

УДК 004.432

ВЫБОР ЯЗЫКА ДЛЯ ИЗУЧЕНИЯ БАЗОВЫХ ПРИНЦИПОВ ПРОГРАММИРОВАНИЯ. СОВРЕМЕННЫЙ PASCALABC.NET

Пылькин А.Н.,

Соколова Ю.С.,

Дмитриева Т.А.

*ФГБОУ ВО «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина»,*

г. Рязань, Российская Федерация

E-mail: dmitrieva.tatiana.al@gmail.com

Аннотация. Рассматривается ряд вопросов, связанных с использованием PascalABC.NET при обучении программированию, а также языковые средства современной версии языка Pascal, позволяющие писать короткий, понятный и эффективный код.

Ключевые слова: концепции программирования, языки программирования, PascalABC.NET, Python, обучение программированию.

CHOOSING A LANGUAGE FOR LEARNING THE BASIC PRINCIPLES OF PROGRAMMING. MODERN PASCAL ABC.NET

Alexander N. Pylkin,

Yuliya S. Sokolova,

Tatiana A. Dmitrieva,

Ryazan State Radio Engineering University named after Utkin.

Ryazan, Russian Federation

E-mail: dmitrieva.tatiana.al@gmail.com

Abstract. A number of issues related to the choice of a modern programming language for teaching are considered, as well as the language facilities of the modern

version of the Pascal PascalABC.NET language, which allow you to write short, understandable and effective code.

Keywords: programming concepts, programming languages, PascalABC.NET, Python, teaching programming.

В последние годы интерес к IT-индустрии продолжает расти, а перед новичками и преподавателями возникает актуальный вопрос, какой язык программирования стоит изучить для успешного старта. Выбор усложняется тем, что рейтинги популярности языков программирования ежегодно меняются [1], а самих языков очень много, и каждый из них имеет свои особенности, сферы применения и решает свои задачи. Выбор конкретного языка в профессиональной сфере зависит от цели деятельности. Например, в качестве серверных языков популярны PHP, Java, Ruby. Из клиентских языков наиболее перспективен Java Script. Для разработки сложных высоконагруженных проектов и игровых приложений требуется C++. Лидерами в разработке приложений для мобильных устройств считаются Java, Objective-C, Swift. Также лидирующее положение занимают C# и Python, которые используются для разработки программ в сфере бизнеса, веб-сервисов, мобильных приложений, игр. Любой опытный программист знает несколько языков, но, как правило, программный код на работе он пишет на одном.

Долгое время базовым для обучения программированию в старших классах и на первых курсах вузов считался разработанный Н. Виртом язык Turbo Pascal, который впоследствии заменил PascalABC.NET, поскольку он готовил хороший фундамент для изучения языков, используемых в промышленной разработке. На Pascal и C++ начинало программировать большинство разработчиков.

Однако в последние годы фаворитом для изучения основ программирования является Python, который славится своей простотой и лаконичностью. Немногословный и понятный синтаксис, похожий на псевдокод и не усложненный громоздкими конструкциями, а также сильная динамическая

типизация способствуют быстрому и безболезненному обучению новичков основам программирования. Кроме того, этот язык постоянно развивается, а сообщество вокруг Python – одно из самых сильных в мире IT.

Вместе с Python развиваются и совершенствуются другие языки программирования, включая PascalABC.NET [2, 3].

PascalABC.NET был создан около 15 лет назад в Южном федеральном университете [3]. Разработчики преследовали цель создания хорошего современного языка программирования для обучения школьников с мощной, хорошо русифицированной средой программирования, поддерживающей технологию IntelliSense, содержащей средства автоформатирования, встроенный отладчик и встроенный дизайнер форм, включающего в себя лучшее, что предлагают другие современные языки, такие как C#, Kotlin, Python, Haskell и другие. В качестве платформы реализации была выбрана платформа Microsoft.NET.

Синтаксически PascalABC.NET близок к языку Delphi, однако включает множество конструкций языка C# и содержит все современные языковые средства: классы, перегрузку операций, интерфейсы, обработку исключений, обобщенные классы и подпрограммы, сборку мусора, лямбда-выражения, средства параллельного программирования [3, 4]. Кроме того, он содержит собственные расширения языка Pascal, нацеленные на написание компактных, интуитивно понятных программ с использованием современных языковых возможностей.

Многое язык PascalABC.NET взял от языка Python. Это прежде всего легковесные языковые и библиотечные средства, обеспечивающие компактность и интуитивную понятность программ. Кроме того, часто код решения задачи, написанный на Python, фактически идентичен коду на PascalABC.NET.

Рассмотрим решение задачи (№ 14 из ЕГЭ по информатике) поиска количества повторений цифры 0 в девятеричной записи числа $N = 729^{77} + 13^{16} - 18$. В обоих случаях будет получено число 214.

Код на PascalABC.NET	Код на Python
<pre>## var N:=729bi**77+13bi**16- 18; var k := 0; while N > 0 do begin if N mod 9 = 0 then k += 1; N := N div 9; end; print(k)</pre>	<pre>N = 729**77+13**16-18 k = 0 while N > 0: if N % 9 == 0: k += 1 N = N // 9 print(k)</pre>

В рассмотренном примере для хранения значения N используется неограниченный целый тип, обозначаемый в PascalABC.NET идентификатором `BigInteger`.

Ещё один пример, демонстрирующий схожесть синтаксических конструкций этих языков программирования, также взят из сборников для подготовки к ЕГЭ по информатике (задача № 12): на вход программы для исполнителя «Редактор» поступила строка из 52 подряд идущих комбинаций «AB» (все буквы заглавные, латинские), требуется определить, какая строка получится после выполнения следующего алгоритма (в результате должна получиться строка 'BA'):

```
ПОКА Нашлось (AA) ИЛИ Нашлось (BB) ИЛИ Нашлось (AB)
    Заменить (AA, B)
    Заменить (BB, A)
    Заменить (AB, BA)
КОНЕЦ ПОКА
```

Код на PascalABC.NET

```
##  
var s := 'AB' * 52;  
while ('AA' in s) or ('BB' in s) or  
( 'AB' in s) do  
  begin  
    s := s.replace('AA', 'B', 1);  
    s := s.replace('BB', 'A', 1);  
    s := s.replace('AB', 'BA', 1);  
  end;  
print(s)
```

Код на Python

```
s = 'AB' * 52  
while ('AA' in s) or ('BB' in s) or  
( 'AB' in s):  
  s = s.replace('AA', 'B', 1)  
  s = s.replace('BB', 'A', 1)  
  s = s.replace('AB', 'BA', 1)  
print(s)
```

PascalABC.NET, аналогично Python, позволяет использовать в программах идентификаторы, содержащие кириллические символы [3]. Так, в предыдущем примере, реализовав подпрограммы Нашлось() и Заменить(), можно было оставить представленный исполнителю «Редактор» алгоритм фактически без изменений:

```
##  
var s: string;  
var Нашлось: string → boolean := v → s.Contains(v);  
var Заменить: (string, string) → () := (v, w) →  
  begin s := s.Replace(v, w, 1) end;  
s := 'AB' * 52;  
while Нашлось('AA') or Нашлось('BB') or  
  Нашлось('AB') do  
  begin
```

```
    Заменить ('AA', 'B');  
    Заменить ('BB', 'A');  
    Заменить ('AB', 'BA');  
end;  
s.Print
```

Как можно заметить из примеров, в коде на PascalABC.NET нет привычных операторов `begin ... end`, разработчики заменили их на символы `##`. Кроме того, использование сочетания символов `###` включает в PascalABC.NET режим спортивного программирования, в котором операторы записываются в краткой форме: вместо `Print` можно писать `Pr`, вместо `Where` – `Wh`, вместо `Count` – `Cnt` и т. п.

Также из последнего листинга видно, что PascalABC.NET работает с кортежами, позволяет писать программы в функциональном стиле с использованием лямбда-выражений, которое легко распознается в тексте программы по характерному знаку операции `→` (произносится «переходит в ... »).

Использование кортежа в качестве выходных параметров у функции позволяет передавать в основную программу сразу несколько значений, например, с помощью функции можно сразу определить сумму и произведение цифр в десятичном числе (пример не является учебным, он демонстрирует работу с кортежами):

```
##  
function SumProduct(n:integer): (integer, integer);  
begin  
    var (s, p) := (0, 1);  
    while n > 0 do  
        begin  
            var d := n mod 10;  
            (s, p, n) := (s + d, p * d, n div 10);  
        end;  
    result := (s, p)
```

```
end;
```

```
var a := ReadInteger('Введите число: ');  
var (s, p) := SumProduct(a);  
s.Println;  
p.Println
```

Модуль `School`, встроенный в современный `PascalABC.NET`, содержит реализацию алгоритмов, часто встречающихся в школьных задачах. Его использование делает текст программы яснее и короче, особенно для тех, кому не требуется углубляться в математические и технические детали этих алгоритмов. Так, с использованием модуля `School` и лямбда-выражений программа нахождения всех простых чисел из диапазона от 0 до 100 будет занимать три строки и иметь следующий синтаксис:

```
##  
uses school;  
(1..100).Where(x → x.isPrime).Print
```

В `PascalABC.NET` программное решение задачи нахождения суммы нечётных положительных чисел, ввод которых заканчивается точкой, может быть записано одной строкой:

```
##  
ReadSeqIntegerWhile(x → x<>0).Where(x → x.isOdd and (x >  
0)).Sum.Print
```

Из приведенных примеров видно, что реализации некоторых функций имеют два формата – для вызова в виде функции и для записи в точечной нотации. Например, для вывода значения переменной `s` можно писать `Println(s)` или `s.Println`.

Но, несмотря на наличие мощной современной системы программирования на базе языка `Pascal`, в учебных заведениях на компьютерах с установленным `PascalABC.NET` продолжают писать код с синтаксисом для `Turbo Pascal` и преподавать устаревшие `Free Pascal` или `Delphi`. Многие не

видят разницы между этими языками, полагая, что она минимальна или вообще отсутствует.

Следующий пример демонстрирует различие в кодах, написанных на Turbo Pascal, на PascalABC.NET и на Python, при решении задачи поиска максимума и его позиции в случайно сгенерированном массиве из 10 чисел, каждое из которых лежит в диапазоне от -100 до 100 (если максимумов несколько, определить номер первого).

Код на Turbo Pascal

```
var
  a: array[1..10] of integer;
  max, imax, i: integer;
begin
  randomize;
  for i:=1 to 10 do a[i]:=random(101)-
random(101);
  for i:=1 to 10 do write(a[i]:4);
  writeln;
  max := a[1];
  imax := 1;
  for i:=1 to 10 do
    if a[i] > max then
      begin
        max := a[i];
        imax := i;
      end;
  writeln(max, ' ', imax)
end.
```

Код на PascalABC.NET

```
##
var a := ArrRandom(10, -100, 100);
a.Println;
a.Max.Print;
```



```
(a.IndexMax+1).Print;
```

Код на Python

```
from random import randint  
a = [randint(-100, 100) for i in  
range(10)]  
print(a)  
print(max(a))  
print(a.index(max(a)) + 1)
```

Как показывает рассмотренный пример, коды на Turbo Pascal и PascalABC.NET сильно различаются. И, несмотря на то, что в PascalABC.NET можно писать код в стиле старого Turbo Pascal, в нем заложены конструкции, присущие другим языкам программирования. Более того, он умеет понимать, форматировать и выполнять с определенными ограничениями программы, написанные на языке C# [3].

Если вспомнить про язык программирования Python [5, 6], который разрабатывался с целью повышения продуктивности самого программиста, нежели кода, который пишется, то PascalABC.NET также позволяет концентрироваться на сути алгоритма, а не на технических деталях благодаря своим встроенным библиотекам. Но, в отличие от языка Python, язык PascalABC.NET имеет статическую типизацию, что позволяет фиксировать множество ошибок на этапе компиляции и делает программу более безопасной. Кроме того, недостаток в скорости ограничивает область применения Python. Высокопроизводительное приложение на чистом Питоне написать не получится, потребуется прибегать к помощи других языков или использовать статически типизированные расширения. Поэтому, если вы планируете стать всесторонне развитым разработчиком, только знания Python недостаточно.

PascalABC.NET за пределами сферы образования практически не используется, но он имеет множество языковых возможностей, поддерживает множество стилей программирования, поскольку обобщает языковые и

библиотечные возможности таких языков, как C#, Python, Kotlin, входящих в TOP-10 рейтинга востребованных языков программирования 2021 г. [1].

Не стоит гнаться за изучением современных языков программирования, поскольку, как показывает практика, на смену одним кодам приходят другие. Гораздо важнее обучать базовым принципам программирования, а этого можно добиться, используя на начальном этапе изучения технологий программирования язык PascalABC.NET. Так, этот язык взят за основу в курсе «Основы программирования» у студентов 1 курса мехмата ЮФУ по направлению «Фундаментальная информатика и информационные технологии» [3]. Объясняется это тем, что PascalABC.NET является мультипарадигменным языком, позволяющим программировать в структурном, объектно-ориентированном и функциональном стилях, а теоретическая и практическая база, полученная при его изучении, позволяет эффективно освоить современные технологии программирования и в очень короткий срок изучать впоследствии такие языки как Python, C++, C#, Java, Haskell, Ruby и др.

PascalABC.NET является удачным решением для начального знакомства с современными технологиями программирования еще и потому, что это программное обеспечение является свободно распространяемой отечественной разработкой с интегрированным электронным задачником Programming Taskbook [7], который предназначен для обучения программированию и содержит учебные задания, охватывающие все основные разделы базового курса программирования: от скалярных типов и управляющих операторов до сложных структур данных и рекурсивных алгоритмов.

Список использованной литературы

1. Рейтинг языков программирования 2021. – Текст : электронный. – URL: <https://habr.com/ru/post/543346/> (дата обращения 01.09.2021).
2. PascalABC.NET и ЕГЭ по информатике 2021. – Текст : электронный. – URL: https://pascalabcnet.github.io/mydoc_for_EGE.html (дата обращения 05.09.2021).

3. Осипов А. В. PascalABC.NET: Введение в современное программирование / А. В. Осипов. – Ростов-на-Дону, 2019.– 572 с. – Текст : непосредственный.
 4. Осипов А. В. PascalABC.NET: выбор школьника. Часть 3. / А. В. Осипов. – Ростов-на-Дону, 2020. – 146 с. – Текст : непосредственный.
 5. Лутц Марк. Изучаем Python, том 1, 5-е изд.: Пер. с англ. / Марк Лутц. – Санкт-Петербург : ООО «Диалектика», 2019. – 832 с. – Текст : непосредственный.
 6. Лутц Марк. Изучаем Python, том 2, 5-е изд.: Пер. с англ. / Марк Лутц. – Санкт-Петербург : ООО «Диалектика», 2020. – 720 с. – Текст : непосредственный.
 7. Абрамян М. Э. Programming Taskbook. Электронный задачник по программированию / М. Э. Абрамян. – Текст : электронный. – URL: <http://www.ptaskbook.com/ru/> (дата обращения 12.09.2021).
-

Информация об авторах:

Пылькин Александр Николаевич, д-р техн. наук, профессор;

Соколова Юлия Сергеевна, канд. техн. наук, доцент;

Дмитриева Татьяна Александровна, канд. техн. наук, доцент,

ФГБОУ ВО «Рязанский государственный радиотехнический университет имени В.Ф. Уткина».

Российская Федерация, 390005 г. Рязань, ул. Гагарина, 59/1.

Alexander N. Pylkin, Dr. Tech. sciences, professor;

Yuliya S. Sokolova, PhD in Technology, assistant professor;

Tatiana A. Dmitrieva, PhD in Technology, assistant professor,

Ryazan State Radio Engineering University named after V.F. Utkin.

Russian Federation, 390005 Ryazan, Gagarina St., 59/1.

Информация о рецензенте:

Андрианов Дмитрий Евгеньевич, д-р техн. наук, доцент, заместитель директора по учебной работе, заведующий кафедрой информационных систем,

Владимирский государственный университет им. А.Г. и Н.Г. Столетовых,
Муромский институт (филиал). *SPIN-код: 7307-5910.*

Российская Федерация, 602264 Владимирская область, г. Муром, Орловская ул.,
23.

Dmitry E. Andrianov, Dr. Tech. sciences, assistant professor, Deputy Director for
Academic Affairs, Head of the Department of Information Systems, Murom Institute
(branch) FSBEI of HE «Vladimir State University named after Stoletovs». *SPIN-
code: 7307-5910.*

Russian Federation, 602264 Vladimir region, Murom, Orlovskaya St., 23.

Поступила в редакцию / Received 08/10/2021.

Принята к публикации / Accepted 12/10/2021.

Опубликована / Published 29/10/2021.