Out[2]= 8

б) решение в процедурен стил

```

**Фиг. 2.** Функционално програмиране а) в сравнение с процедурното програмиране б)

На Фиг. 2 е показан пример от лекционния материал, който сравнява решението на задачата за намиране на броя на простите числа, по-малки от дадено число по два метода:

1) функционално програмиране - чрез прилагане на възможността да се използва функция като аргумент на друга функция и филтриране на списък чрез предикатна функция;

2) процедурно програмиране – решение чрез „решето на Ератостен”.

Чрез примера от Фиг. 2 се акцентира на лекотата и интуитивния подход при функционалния метод на програмиране в сравнение с процедурния стил. От значение е и много по-краткото решение в първия случай.

На Фиг. 3. е показан пример от Упражнение 5 (Решаване на задачи от линейната алгебра. Детерминанта на матрица – пресмятане, минори и адюнгирани количества. Обратна матрица, ранг на матрица, намиране на базисен минор). Представен е един от начините за решение на задачата:

*Да се намери броят на базисните минори на дадената матрица.*

Примерът показва дефиниране на единствена потребителска функция  $f$  с една локална променлива  $d$  за получаването на крайния резултат, което

включва конструиране на едномерен списък чрез филтриране на матрицата от минорите с помощта на шаблон и преброяване на елементите от всеки ред, отговарящи на шаблона, намиране на сумата на елементите на генерирания списък, използване на анонимна функция с анонимни аргументи като аргумент на друга функция, извличане на елемент на списък и др.

```

II начин: Броят на минорите на m3, които са различни от нула и имат
ред, равен на ранга на матрицата, намираме по следния начин:

In[69]:= m3 =  $\begin{pmatrix} 3 & 5 & 0 & 7 \\ -1 & -3 & -3 & -5 \\ 3 & 2 & -5 & 1 \\ 2 & 3 & 0 & 4 \end{pmatrix};$ 

In[70]:= f[n_] := Module[{d},
  d = Dimensions[Minors[n, 3]];
  d[[1]] * d[[2]] -
  Total[Array[Count[Minors[n, 3][[#]], _?_0] &, d[[1]]]]
]

In[71]:= f[m3]

Out[71]:= 9

```

Фиг. 3. Функционално програмиране – пример

### ПРОГРАМИРАНЕ, БАЗИРАНО НА ПРАВИЛА

Употребата на правила за трансформиране на израз от една форма в друга е един от най-мощните инструменти, предоставени от Mathematica като език за програмиране (WKG, 2005). В лекционния материал се показва, че програмирането, базирано на правила използва символите, свързани с присвояване (=, :=), трансформации (→, :>) и шаблони ( \_ ), без да се налага въвеждане на нови конструкции или оператори за програмиране. На практика студентите се запознават с прилагането на правила за трансформация още от първото упражнение върху система Mathematica, но чак по-късно това се коментира като специфичен стил на програмиране при решение на голям клас от задачи, които обикновено се свързват с по-традиционните конструкции, характерни за процедурното и функционалното програмиране.

Отделя се внимание на задаването на шаблони и на разбирането, че програмата, използвайки съвпадение на шаблони, замества срещанията на променливите в правилото, докато повече замествания не са възможни.

На Фиг. 4 са показани някои примери от лекционния материал, демонстриращи силата и лекотата на използване на този подход.

```

In[32]:= {a, b, c} /. List -> Plus
Out[32]= a + b + c
а) прилагане на правило за заместване на един функционален израз с друг
In[33]:= {{1, 2}, {3, 4}, {5, 6}} /. {x_, y_} -> {y, x}
Out[33]= {{2, 1}, {4, 3}, {6, 5}}
б) прилагане на правило за трансформация с шаблон
In[38]:= Clear[f];
In[39]:= f[1] = f[2] = 1;
In[40]:= f[n_] := f[n - 1] + f[n - 2] /; IntegerQ[n] && Positive[n]
In[37]:= {f[-3], f[7]}
Out[37]= {f[-3], 13}
в) прилагане на правило за заместване в резултат на присвояване (незабавно и отложено присвояване)

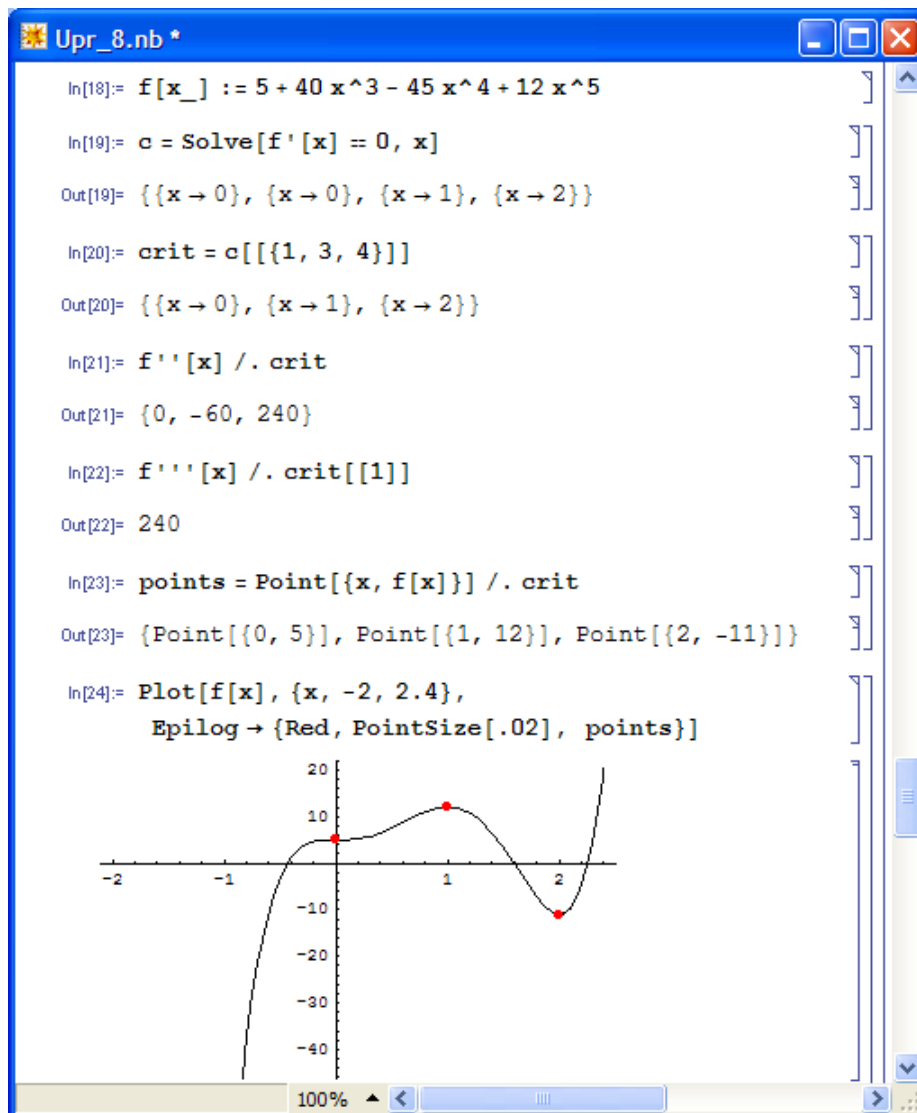
```

**Фиг. 4.** Програмиране, базирано на правила – примери

На Фиг. 5 е показан пример от Упражнение 8 (Решаване на задачи от анализа - производни граници, суми. Графика на функция. Изследване на функция). Представено е едно решение на задачата :

*Да се определят интервалите на растене и намаляване, екстремумите и инфлексните точки на функцията  $y=5+40x^3-45x^4+12x^5$ .*

Примерът демонстрира неколккратно прилагане на правила за трансформации – както единично правило, така и списък от правила.



**Фиг. 5.** Програмиране, базирано на правила – пример за прилагане на единична трансформация и на списък трансформации

### ЗАКЛЮЧЕНИЕ

Статията демонстрира част от богатите възможности на система Mathematica за обучение по програмиране, което е базирано на различни парадигми, някои от които не са изучавани предварително от студентите. Поради ограничение в обема не бяха разгледани примери за създаване на потребителски интерфейс,



интегриране с външни системи, обектно ориентирано програмиране, обработка на символни низове и др. с Mathematica.

Обучението на студенти в използване на система Mathematica показва, че компютърното подпомагане на решаването на математически задачи способства за развиване както на умения за боравене с изучавания математически апарат, така и на разнообразни умения по програмиране. Съчетаването на математически знания с умения по програмиране прави възможно бързото решаване на голям брой задачи от различен тип и сложност и поставянето на акцент върху разбирането и осмислянето на получените резултати и експериментирането с различни начални условия.

### ЛИТЕРАТУРА

ГОЧЕВА-ИЛИЕВА, С. & РАХНЕВ, А. (ГР), (2008) Междуфакултетен Разпределен Център за е-обучение в Пловдивския университет „Паисий Хилендарски”, Межд. конф. Компютърни методи в науката и образованието, Варна, 12-14.09, 2008.

ГОЧЕВА-ИЛИЕВА, С. (ГОЧ), (2009) Въведение в система Mathematica, Габрово: ЕксПрес.

КЕНДЕРОВ, П. (КЕН), (2010), Иновации в математическото образование: европейските проекти InnoMathEd и Fibonaccі, Сборник доклади на Тридесет и девета пролетна конференция на Съюза на математиците в България, Албена 2010, доклад по покана, стр. 63-73.

ТОДОРОВА, М. (ТОД), (2003) Езици за функционално и логическо програмиране. Първа част – функционално програмиране, София: Сиела.

ЧЕХЛАРОВА, Т. (ЧЕХ), (2010) Изследвания в обучението по математика чрез симулиране с динамичен софтуер, Национална конференция „Образованието в информационното общество”, Пловдив, 27-28 май 2010 г., стр. 205-212.

ШИШКОВ, Д., ВЪРБАНОВ, М., ГОЛЕВ, А. & др. (ШВГ), (1995) Структури от данни, Добрич: Интеграл.

GANCHEV, I., GROZDEV, S. (GG), (2009) On Two Fundamental Approaches to the Development of Scientific Knowledge and their Execution in Didactics of Mathematics, Proceedings of the 6th Mediterranean Conference on Mathematics Education “MEDCONF 2009”, Plovdiv, 22-26 April, 2009, pp. 17 – 27.

GOCHEVA-ILIEVA, S. & RAHNEV, A. (GR), (2008) New Challenges in e-Learning of Mathematics via EVLM and IDeLC Projects, in New Aspects of Engineering Education, Proc. of the 5th WSEAS / IASME Int. Conf. on Engineering Education (EE'08), Heraklion, Crete Island, Greece, July 22-24, 2008, Published by WSEAS Press, invited paper, pp. 437-442.

MORALIYSKI, R. & DIAS, G. (MD), (2007) One sense per discourse for synonymy extraction, International Conference On Recent Advances in Natural Language Processing, RANLP 2007, Borovets, Bulgaria, pp. 383–387.

RAHNEV, A. & ANGELOVA, E. (RA), (2009) Training Teachers of Mathematics in the Use of Modern Information Technologies for Teaching, Proceedings of the 6th Mediterranean Conference on Mathematics Education “MEDCONF 2009”, Plovdiv, 22-26 April, 2009, pp. 79 – 83.

RAHNEV, A. & GAROV, K. (RG), (2009) Integrating Mathematics and Informatics Content Knowledge in Specialized Mathematics Training, Proceedings of the 6th Mediterranean Conference on Mathematics Education “MEDCONF 2009”, Plovdiv, 22-26 April, 2009, pp. 365 – 374.

RUSKEEPAA, H. (RUS), (2009) Mathematica navigator: mathematics, statistics, and graphics, 3rd ed. USA: Elsevier, pp. 541-615.

STARIBRATOV, I. & RAHNEV, A. (SR), (2010) Enhancing Mathematics Education through Information Technologies Integration, Scientific Works, Plovdiv University, vol. 37, book 3, 2010 – Mathematics (to appear).

VARBANOVA, E. (VAR), (2009) Computer Algebra – a Good Reason to Enhance Interest and Enthusiasm in Mathematics Classes, Proceedings of the 6th Mediterranean Conference on Mathematics Education “MEDCONF 2009”, Plovdiv, 22-26 April, 2009, pp. 279 – 288.

WELLIN, P., KAMIN S. & GAYLORD R. (WKG), (2005) An Introduction to Programming with Mathematica. New York: Cambridge University Press, pp. 75-176.

WOLFRAM, S. (WOL), (2003) The Mathematica Book, 5<sup>th</sup> ed. USA: Wolfram Media.

WOLFRAM RESEARCH (WRES), (2010) Mathematica 7.0,  
<http://www.wolfram.com/products/mathematica/index.html>.

## DEVELOPING PROGRAMMING SKILLS USING “MATHEMATICA”

Anna Malinova

### ABSTRACT

*This paper presents a study about the possibilities for developing programming skills through the use of the computer algebra system Mathematica. Presented are lecture and laboratory examples from the course Information technologies in mathematics, connected with the development of procedural programming skills, as well as with the introduction of paradigms yet unfamiliar to the students, such as functional and rule-based programming.*

**Keywords:** Mathematica, programming skills, procedural programming, functional programming, rule-based programming.

Anna Malinova  
Faculty of Mathematics and Informatics  
University of Plovdiv  
236 Bulgaria Blvd., 4003 Plovdiv, Bulgaria  
e-mail: malinova@uni-plovdiv.bg