

МАТЕМАТИКА С КОМПЮТЪР

Сава Гроздев, Деко Деков

Резюме. Авторите дискутират въпроса за използването на език за програмиране в средното училище за решаване на задачи от учебния материал по математика. Предложена е програма за решаването на конкретна задача.

Keywords: problem solving, computer, programming language, browser, JavaScript, PHP, the money-exchanging problem

С навлизането на компютрите в училище се появяват възможности образованието по математика да стане по-ефективно. Подходящо е учениците да решават задачи от учебния материал и извън него с помощта на компютърни програми, съставени от самите тях. Съответните дейности могат да стартират в 9. клас, когато се изучават основите на информатиката. Трябва да се отбележи обаче, че при писане на програми за решаване на математически задачи е възможно използването само на малка част от език за програмиране, като за разбирането и използването на тази малка част не е необходимо учениците да са запознати с учебния материал по информатика. Поради тази причина, ползването на език за програмиране за решаване на математически задачи може да стане и преди 9. клас.

Един език за програмиране, който да е подходящ за използване от всеки ученик в средното училище, трябва да отговаря на няколко критерия. На първо място той трябва да бъде максимално лесен. Учениците не би трябвало да се затрудняват при използването му. Освен това, езикът за програмиране трябва да е широко използваем, за да може учениците да са убедени, че учат нещо, което всекидневно се прилага навсякъде по света. Езикът трябва да е достъпен и да не изисква инсталирането на допълнителен софтуер. От няколко години имаме на разположение такъв език за програмиране. Това е JavaScript.

Езикът JavaScript е максимално лесен за изучаване и използване. При писане на компютърни програми за решаване на математически задачи е достатъчно да се използва само малка част от него, която учениците могат лесно и бързо да усвоят. Освен това, JavaScript е най-широко използваният език за програмиране в днешно време – това е езикът за програмиране в Интернет от страна на клиента. През 2010 г. JavaScript са ползвали около 99.4% от уеб сайтовете във връзка с програмиране от страна на клиента. JavaScript удовлетворява и условието да

може да бъде използван без инсталиране на допълнителен софтуер. Достатъчен е браузър, а днес браузър има на всеки компютър.

Конкурент на JavaScript би могъл да бъде езикът за програмиране PHP. Този език е вторият по разпространение. PHP е основният език за програмиране в Интернет от страна на сървъра. През 2010 г. от уеб сайтовете, които ползват език за програмиране от страна на сървъра, 75.9% се падат на PHP. Този брой расте устойчиво през последните 10 години. Както JavaScript, така и PHP е предназначен да обслужва Интернет, което определя възможностите му. Както при JavaScript, така и при PHP, за решаване на математически задачи е необходима само малка част от езика. Както JavaScript, така и PHP е максимално прост и лесен за употреба. Предимство на PHP пред JavaScript е великолепната система за корекции, която е част от езика. Ако потребителят има някаква грешка в кода, системата за корекции посочва точно на кой ред от кода и каква точно е грешката. Това е голямо удобство. Предимство на PHP е и наличието на голям брой вградени функции за работа с низове и масиви. При някои програми за решаване на задачи по математика посоченото предимство е съществено. Друго предимство на PHP е във възможността да бъде използван съвместно с база данни, което може да се осъществява съвместно с различни бази данни. Използва се главно с релационната база данни MySQL. Релационните бази данни се прилагат с помощта на езика SQL – лесен и забавен, с който се постигат отлични резултати. Недостатък на комплекта PHP и MySQL е, че за ползването на тези софтуерни инструменти е необходима връзка с Интернет и хостинг, т.е. необходимо е инсталирането на специален софтуер на компютъра. В днешно време съществуват готови програми от типа на WAMP, които могат да бъдат инсталирани с едно щракване на мишката. След това PHP и MySQL могат да се ползват без проблеми. Трябва да се отбележи, че ако PHP и MySQL са инсталирани, това води до съществени удобства във връзка с полесното управление на ползваните файлове. В този случай можем да инсталираме и програма от типа на HeidiSQL, която предлага допълнителни удобства при управлението на базата данни. За решаване на математически задачи база данни не е необходима, но изкуството да бъде ползвана базата данни MySQL с помощта на PHP и SQL се усвоява лесно, а и е полезно. Съгласно статистиката, през 2010 г. най-търсените програмисти в света са били именно тези с PHP.

Една друга устойчива тенденция е използването на готови шаблони на уеб сайтове. Тези шаблони, с незначителни изключения, са изготвени с помощта на JavaScript, PHP и MySQL. С малки промени на шаблона потребителят получава готов уеб сайт с пълна функционалност – например магазин за електронна търговия, корпоративен сайт, портален сайт, сайт на банка, сайт на университет, сайт за дистанционно обучение, сайт на държавна институция, личен сайт и други. По-

настоящем фаворит при шаблоните е Wordpress. До 2010 г. вече над 20 милиона уеб сайта са били изготвени с използването на този шаблон.

Всички изброени по-горе технологии са безплатни и не изискват от потребителя нито стотинка за ползването им. Впрочем, това е една от причините за успеха на въпросните технологии.

В България се предлагат и други езици за програмиране, главно в университетите. Един професионален програмист трябва задължително да владее C/C++ и Java. Това се отнася обаче до професионалните програмисти. Не всеки ученик в средното училище планира да стане програмист. Ако целта е да използваме лесен за усвояване и употреба език за програмиране за решаване на математически задачи, подходящ избор е JavaScript или PHP. Учениците, които се готвят за олимпиади по информатика, също могат да използват JavaScript или PHP. Всъщност, що се касае до задачите по математика, PHP има едно предимство пред C/C++, Java и JavaScript, а именно – в PHP има вградена библиотека за работа с числа с произволна точност. Това предимство обаче при училищните задачи по математика е без значение, тъй като такива задачи изискват решението да бъде с ограничена точност, най-много до пет или десет верни знака след десетичната запетая на число в десетичен запис. Все пак, възможността да получим решение на задачата със сто или хиляда верни цифри след десетичната запетая би могло да привлече някои ученици.

Трябва да отбележим, че PHP, MySQL и JavaScript са платформено-независими, т.е. те могат да се използват с всяка операционна система. Това е важно обстоятелство, тъй като езици за програмиране, лансирани от отделни фирми, като C#, Visual Basic, фирмени варианти на C/C++, Java и др., зависят от решенията на фирмата, която ги предлага, а практиката показва, че много често подобни технологии след време не се поддържат и не могат да бъдат използвани. Преди време вторият от авторите на тази статия изготви компютърни програми с използване на DirectX, след това Майкрософт се отказа от устойчиво поддържане на DirectX, поради което компютърните програми станаха неизползваеми. Подобни ситуации се срещат доста често, поради което платформената независимост е важна.

Езиците за програмиране, споменати по-горе, а именно C/C++, Java, JavaScript и PHP имат сходен синтаксис, произлизащ от C. Усвояването на този общ синтаксис при един език за програмиране подпомага усвояването на друг език от тази група. По-стари езици за програмиране от типа на Visual Basic, Паскал и други подобни имат различен синтаксис, което води до неудобства за потребител, желаещ да усвои модерен език за програмиране. Друг недостатък на по-старите езици е този, че те днес имат ограничено използване или въобще не се използват и не могат да бъдат полезни в реалния живот.

Задачи по математика, решавани с компютърни програми

При съставяне на компютърна програма за решаване на дадена математическа задача ученикът не извършва пресмятания, а пише компютърен код. След като кодът е готов, пресмятанията се извършват от компютъра. По този начин, като напише една компютърна програма, ученикът си спестява пресмятанията.

Кое е по-удобно – да пресмятаме или да пишем код? Авторите поддържат тезата, че всъщност писането на една малка компютърна програма в много случаи е по-лесно от пресмятанията, които трябва да бъдат извършени, ако не ползваме компютърен код. По този начин използването на език за програмиране в много случаи облекчава учениците и прави по-лесно решаването на задачите. Освен това, писането на компютърни програми помага да се усвоят по-добре алгоритмите за решение на задачите. Учениците се съсредоточават върху самия алгоритъм, пропускайки пресмятанията.

С малки и лесни за писане компютърни програми могат да бъдат решени широк кръг от задачи от учебния материал по математика. Усвояването на писането на такива програми има още едно съществено предимство. То отваря вратата и към възможността да бъдат решени редица задачи, които не са включени в учебния материал, но които са полезни за учениците. На тази тема авторите ще се спрат по-подробно в друга статия.

Ще отбележим, че усвояването на уменията да се решават математически задачи с помощта на компютърни програми трябва да бъде част от образованието по математика и трябва да е грижа на учителите по математика, а не на тези по информатика, защото всъщност се касае именно за решаване на задачи. Не е без значение и фактът, че за усвояване на PHP и JavaScript в последните години има издадени няколко книги на български език. Тези книги обаче не могат да бъдат считани за подходящи, тъй като в тях се разглеждат онези аспекти на съответни езици за програмиране, които се използват за добро функциониране на Интернет, а на частта от тези езици, която може да се използва за решаване на математически задачи, не се обръща внимание.

За реализиране на идеите, изложени по-горе, е необходимо учителите и учениците да разполагат с кратко учебно пособие, изучаването на което ще позволи да се усвои такава част от съответен език за програмиране, която е необходима за решаване на математически задачи. С поредица от примери учениците ще могат да усвоят и писането на програми. Изучаването на пособието би дало възможност да се дублират традиционните методи за решаване на задачи и да се видят предимствата на компютърния подход. В много случаи тези предимства са впечатляващи. Освен това, изучаването на пособието ще даде възможност учениците да решават и кръг от полезни задачи, доста по-широк от този, който включва днешния

учебен материал. Доколкото е известно на авторите, понастоящем не съществува подходящо пособие и те планират да създадат такова. Авторите се надяват, че пособието ще бъде полезно и на ученици с повишен интерес към математиката, за кръжочна работа, при подготовка за участие в олимпиади и т.н.

Задачата за размяната на един лев на стотинки

Задачата за размяна на един лев на стотинки не е включена в учебния материал по математика в средното училище. Независимо от това, тук ще използваме тази задача като пример. Задачата се формулира лесно. Искаме да разменим един лев на стотинки. По колко начина може да стане това? Това е една от популярните задачи, които се срещат в книги по математика и по информатика. В англоезичната литература задачата се формулира като задача за размяна на един долар на центове.

Горната задача се разглежда в параграф 7 на (Кац & Улам, 1968). Това е времето преди появата на персоналните компютри и не е учудващо, че Кац и Улам говорят за тази задача като за задача, усилието за решаването на която „много бързо води до обезкуражаване, тъй като се убеждаваме в неговата почти пълна безнадеждност“.

Задачата за размяна на един лев на стотинки може да бъде формулирана, както следва: *Колко различни решения има уравнението*

$$100 = x_1 + 2x_2 + 5x_3 + 10x_4 + 20x_5 + 50x_6,$$

където x_1, \dots, x_6 са неизвестните, които могат да приемат като стойности само цели неотрицателни числа? В това уравнение x_1 е броят на монетите от по 1 стотинка, x_2 е броят на монетите от по две стотинки и т.н.

Подходът, за който говорят Улам и Кац, и който би ни обезкуражил, ако решим да го използваме, е следният: Да започнем да проверяваме една след друга шесторки от цели неотрицателни числа (x_1, \dots, x_6) , за да видим кои от тези шесторки са решения на уравнението. Същият подход обаче ни дава лесно решението на задачата, ако използваме компютърна програма.

В разглеждания пример ще използваме езика за програмиране РНР. Читателят може лесно да усвои изкуството да ползва РНР, а веднъж усвоено, това изкуство би се оказало доста полезно. За целта е необходимо, ако операционната система на компютъра е Windows, да се инсталира програмата WAMP. След като програмата е инсталирана, файловете с разширение `php` се поставят в директорията `www` на тази програма, а се стартират от браузъра. РНР предлага редица удобства, като например да посочим колко секунди може да продължи изпълнението на един скрипт. Това може да се отбележи във файла `php.ini`. По подразбиране из-

пълнението на един скрипт може да продължи не повече от 30 секунди, но можем да увеличим времето до часове, ако е необходимо. Читателят би могъл да усвоява удобствата постепенно.

При писането на компютърната програма всъщност описваме начина, по който трябва да бъде решена задачата. Както знаем, едно ясно описание на начина на решаване на една задача, се нарича алгоритъм. Що се касае до трансформирането на алгоритъма в код на език за програмиране, това е процедура, която е доста праволинейна и може да бъде добре усвоена след неголеми тренировки.

Необходимо е да знаем понятието масив (array) и да използваме това понятие. Ако ползваме само монети от една стотинка, ще можем да ползваме не повече от 100 такива монети за размяна. Дефинираме масив, означен с \$X1, елементите на който са числата 0,1,2,...,100. Променливата x_i приема като стойности елементи от този масив. При размяната можем да ползваме не повече от 50 монети по 2 стотинки. Дефинираме съответен масив \$X2 с елементи 0, 1, 2, ..., 50, а също така и масиви \$X3, ..., \$X6, имащи за елементи възможния брой на другите видове монети. Кодът е следният:

```
<?php
echo „Money-exchanging problem.<br>“;
echo „Given 1 lev and denominations of 1,2,5,10,20,50
stotinki.<br>“;

for ($i = 0; $i <= 100; $i++)
{
    $X1[$i]= $i;
}

for ($i = 0; $i <= 50; $i++)
{
    $X2[$i]= $i;
}

for ($i = 0; $i <= 20; $i++)
{
    $X3[$i]= $i;
}

for ($i = 0; $i <= 10; $i++)
{
```

```
$X4[$i]= $i;
}

for ($i = 0; $i <= 5; $i++)
{
$X5[$i]= $i;
}

for ($i = 0; $i <= 2; $i++)
{
$X6[$i]= $i;
}
```

Вече сме готови да проверим дали една шесторка от цели неотрицателни числа (x_1, \dots, x_6) е решение на задачата. За всяка шесторка компютърът пресмята дали е решение. Шесторките, които са решения, включваме в масив, който означаваме с \$A. Дефинираме и променлива, означена с \$n, стойност на която е броят на елементите на масива \$A. За целта използваме вградената функция count() на PHP.

```
foreach ($X1 as $x1)
{
foreach ($X2 as $x2)
{
foreach ($X3 as $x3)
{
foreach ($X4 as $x4)
{
foreach ($X5 as $x5)
{
foreach ($X6 as $x6)
{
$c = 1*$x1 + 2*$x2 + 5*$x3 + 10*$x4 + 20*$x5 + 50*$x6;

if ($c == 100)
{
$A[] = „($x1, $x2, $x3, $x4, $x5, $x6) “;
}}}}}}

$n = count($A);
```

Остава само да визуализираме решението на екрана на браузъра, т.е. да визуализираме елементите на масива \$A. Можем да използваме конструкцията echo, която се използва, за да бъде изобразен върху екрана на браузъра текста, следващ в кавичките след echo. Същата конструкция използваме и в началото на програмата, за да отбележим коя задача решаваме. За наше удобство номерираме наредените шесторки, които са решения на задачата:

```
echo „<br>Answer:<br>“;
echo „There are $n exchanges.<br>“;
echo „List of all exchanges:<br>“;
for ($i = 0; $i < $n; $i++)
{
    $j = $i + 1;
    echo „$j. &nbsp; $A[$i]<br>“;
}
?>
```

В PHP променливите се означават, като пред името се поставя знакът за долар (\$). Това позволява както на програмиста, така и на евентуалния читател на кода веднага да различи променливите. Цикълът for, условията if и foreach са стандартни и тук няма да ги поясняваме. Дефиницията на масив с цикъл for е един от стандартните начини за дефиниране на масив. Всеки файл, който включва код на php, има разширение php, като в този файл кодът на php се огражда с отваряща скоба <?php и затваряща скоба ?>.

Кодът, даден по-горе, записваме в текстов файл с разширение php. Той може да се напише с текстов редактор, като един удобен такъв е например Wordpad. Удобен текстов редактор е и безплатната програма Notepad++. Освен другите удобства, този текстов редактор номерира редовете. Ако имаме грешка в кода, системата за корекции на PHP посочва номера на реда, на който е грешката. Така с помощта на Notepad++ можем веднага да видим къде точно е грешката. След като щракнем с мишката, за да стартираме php файла, получаваме отговора върху екрана на браузъра. При този пример е възможно да се наложи да изчакаме няколко секунди, докато отговорът се появи. Това е така, защото компютърът трябва да извърши относително голям брой пресмятания.

Отговорът е произведен като HTML файл и ако желаем, можем да запишем този файл върху диска на компютъра. За разглеждания пример файлът с кода на php, отговорът в HTML формат и отговорът, презаписан в PDF формат, са включени във файла answers.zip. Виждаме, че един лев може да бъде разменен на стотинки по 4562 начина, като всички тези начини са визуализирани на екрана на браузъра като номерирани наредени шесторки от цели неотрицателни числа. Първата от тези наредени шесторки ни съобщава, че един лев може да бъде разменен с две

монети от по 50 стотинки. Втората шесторка – един лев може да бъде разменен с пет монети от по 20 стотинки и т.н. В какъв вид браузърът ще изобрази отговора, зависи разбира се от нас, т.е. от това, как сме указали в кода да бъде изобразен отговорът. Фотография на част от екрана на браузъра е дадена по-долу:

```
Money-exchanging problem.  
Given 1 lev and denominations of 1,2,5,10,20,50 stotinki.  
Answer:  
There are 4562 exchanges.  
List of all exchanges:  
1. (0,0,0,0,0,2)  
2. (0,0,0,0,5,0)  
3. (0,0,0,1,2,1)  
4. (0,0,0,2,4,0)  
5. (0,0,0,3,1,1)  
6. (0,0,0,4,3,0)  
7. (0,0,0,5,0,1)  
8. (0,0,0,6,2,0)  
9. (0,0,0,8,1,0)
```

Задачата за размяна на един лев на стотинки е една от многото илюстрации на разликата между писането на код и извършването на пресмятания. Да съобщим на компютъра да извърши пресмятания е едно, а да извършим ние пресмятанията, е друго. Първото почти винаги е по-лесно. В този пример кодът е общо едва една страница, а отговорът заема сто страници във файла, който е в PDF формат. Не винаги разликата е толкова голяма както при тази задача, но почти винаги сме облагодетелствани от обстоятелството, че си спестяваме пресмятанията.

Веднъж написана, компютърната програма може да бъде използвана многократно, евентуално с малки промени. Малки промени в горната компютърна програма са необходими, ако искаме да решим вариантите на задачата, дадени по-долу.

Задачи за читателя

1. По колко начина може да бъде разменен един долар с монети от по 1, 5, 10, 25 и 50 цента?
2. По колко начина може да бъде разменена една монета от 50 цента с по-дребни монети от по 1, 5, 10 и 25 цента?
3. По колко начина може да бъде разменена една монета от 25 цента с по-дребни монети от по 1, 5 и 10 цента? Тази задача е дискутирана в книгата (Нивергелтъ, Дж.Фаррар & Э. Рейнголд, 1971), на стр 68.

Отговорите на горните задачи, както и кодът за решаване на задачите, са включени във файла answers.zip и могат да бъдат изтеглени от линка <http://azbuki.bg/editions/journals/mathinfo/contents>.

ЛИТЕРАТУРА

Гроздев, С., Деков, Д. Математика с компютър, в подготовка.

Кас, М. & Улам, С. (1968). Mathematics and Logic, New York, превод на руски език:

Кац М. & Улам С. (1971). Математика и логика, издателство „Мир“, Москва.

Нивергелт, Ю., Фаррар, Дж. & Рейнголд, Э. (1977). Машинный подход к решению математических задач, издателство „Мир“, Москва.

MATHEMATICS WITH COMPUTER

Abstract. The authors discuss the use of programming language to solve curriculum mathematical problems in high school. A programme is proposed for the solution of a particular problem.

Sava Grozdev

✉ Professor, Doctor in Mathematics, DSc in Pedagogy
Institute of Mathematics and Informatics – BAS
Acad. G. Bonchev Street, bl. 8
1113 Sofia, Bulgaria
E-mail: sava.grozdev@gmail.com

Deko Dekov

✉ Associated Professor, PhD in Mathematics
81, Zahari Knjazhevski Str.
Stara Zagora
E-mail: ddekov1@gmail.com