

## ПАСКАЛ ИЛИ СИ КАТО ПЪРВИ ЕЗИК ЗА ПРОГРАМИРАНЕ\*

Маргарита Радоева Бърнева, Теодоси Кирилов Теодосиев

Предлаганата работа представя гледната точка на авторите за обучението по програмиране. Сравняват се два от най-популярните езици за програмиране (*Паскал* и *Си*) в контекста на първи език за програмиране. Представени са предимствата и недостатъците на двата езика по отношение на обучението по програмиране (цели и особености) и качествата на обучаемите. Те служат за изразяване на две различни философии на програмиране и имат съществени различия. Направен е сравнителен анализ по няколко критерия: типове данни, изрази, оператори, структура на програмите и диагностика.

**1. Увод.** В последните години във връзка с широкото разпространение на езиците *Си* и *C++* за системно програмиране възниква спор: целесъобразно ли е първият език за програмиране за обучение да е *Паскал*, не е ли по-добре да се изучава езика *Си*. Това е поводът за настоящия материал. В него се излагат някои мисли и разсъждения на авторите, които имат дългогодишен опит в преподаването и воденето на практически занятия по дисциплината.

Целта на курса по Програмиране е да се развие алгоритмичното мислене на обучаемите, паралелно с усвояване на формален език за програмиране (ЕП). Сама по себе си тази задача е твърде сложна от една страна и много важна от друга, поради което очевидно ЕП трябва да е по-лек за обучаемите. Трудностите са свързани с дуалността в гледната точка към ЕП: от една страна като инструментално средство за формално описание на алгоритмите, а от друга като обект за изучаване. Задачата на преподавателя е да научат обучаемите на две нови дейности – алгоритмизиране и програмиране. Естествено е, че в преследването на тези цели успехите (освен заинтересоваността и интелектуалните възможности на обучаемите) до голяма степен зависят и от много други фактори – материалната база и възможността обучаемите да се упражняват самостоятелно, наличието на добри учебни помагала, компетентност и методически опит на преподавателите и не на последно място – езикът за програмиране и реализационната среда.

**2. Някои особености, свързани с обучението по програмиране.** Това, което цени начинаещият програмист в ЕП – достъпност, яснота и гаранции от грешки

---

\*Работата е финансирана по договор №14/ 4.06.01

– на системния програмист му се струва излишно и даже го дразни. Ще споменем някои от спецификите на програмирането на етапа на обучението:

◇ На обучаемите им се налага да четат чужди програми – може би повече, отколкото да съставят самостоятелно програми. Особено необходима е читаемостта в обучението поради това, че обучаемите трябва да разчитат програми, написани от преподавателя или автора на книгата, която ползват.

◇ Програмите, които съставят обучаемите са с малки или средни размери.

◇ От съществено значение е диагностиката на откритите грешки по отношение на близостта до истинската грешка, както по място, така и по вид.

◇ От значение е простотата на езиковите средства, тяхната яснота и еднозначност, липса на необходимост от тълкуване.

**3. Сравнение на Си и Паскал.** Общоприета е гледната точка, че основна цел на ЕП се явява възможността за програмиста да формулира своите мисли в термините на абстракции, съответстващи на решаваната задача, а не в термините на възможностите на апаратните средства [4].

И *Паскал* и *Си* са процедурни езици от високо ниво и имат сходни управляващи конструкции и типове данни. В същото време те служат за изразяване на две различни философии на програмиране и имат съществени различия. Ще се спрем на тези, които по наше мнение са съществени.

**3.1. Сравнение по отношение на типовете.** Съгласно прието определение [3], *силно типизиран* ЕП е този, в който:

- 1) Всеки обект има уникален тип;
- 2) Всеки тип определя множества от стойности и операции;
- 3) Във всяка операция за присвояване типа на присвояваната стойност и типа на обекта данни, на който се присвоява трябва да бъдат еквивалентни;
- 4) Всяка прилагана към обекта операция трябва да принадлежи на множеството операции на типа на обекта.

Експериментите показват, че ЕП със силна типизация способстват за увеличаване на яснотата и надеждността на програмите.

*Паскал* почти съответства на това определение. “Почти” заради записите с варианти и предаването в качеството на параметри на процедури и функции.

При все, че в *Си* има развити типове данни, той е далеч от това определение. Той е разработен, за да предоставя възможност на програмиста активно да използва механизма на трактовка на стойностите като данни от различен тип.

◇ Езикът Паскал има *по-богат набор от типове*.

◇ В *Си* няма логически тип, но има логически изрази и логически операции. По-точно всеки целочислен израз може да се интерпретира като логически израз. Смесването на булев и цял тип в *Си* води до невъзможност за транслятора да открие много грешки, лесно откриваеми в *Паскал*. Например:

- 1)  $1 < 2 < 3$  и  $3 < 2 < 1$  имат стойност 1, т.е. истина.
- 2) Ако  $I$  и  $J$  са цели променливи и в булев израз запишем вместо  $I < J \rightarrow I + J$  тази грешка не се открива [3].

◇ Един сериозен източник на трудности за начинаещия програмист са указателите. Те представляват сами по себе си разновидност на косвената адресация,

която изисква по-дълбоко познаване на архитектурата на компютрите и са потенциален източник на грешки – използването на неправилна адресация може да има катастрофални последици за програмата.

Ограниченията в *Паскал* правят използването на указателите от една страна по-безопасно, но от друга ползата от тях се ограничава. Отделянето и освобождаването на памет (чрез `new` и `dispose`) е част от езика *Паскал*, за разлика от *Cи*. За отделянето на необходимата памет за разполагане на някакъв обект се грижи транслятора. При *Cи* отговорността за тази дейност се възлага на програмиста, който е длъжен явно да укаже размера на отделяната памет, а това е допълнителен потенциален източник на грешки.

◇ Различия има в начина за *преобразуване на типовете* и свързания с това *контрол при присвояванията*.

Различието в трактовката на типовете данни е следствие от различните цели, които са били поставени при създаването на езиците. В *Паскал* се прави опит да се скрие от програмиста реалната машина, докато в *Cи* за това не се полагат практически никакви усилия. Напротив *Cи* предоставя на опитния програмист удобни средства за взаимодействие с реалната машина и нейното управление.

**3.2. Сравнения по отношение на изразите.** В *Паскал* операциите не са голям брой – обичайните операции за сравнение, аритметични операции, допълнени с операции за цяло частно и остатък от деление на цели, четири логически операции и операции над множества и над битови редици (във версии на езика *Паскал*). Приоритетите на операциите са четири – на унарните, на мултипликативните, адитивните и на операциите за сравнение. Вследствие само на четирите нива на приоритети в *Паскал* се налага използването на скоби (един вид – кръгли), дори когато математическата нотация не го изисква. Използват се само един вид скоби, които превръщат един подизраз в аргумент. Включени са и редица стандартни функции, които участват в изразите (обикновено като едноаргументни операции).

В *Cи* операцията има обобщен смисъл – присвояването също се разглежда като операция. Има над 30 операции, групирани в 11 приоритетни групи. Тези приоритети трябва или да се запомнят или да се ползува справочник, но и тук за по-добра читаемост се налага използването на скоби. Възможностите за преработка на данните се допълват с библиотечни функции.

Използуване на оператор за присвояване в изрази на *Cи* подтиква към използване на странични ефекти, което води до загуба на надеждност, неяснота при четене и трудно модифициране (например:  $a + (b = 5) - 1$ ). От друга страна програмите стават по-кратки.

**3.3. Диагностика.** Диагностицираните от транслятора синтактични грешки представляват първата бариера по пътя на грешките, допускани от програмиста. Типовете данни играят ключова роля в автоматизацията на диагностиката на грешките и обезпечават съществено подобрение на читаемостта на програмите [2]. В *Паскал* локализирането на синтактичните грешки и извеждане на диагностични съобщения е реализирано по-удачно отколкото в *Cи*. Много рядко една грешка предизвиква извеждане на две съобщения, докато при *Cи* е възможно (и често се случва) да се укаже цял екран от “индикации” за грешки. Наборът от проверки на етапа на изпълнение, обаче, е непълен.

**4. Заключение.** *Паскал* е сравнително прост език, който позволява систематично и точно да се изразят концепциите на структурното програмиране. Той притежава развити средства за описание на сложни типове данни, позволяващи моделиране на информационни структури от всяка предметна област и води до по-добра читаемост на програмите. Поради силната типизация, програмите на *Паскал* се отличават с по-висока надеждност отколкото на *Си*. Има развити средства за диагностика и по тази причина е удобен за обучение. Принуждавайки програмиста да следва ограниченията в *Паскал*, му се оказва съществена помощ при диагностиката на грешки. За *Паскал* има много и то качествени учебници и ръководства от български автори, което не е без значение за обучаемите.

*Си* е мощен език с много широко приложение. Поради своята гъвкавост и отсъствието на ограничения, *Си* дава възможност за използване в по-широк спектър от области. Той представлява удачен компромис между желанието да разполагаме с тези възможности, предоставяни от по-понятните и удобни ЕП и стремежа за ефективно използване на особеностите на хардуера. Този ЕП е важен за обучаемите от професионална и мирогледна точка. Особеностите на *Си*, които го определят като достоен заместник на асемблерните езици – указатели, операции с указатели, операции с отделни битове, представляват повишена трудност за начинаещия програмист.

Ще завършим с два цитата от книги за *Си*, които, хвалейки качествата на *Си*, потвърждават тезата, че *Паскал* е по-подходящ като първи език за програмиране.

“Е много програмисти, които започваха ‘любовен роман’ с *Паскал* завършваха с ‘частлив брак’ със *Си*” [1].

“Сравняването на *Си* и *Паскал* е все едно да се сравняват самолетите ‘Лир Джет’ и ‘Пайпер Каб’\* , поради това че *Си* е предназначен за създаване на програмно осигуряване, а *Паскал* за обучение по програмиране. . .” [4].

#### ЛИТЕРАТУРА

- [1] Л. ХЭНКОК, М. КРИГЕР Введение в программирование на языке Си. Москва, Радио и связь, 1986.
- [2] А. ФЬЮЕР, Н. ДЖЕХАНИ. Языки программирования Ада, Си, Паскаль. Москва, Радио и связь, 1989.
- [3] А. FEUER, N. GENANI. A comparison of the programming languages C and Pascal. ACM Computing Surveys, March 1982.
- [4] N. WIRTH. Programming Languages: What to Demand and How to Assess Them. Software Engineering, ed. by R. H. Perrott, Academic Press, New York, 1977.

М. Бърнева, Т. Теодосиев  
Шуменски Университет “Епископ К. Преславски”  
кафедра Информатика  
Шумен  
e-mail: t.teodosiev@fmi.shu-bg.net

---

\*“Лир Джет” е комфортен реактивен самолет – символ на благополучие и преуспяване. “Пайпер Каб” е малък учебен самолет. За всеки американец се асоциира с първия самолет за тези, които се учат да пилотират.

## PASCAL OR C AS FIRST LANGUAGE FOR PROGRAMMING

M. Barneva, T. Theodosiev

The following work presents the point of view of authors for education in Programming in University. It compares two programming languages (Pascal and C) in context of first programming language. We consider advantages and disadvantages of two languages in the sense of education of programming (goals and features), training of Informatics and quality of first year's students. It is done comparative analysis with some criteria: data types, expressions and operators, program structure and diagnostics.

**Key words:** Programming, Education, Pascal, C, and Compare.