

И. В. Панова

Нижегородский государственный педагогический университет имени Козьмы Минина, Россия

А. А. Коливных

лицей № 82, Нижний Новгород, Россия

МЕТОДИЧЕСКИЕ АСПЕКТЫ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ НА ЯЗЫКЕ PYTHON В ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ

Аннотация

В статье рассматриваются особенности обучения программированию на языке Python старшекласников с учетом опыта программирования на языке Pascal, полученного учащимися в основной школе. На основе анализа содержания учебников для углубленного изучения информатики и собственного опыта преподавания информатики авторы показывают преимущества языка Python для мотивации обучающихся к его изучению и на конкретных примерах раскрывают отдельные методические аспекты обучения программированию на языке Python. В частности, используя сравнительный анализ программных кодов для реализации процедур ввода и вывода данных, условной алгоритмической конструкции, а также для описания типов данных на двух языках программирования, авторы наглядно показывают особенности разработки программ на языке Python. Более конкретные детали программирования рассматриваются на примере темы «Массивы».

Ключевые слова: язык программирования Python, обучение программированию на языке Python, методика обучения программированию, методика обучения информатике.

DOI: 10.32517/2221-1993-2020-19-6-47-50

Появление с 2015 года в КИМ ЕГЭ по информатике и ИКТ в заданиях, направленных на проверку знаний основных конструкций языка программирования и умений исполнять и анализировать алгоритмы, наряду с алгоритмическим языком, языками Basic, Pascal и C пятого языка программирования — Python заставило учителей информатики обратить на него внимание. Этому событию предшествовал ряд публикаций, раскрывающих преимущества использования языка Python в образовании с точки зрения специалиста (см., например, [11]) и учителя (см., например, [8]). Однако

и сегодня, спустя пять лет, программирование изучается в школьном курсе информатики в основном на языке Pascal, за редким исключением.

Так нужно ли менять язык, и если да, то почему именно на Python?

И насколько процесс перехода на новый язык будет сложным для учеников?

Анализируя содержание учебников для основной школы [4, 5] и углубленного изучения информатики в старшей школе [6, 7], мы можем заметить, как увеличиваются объемы программ для решения некоторых

Контактная информация

Панова Ирина Валентиновна, канд. пед. наук, доцент, доцент кафедры прикладной информатики и информационных технологий в образовании, Нижегородский государственный педагогический университет имени Козьмы Минина, Россия; *адрес:* 603005, Россия, г. Н. Новгород, ул. Ульянова, д. 1; *e-mail:* panova_iv@mininuniver.ru; 0000-0002-0648-2931

Коливных Анна Александровна, учитель информатики, лицей № 82, Нижний Новгород, Россия; *адрес:* 603003, Россия, г. Нижний Новгород, ул. Культуры, д. 1;

студентка 2-го курса магистратуры, Нижегородский государственный педагогический университет имени Козьмы Минина, Россия; *адрес:* 603005, Россия, г. Нижний Новгород, ул. Ульянова, д. 1; *e-mail:* kolivnykanya@mail.ru

I. V. Panova

Minin Nizhny Novgorod State Pedagogical University, Russia

A. A. Kolivnyk

Lyceum 82, Nizhny Novgorod, Russia

METHODOLOGICAL ASPECTS OF TEACHING PYTHON PROGRAMMING IN THE SCHOOL INFORMATICS COURSE

Abstract

The article discusses the features of teaching programming in Python of high school students, taking into account the experience of programming in the Pascal language obtained by students in the main school. Based on the analysis of the content of textbooks for in-depth study of informatics and their own experience of teaching informatics, the authors show the advantages of the Python language for motivating students to study it and use specific examples to reveal certain methodological aspects of teaching programming in the Python language. In particular, using a comparative analysis of program codes for implementing data input and output procedures, conditional algorithmic construction, as well as for describing data types in two programming languages, the authors clearly show the advantages of developing programs in Python. More specific details of programming are considered on the example of the theme "Arrays".

Keywords: Python programming language, teaching programming in Python, methods of teaching programming, methods of teaching informatics.

задач: если в восьмом и даже в девятом классе изучаются алгоритмы на простейших задачах, которые занимают в среднем 15–20 строк, то в старшей школе изучаются более сложные структуры и алгоритмы, написание которых на языке программирования Pascal может занимать до 80 строк кода. Например, любая задача для работы с массивом имеет как минимум три обязательные части: создание и ввод массива, его вывод и работа с ним.

Для сравнения объема работы на языках Pascal и Python решим следующую показательную задачу: создать массив из 20 четных элементов, начиная с двух, и найти их сумму (табл. 1).

Как видно из таблицы 1, решение данной задачи на языке Python сократится с 14 до четырех строк, а результат на экране будет один и тот же.

Объемные алгоритмы, написанные на языке программирования Pascal, можно без труда уменьшить в два, а то и в три-четыре раза, как мы это увидели на конкретном примере, если использовать возможности языка Python, а компактность и простота написания кода дает возможность изучить язык Python с гораздо более низким уровнем начальной подготовки, поскольку самая маленькая программа занимает всего одну строчку:

```
print ('это рабочая программа')
```

С методической точки зрения в предыдущей задаче лучше было бы показать, что массив является именованным набором переменных, расположенных в памяти непосредственно друг за другом, что к элементам массива можно получить доступ при помощи их индекса, продемонстрировать алгоритм для нахождения суммы элементов массива, а не использовать встроенную функцию нахождения их суммы. Поэтому как *первый* язык программирования для школьников Python, скорее всего, не подходит. Но как только учащиеся уже понимают все вышеупомянутые моменты, краткость кода является неоспоримым преимуществом. К аналогичному выводу пришли К. Ю. Поляков и И. А. Сукин в своих статьях [8, 11].

Вместе с тем можно встретить и другие точки зрения, например, что язык Python идеально подходит для изучения программирования с нуля и должен вытеснить изучаемые сегодня в школах языки Pascal и Basic [10],

и рассуждения, что выбор языка — задача второстепенная, ведь самое важное для школьного программирования — научить школьников алгоритмическому мышлению, а этого можно достичь при помощи любого языка программирования [1].

Pascal — это учебный язык программирования, который поможет учащимся разобраться в основах программирования: покажет различие между типами переменных и операциями над ними, продемонстрирует, как работает операция присваивания и в чем ее отличие от операций сравнения, как строятся сложные условия и в каких случаях мы можем использовать цикл с условием, а в каких достаточно применить цикл с заданным количеством повторений.

А изучение языка Python в старших классах обусловлено интересом к нему на более высокой ступени обучения — в российских и зарубежных университетах [14], популярностью в ИТ-индустрии, о чем свидетельствует и достаточное количество онлайн-курсов, предлагающих научить программировать на Python и новичка, и специалиста (см., например, [3, 9, 13]).

Учитывая отзывы специалистов и реальный рейтинг популярности языка Python [15], считаем целесообразным ввести этот язык программирования в школьный курс информатики в качестве *второго* языка программирования, например, в старшей школе в классах с углубленным уровнем изучения информатики после изучения языка Pascal в основной школе. В частности, именно так предлагают обучать программированию авторы УМК по информатике для VIII—XI классов К. Ю. Поляков и Е. А. Еремин, выбирая в качестве базового языка программирования в VIII—IX классах Pascal [4, 5], а уже в X—XI классах — именно язык Python [6, 7, 12].

В чем же заключается особенность перехода с языка программирования Pascal на язык Python в старшей школе?

После изучения основ программирования в основной школе учащиеся уже владеют навыками программирования линейных, условных и циклических алгоритмов, алгоритмов обработки массивов, строк, процедур и функций, поэтому в X классе можно начать с изучения этих же тем, но более углубленно с использованием языка программирования более высокого уровня

Таблица 1

Сравнение кода программ на языках Pascal и Python

Pascal	Python
<pre>var A: array [1..20] of integer; i, sum: integer; begin for i := 1 to 20 do begin A[i] := 2 * i; write (A[i], ' '); end; writeln; for i := 1 to 20 do sum := sum + A[i]; write (sum); end.</pre>	<pre>A = [2 * i for I in range (1, 21)] print (A) suma = sum (A) print (suma)</pre>

Python, далее уже продолжить его изучение в XI классе. Необходимо объяснить учащимся цель смены языка программирования (решение более трудоемких задач с меньшими затратами во времени, перспектива дальнейшего изучения языка в вузе и будущее трудоустройство) и показать все преимущества языка Python над изученным ранее языком Pascal, а также обратить внимание на особенности нового языка.

Обратим внимание школьников на некоторые отличительные черты языка программирования Python.

Самое первое, что нужно дать им понять, — это то, что Python — интерпретируемый, объектно-ориентированный язык программирования с динамической семантикой. При написании программы нет четкого объявления типов переменных: они создаются по ходу написания кода, и программа сама определяет, с переменной какого типа она работает и какие операции к ней применимы. Например, при выполнении оператора «a = 2020» в памяти создается переменная (целого типа) и происходит связь этого объекта с именем «a». Но, если мы попробуем ввести это же число с клавиатуры: «a = input ()», то введется уже в любом случае строка и для работы с ней как с числом нужно об этом сообщить программе и поменять ее тип, например, так: a = int (input ()).

Помимо целого типа int, который заменяет собой и короткое целое, и длинное целое числа, в Python встроены также еще три простейших типа данных:

- булевы значения — bool (имеют значения True или False);
- числа с плавающей точкой — float (числа с десятичной запятой, такие как 56,8544 или 19,0, а также экспоненты 1,3e5, что означает «умножить на 10 в пятой степени»);
- строки — str (любая последовательность символов).

Главную особенность *ввода* с клавиатуры мы рассмотрели, но у *вывода* на экран тоже есть свои нюансы. Всегда нужно учитывать, что если в одном операторе вывода print находится несколько элементов, то они по умолчанию выводятся через пробел и в конце ставится перенос на следующую строку:

```
print (a, 'год')
⇒2020 год
```

Если пробелы в выводе нам не нужны, мы можем указать, что будет использоваться в текущем *print* разделителем вместо них, — запись: «print ('Один', 'Два', 'Три', sep = ' ')» означает, что разделителями в данном выводе будет выступать восклицательный знак с пробелом после него, и в результате мы увидим на экране «Один! Два! Три!».

Рассмотрим более конкретные детали программирования на языке Python на примере темы «Массивы».

С понятием «массив» учащиеся впервые сталкиваются в VIII классе по программе К. Ю. Полякова, Е. А. Еремина [4] или в IX классе по программе Л. Л. Босовой [2]. Они изучают, что такое массив, как работать с его элементами, и осваивают некоторые алгоритмы работы с массивом: нахождение суммы элементов массива, количества элементов, удовлетворяющих некоторому условию, а также поиск максимального и минимального

элементов массива, сортировка массива. В IX классе по программе К. Ю. Полякова, Е. А. Еремина [5] дополнительно рассматриваются такие понятия, как «матрица» и «двумерный массив».

При переходе на Python в X классе необходимо объяснить учащимся, что понятие «массив» далее заменяется более обширным понятием «список», в котором могут находиться уже переменные различных типов, и это может стать ошибкой при неверном заполнении списка. Допустим, изначально в списке хранятся некоторые слова (строковый тип), в ходе выполнения программы ученик при добавлении элементов в массив забывает поставить апострофы (или кавычки, в данном случае это не имеет значения) и случайно заносит в список числовое значение:

```
A = ['2'] * 3
A = A + [5]
```

Программа не увидит в данном действии ничего недопустимого и при выводе на экран данного массива выполнит это действие:

```
print (A)
⇒['2', '2', '2', 5]
```

Однако при составлении предложения из элементов списка мы получим ошибку:

```
k = ''
for i in range (4):
    k = k + A[i] + ' '
print (k)
⇒TypeError: can only concatenate str
(not "int") to str
```

Получили ошибку несовпадения типов. Почему это произошло? Все просто, Pascal не допускает добавления в строковый массив переменных другого типа, он перед запуском программы проверяет весь код на наличие ошибок, при их обнаружении он сразу на это указывает и заканчивает свою работу. То есть программа не запустится, пока все ошибки не будут исправлены.

Python же работает с кодом построчно: если в данной строке нет ошибки, то он ее выполняет. Как уже было сказано ранее, в списках разрешено хранение данных разных типов, поэтому во второй строке кода нет никакой ошибки и программа без колебаний выведет нам список со смешанным типом переменных. Но операция конкатенации уже не может выполняться с числом, поэтому далее мы получим ошибку.

Здесь же необходимо отметить, что в Pascal создание массива происходило в блоке описания переменных *var*, но Python — динамический язык программирования, и данный блок просто отсутствует, поэтому программе нужно для начала сообщить, что в некоторой переменной мы хотим создать именно список, состоящий из нескольких ячеек:

```
A = [0] * 10
For i in range (10):
    A[i] = i * 2
```

Отсутствие первой строчки в данной программе приведет к ошибке:

```
NameError: name 'A' is not defined,
```

Примеры фрагментов программ с условным оператором на языках Pascal и Python

Pascal	Python
<pre> if b <> 0 then begin c := a / b; writeln ('Частное равно ', c); end else writeln ('Деление на ноль запрещено'); </pre>	<pre> if b != 0: c = a / b print ('Частное равно', c) else: print ('Деление на ноль запрещено') </pre>

что дословно переводится как «имя А не определено» и будет означать невозможность использовать переменную А в качестве списка.

Как можно было заметить, в данных примерах мы рассматриваем цикл «for i in range (n)», что означает «пока переменная i в диапазоне от 0 до n–1», используется n значений, включая 0 и исключая последнее. Нетрудно догадаться, что в Python нумерация элементов списка начинается с нуля, а в Pascal мы могли самостоятельно задавать нумерацию в блоке описания переменных и использовать собственные значения в циклах работы с массивом.

При изучении алгоритмов работы с массивом необходимо сначала повторить с учениками алгоритмы, которые им были известны ранее, и предложить им переписать их с Pascal на Python самостоятельно или при помощи учителя. Только после этого следует рассказать о существовании встроенных функций для облегчения работы программиста. Например: A.pop (n) удалит из списка элемент с индексом n, max (A) найдет значение максимального элемента массива, а A.reverse () выполнит реверс массива. Если же во время повторения алгоритмов работы с массивами рассказать о существовании встроенных функций, у учащихся может возникнуть ряд вопросов с мотивацией изучения объемного алгоритма, который можно заменить одной короткой строчкой. Например, если при изучении сортировки массива сразу рассказать про наличие функции A.sort (), а после «открыть» для них сортировку пузырьком, выбором или быструю сортировку при помощи рекурсивной функции, ученики просто не поймут необходимости изучения таких сложных понятий алгоритмов при наличии коротенькой A.sort ().

Аналогичные нюансы есть и при изучении каждой отдельно взятой темы. Например, в процессе изучения темы «Условные алгоритмы» учащиеся, скорее всего, столкнутся с проблемой «обязательности» отступов в Python, т. е. если в Pascal несколько действий в условии ограничивались операторными скобками, то в Python это выделяется в виде отступов перед строчками (табл. 2). Это может вызывать определенные трудности поначалу, особенно, если ученики ранее не стремились писать «красивый код», но в конечном итоге это дисциплинирует и учит следить за каждой написанной строкой.

В результате можно отметить, что не только нет необходимости отказываться от идеи изучения высоко-

уровневого языка программирования Python в школе, а, напротив, при учете методических особенностей это может дать преимущества как учителю, так и учащемуся — это и больший объем пройденного материала, и огромный потенциал самого языка программирования в современной ИТ-индустрии. Описанный подход к обучению позволит школе подготовить начинающих программистов, имеющих опыт использования нескольких языков программирования и умение реализовывать одну и ту же задачу различными способами.

Список использованных источников

1. Бобров А. Н. Проблемы выбора языка программирования в школьном курсе информатики // Молодой ученый. 2015. № 24 (104). С. 61–64. <https://moluch.ru/archive/104/24471/>
2. Босова Л. Л., Босова А. Ю. Информатика. 9 класс. М.: БИНОМ. Лаборатория знаний, 2017. 208 с.
3. Панова И. В., Коливын А. А. Обзор содержания онлайн курсов по обучению основам программирования на языке Python // Современные образовательные Web-технологии в реализации личностного потенциала обучающихся. Сборник статей участников Международной научно-практической конференции (20–21 мая 2020 года). Арзамас: Арзамасский филиал ННГУ, 2020. С. 523–528.
4. Поляков К. Ю., Еремин Е. А. Информатика. 8 класс. М.: БИНОМ. Лаборатория знаний, 2017. 256 с.
5. Поляков К. Ю., Еремин Е. А. Информатика. 9 класс. М.: БИНОМ. Лаборатория знаний, 2017. 288 с.
6. Поляков К. Ю., Еремин Е. А. Информатика. 10 класс. Базовый и углубленный уровни: учебник. В 2 ч. Ч. 2. М.: БИНОМ. Лаборатория знаний, 2016. 352 с.
7. Поляков К. Ю., Еремин Е. А. Информатика. 11 класс. Базовый и углубленный уровни: учебник. В 2 ч. Ч. 2. М.: БИНОМ. Лаборатория знаний, 2016. 304 с.
8. Поляков К. Ю. Язык Python глазами учителя // Информатика. 2014. № 9. С. 4–16.
9. Программирование на Python. <https://stepik.org/course/67/syllabus/>
10. Сорокина Н. А. Python как основной язык программирования в средней школе // Молодой ученый. 2019. № 5 (243). С. 15–16.
11. Сукин И. А. Python, проглатывающий слона // Информатика. 2012. № 2. С. 22–42.
12. Язык Python // Сайт Константина Полякова. <http://kpolyakov.spb.ru/school/probook/python.htm>
13. Python: основы и применение. <https://stepik.org/course/512/syllabus>
14. Schools Using Python. <https://wiki.python.org/moin/SchoolsUsingPython/>
15. TIOBE Index. <https://tiobe.com/tiobe-index/>