

ИНВАРИАНТИ В ОБУЧЕНИЕТО ПО ПРОГРАМИРАНЕ

Елена Сомова, Йордан Енев, Георги Тотков

*ПУ „Паисий Хилендарски”, Факултет по математика и информатика, 4000 Пловдив,
България, yordan@enevsoft.net, eledel@uni-plovdiv.bg, totkov@uni-plovdiv.bg*

INVARIANTS IN LEARNING OF PROGRAMMING

Elena Somova, Yordan Enev, Georgi Totkov

*The University of Plovdiv „Paisii Hilendarski”, Faculty of Mathematics and Informatics, 4000
Plovdiv, Bulgaria, yordan@enevsoft.net, eledel@uni-plovdiv.bg, totkov@uni-plovdiv.bg*

ABSTRACT

The main goal of the research is to discover the programming algorithms, which are taught in basic courses of imperative programming and which do not depend on used algorithmic language. These algorithms can be presented as template codes in different programming languages. The paper shows an approach for teaching imperative language programming on the base of set of templates, called invariants. 98 invariants of algorithms from course “Programming” and 44 invariants of algorithms from course “Algorithms and data structures” are proposed. Example programming codes on two languages (C# and Visual Basic) are given for the particular realization of invariants. The work represents the first step in realization of software system for learning programming on the base of invariants.

Key words: teaching programming, invariant algorithms in programming, programming languages

1. Въведение

Обучението по програмиране е тясно свързано с използване на конкретни програмни езици и версии. Честата смяна на програмен език или версия повдига въпроса за методиката на преподаване и на учебните материали, независеща от конкретните използвани софтуерни средства, а използваща инвариантни елементи.

Съществуват опити за решаване на подобен проблем в областта на обучението по информационни технологии и основи на компютърната информатика.

В рамките на проекта RAPIDITY, свързан с развитието и провеждането на дистанционно обучение в областта на ИТ, са създадени учебни материали [10, 11] с инвариантни елементи на 3 езика (български, френски и литовски). В резултат на изследванията и резултатите на проекта RAPIDITY е създаден базовият университетски курс „Основи на компютърната информатика“ [6]. Издадено е учебно помагало по Информатика [5] (от Регионалния център за дистанционно обучение към ПУ), в което темата „Текстообработка“ е развита според споменатата методика. На базата на инвариантни елементи също е създаден пакет от учебни пособия за средното училище [1, 2, 5, 6, 7, 8].

Идеята за инвариантно обучение е развита в [3, 4, 9], като са дадени примери за инвариантно преподаване на електронни таблици в [3, 4] и компютърна текстообработка в [9].

В [9] се поставя акцент върху т. нар. ‘инвариантни знания’ (т. е. тези, които са относително статични във времето и устойчиви на евентуални промени) при представяне/преподаване на учебното съдържание.

В [3, 4] се предлага разработването на ‘инвариантна рамка’, която да бъде определена като стандарт за инвариантно представяне на учебното съдържание, чийто предмет е конкретна технология от областта на ИТ и за която са определени елементите, които да съдържа.

Статията показва подход за преподаване на програмирането на алгоритмичен език, независимо от конкретния език, на базата на съвкупност от шаблони, наречени **инварианти**. Всеки шаблон се представя на съответния език за програмиране чрез примерни кодове.

2. Инварианти в програмирането

Първата цел на изследването е да се открият основните алгоритми (инварианти) при преподаване на програмирането на императивен език, които не зависят от преподавания език и да се представят като шаблонни кодове на различни езици. **Втората цел** е на базата на тези инварианти да се разработи система, подходяща за обучение, която да предоставя приятелски визуален интерфейс за изучаване на инвариантите в програмирането, сглобяване на програми от инварианти и проверка на програми чрез съществуващите инварианти.

Статията отразява резултатите от първия етап на изследването, като представя откритите инварианти в преподаването на програмиране на императивни езици.

За преподаване на дисциплината „Програмиране” (П) са предложени 98 инварианти, а по „Алгоритми и структури от данни” (АСД) – 44 инварианти. Инвариантите по „Програмиране” обхващат темите за: примитивни типове данни, съставни типове данни, основни оператори за управление и подпрограми, а инвариантите по „Алгоритми и структури от данни”: динамични линейни и нелинейни структури от данни, алгоритми за търсене и сортиране и други основни алгоритми върху динамични структури от данни.

Предложените инварианти са реализирани с шаблонни кодове на два езика за програмиране (C# и Visual Basic) с повече от 170 реализации на език, т.к. някои инварианти са реализирани по няколко начина (с различни алгоритми или оператори).

3. Видове инварианти

Инвариантите са систематизирани в **13 групи** на базата на основните видове подусловия на задачи (алгоритми), които се решават при обучението по програмиране:

- Деклариране на данни – 15 броя (8 броя за П и 7 броя за АСД);
- Въвеждане на данни – 11 броя (за П);
- Извеждане на данни – 14 броя (7 броя за П и 7 броя за АСД);
- Въмъкване на данни – 9 броя (2 броя за П и 7 броя за АСД);
- Изключване на данни – 12 броя (5 броя за П и 7 броя за АСД);
- Преобразуване на данни – 4 броя (за П);
- Обхождане на данни – 14 броя (7 броя за П и 7 броя за АСД);
- Търсене на данни – 5 броя (за П);
- Сортиране на данни – 10 броя (8 броя за П и 2 броя за АСД);
- Проверка за условие – 17 броя (10 броя за П и 7 броя за АСД);
- Пресмятане на разни – 10 броя (за П);
- Подалгоритми – 5 броя (за П);
- Математика – 16 броя (за П).

Предложените инварианти се разглеждат в зависимост от параметрите, които имат. **Параметрите** са **от 5 типа** (вж. Таблица 1) – променлива, тип на данни, произволен инвариант, инвариант от определен списък и инвариант от даден вид. Инвариантите могат да имат един или няколко параметри от един тип или от различен тип. Параметрите (променливи и типове на данни) на инвариантите могат да бъдат входни, изходни или входно-изходни, а останалите параметри са само входни.

Таблица 1. Типове параметри на инварианти

Тип	Параметри	Избор на подинвариант
1	променлива	няма
2	тип на данни	няма
3	произволен инвариант	избор от всички инварианти
4	инвариант от определен списък	избор от дадени инварианти
5	инвариант от даден вид	избор на инвариант от определен вид

За всеки от типовете параметри са дадени примери в Таблица 2 до Таблица 6. Структурата на инвариантите с техните параметри е особено важна при по нататъшното проектиране на системата за обучение по програмиране.

За всеки от типовете параметри са дадени като пример по един инвариант с два конкретни шаблонни кода на C# и Visual Basic (вж. от Таблица 2 до Таблица 6). Параметрите на инвариантите в шаблонните кодове са дадени с наклонен шрифт.

При използването на *инварианти с* първия тип параметри – *променливи* (вж. Таблица 2), при решаване на задачи (чрез сглобяване на решения на базата на инварианти) могат да възникнат проблеми с наименоване на променливите при използване на повече от една променлива. В такава ситуация всяка нова променлива трябва да получава ново име.

Таблица 2. Пример за инвариант с параметър от Тип 1

Инвариант	Намиране на сумата <i>sum</i> на елементите на масив <i>arr</i> с <i>n</i> елементи	
Входни параметри	<i>arr</i> , <i>n</i>	
Изходни параметри	<i>sum</i>	
Шаблонен код	<pre>int <i>sum</i>=0; for (int <i>i</i>=0; <i>i</i><<i>n</i>; <i>i</i>++) <i>sum</i>+=<i>arr</i>[<i>i</i>];</pre>	<pre>Dim <i>sum</i> As Integer = 0 For <i>i</i> As Integer = 0 To <i>n</i>-1 Step 1 <i>sum</i> = <i>sum</i> + <i>arr</i>(<i>i</i>) Next <i>i</i></pre>
	Пример на C#	Пример на Visual Basic

При *инвариантите с* параметри от втория тип – *тип на данни* (вж. Таблица 3) трябва да се прави избор само за името на подходящия тип.

Всички примери по-долу в текста комбинират някои от вече изложените видове параметри. В текущия пример трябва да зададем два параметъра-променливи *x* и *y*, които са входно-изходни параметри за инварианта и тип на данните, записани в променливите *x* и *y*. Чрез този инвариант могат да се разместват стойностите на променливи от различни типове.

Таблица 3. Пример за инвариант с параметър от Тип 2

Инвариант	Размяна стойностите на две променливи x и y	
Входни параметри	x, y, тип	
Изходни параметри	x, y	
Шаблонен код	<i>min</i> buf= x; x=y; y=buf;	Dim temp as <i>min</i> temp = x x = y y = temp
Примерен тип на данни	<i>int</i>	<i>Variant</i>
	Пример на C#	Пример на Visual Basic

При *инвариантите*, които имат параметри от третия тип – *произволен инвариант*, може да се избира кой да е инвариант от общото множество от инварианти (вж. Таблица 4). За всяка конкретна ситуация трябва да се избере подходящия подинвариант. В примера от Таблица 4 за подинвариант могат да се използват инвариантите за изход.

Таблица 4. Пример за инвариант с параметър от Тип 3

Инвариант	Делимост на числото a на числото b	
Входни параметри	a, b, инвариант	
Изходни параметри	–	
Шаблонен код	if (a%b == 0) <i>инвариант</i> ;	If (a Mod b = 0) Then <i>инвариант</i>
Пример за подинвариант	Console.WriteLine (“{0} се дели на {1}”, a, b);	Console.WriteLine (“{0} се дели на {1}”, a, b)
	Пример на C#	Пример на Visual Basic

За *инвариантите* с четвърти тип параметър – *инвариант от определен списък* се избира инвариант от списък от дадени инварианти в зависимост от условието на задачата (вж. Таблица 5). В случая на примера от Таблица 5 се избира от два подинварианта, които са едни и същи (*Проверка дали a е в релация rel с b*), но с различни стойности на параметрите: (*arr[i], arr[i+1], <*) или (*arr[i], arr[i+1], >*), които дават възможност този код да решава и двете задачи: сортиране в нарастващ ред и сортиране в намаляващ ред.

Таблица 5. Пример за инвариант с параметър от Тип 4

Инвариант	Сортиране на масив arr с n елемента по метода на мехурчето	
Входни параметри	arr, n, тип, списък от инварианти: Проверка дали a е в релация rel с b (arr[i], arr[i+1], <), Проверка дали a е в релация rel с b (arr[i], arr[i+1], >)	
Изходни параметри	arr	
Шаблонен код	<pre> <i>min</i> buf; for (int i=1; i<n; i++) for (int j=0; j<n-i; j++) <i>инвариант</i> { buf=arr[j]; arr[j]=arr[j+1]; arr[j+1]=buf; } </pre>	<pre> Dim buf As <i>min</i> For i As Integer = 1 To n-1 Step 1 For j As Integer = 0 To n-i Step 1 <i>инвариант</i> buf = arr(j) arr(j) = arr(j+1) arr(j+1) = buf End If Next j Next i </pre>
Пример за подинвариант	if (arr[j]>arr[j+1])	If (arr(j) > arr(j+1))
	Пример на C#	Пример на Visual Basic

При *инвариантите* с параметри от последния пети тип – *инвариант от даден вид* (вж. Таблица 6) се налага да се избере един подинвариант от определения вид. Например за подинвариант от вида *Извеждане на данни* (този вид включва всички инварианти, които служат за извеждане на данни от различни типове) може да се избере инварианта *Извеждане на цяло число a* при параметър *a* със стойност *arr[i]*, както е избрано в Таблица 6. Чрез този инвариант се осигурява извеждането на елементите на масив, независимо от какъв тип са.

Таблица 6. Пример за инвариант с параметър от Тип 5

Инвариант	Извеждане на масив arr от n елемента	
Входни параметри	arr, n, инвариант: Извеждане на данни (arr[i])	
Изходни параметри	–	
Шаблонен код	<pre> for (int i=0; i<n; i++) { <i>инвариант</i> } </pre>	<pre> For i As Integer = 0 To n-1 Step 1 <i>инвариант</i> Next i </pre>
Пример за подинвариант	Console.WriteLine("arr[{0}] = {1}", i, arr[i]);	Console.WriteLine("arr({0}) = {1}", i, arr(i))
	Пример на C#	Пример на Visual Basic

Възможно е в инвариант да се съдържа друг точно определен инвариант. Например в инварианта от Таблица 2 се съдържа подинварианта *Обхождане на масив arr с n елемента*, а в инварианта от Таблица 5 – *Размяна на стойностите на променливи x и y*. Така не се получава нов вид инвариант, защото не трябва да се правим избор на инвариант, т.е. инвариантът не е параметър.

Заклучение

Статията показва подход за преподаване на програмирането на алгоритмичен език, независимо от конкретния език, на базата на съвкупност от шаблони, наречени инварианти. В работата се предложени 142 инварианти на алгоритми от дисциплините „Програмиране” и „Алгоритми и структури от данни”. Подробно са разгледани параметрите на инвариантите. Работата представя първата стъпка в реализирането на софтуерна система за обучение по програмиране на базата на инварианти.

Работата е осъществена и финансирана по проект BG051PO001-4.3.04-0064 „Пловдивски електронен университет: национален еталон за провеждане на качествено електронно обучение във висшето обучение” на Оперативна програма "Развитие на човешките ресурси" на ЕС.

Литература

1. Бърнев П., Г. Тотков, Вл. Шкуртов, Р. Донева, К. Гъргов, Информатика, учебник за 9-ти клас, изд. Летера, 2001.
2. Бърнев П., Г. Тотков, Вл. Шкуртов, Р. Донева, К. Гъргов, Информатика+, учебник за профилирана подготовка в 9-ти клас, изд. Летера, 2001.
3. Донева Р., С. Гафтанджиева, Инварианти в обучението по електронни таблици, Сборник с доклади на националната конференция "Образованието в информационното общество", Пловдив, 2011, с. 293-302.
4. Донева Р., С. Гафтанджиева, Обучението по информационни технологии в бакалаварски програми за непрофесионалисти по информационни технологии, Научна сесия "Дни на науката 2010" на СУ - Пловдив, Пловдив, 2010, с. 39-42.
5. Тотков Г., Вл. Шкуртов, Р. Донева, Б. Видолов, Информатика (учебно помагало), РЦДО, Пловдив, 1999.
6. Тотков Г., Вл. Шкуртов, Р. Донева, К. Гъргов, Информационни технологии, учебник за 9-ти клас, изд. Летера, 2001.
7. Тотков Г., Вл. Шкуртов, Р. Донева, К. Гъргов, Информационни технологии, учебник за 10-ти клас, изд. Летера, 2001.
8. Тотков Г., Вл. Шкуртов, Р. Донева, Основи на компютърната информатика, Университетско издателство, ПУ, Пловдив, 2001.
9. Тотков Г., Р. Донева, Л. Бесалева, И. Чакърва, Инварианти в обучението по информационни технологии, Сборник с доклади на националната конференция "Образованието в информационното общество", Пловдив, 2010, с. 22-29.
10. Totkov G. et al., Computer Science: Overview, (G. Totkov ed.), PHARE, 1999.
11. Totkov G. et al., Course Guide 'RAPIDITY', (G. Totkov ed.), PHARE, 1999.