

Методика обучения основам программирования на языке Python. Планируемые результаты обучения программированию в соответствии с ФГОС ООО

Информатика в настоящее время – наиболее изменяющаяся в области содержания учебная дисциплина, как среди школьных дисциплин, так и среди предметов, изучаемых в вузах и образовательных учреждениях среднего профессионального образования.

Динамичное развитие информационных технологий в современном мире приводит к появлению все новых языков программирования. Отражение этого процесса мы видим и в школьном образовании, в частности, в расширении перечня языков программирования, допустимых к применению на ОГЭ и ЕГЭ по информатике и ИКТ. Вопросы повышения уровня обучения программированию в школе продолжают оставаться актуальными. Перед учителем поставлена задача, не просто предоставить выпускнику набор знаний, а подготовить учащегося, умеющего самостоятельно решать возникающие перед ним проблемы, способного отвечать за собственное благополучие и общества в целом. Тема «Алгоритмизация и программирование» является одной из самых сложных тем при изучении курса информатики.

Традиционно в школьном курсе информатики отводится место на рассмотрение базовых алгоритмических конструкций и основ программирования на одном из языков высокого уровня, который должен быть учебным и универсальным, иметь достаточно простой синтаксис и низкие аппаратно-системные требования. В традициях отечественного образования прочно закрепился язык программирования Pascal, однако в последнее время получили распространение и другие языки. В пробных вариантах заданий по теме «Алгоритмизация и программирование» Единого государственного экзамена выделяют пять языков, а именно: Basic, Pascal, C, Python, алгоритмический язык.

Если же обратиться к федеральному государственному образовательному стандарту основного общего образования, то в нем и на базовом, и на

углубленном уровне язык Python стоит среди рекомендованных языков уже на первом месте.

Также необходимо отметить, что в большинстве школ до недавнего времени в качестве языка программирования чаще всего изучался Pascal, но в последние два года тенденция существенно изменилась и все больше и больше учеников при выполнении заданий ОГЭ и ЕГЭ по информатике в качестве языка программирования используют язык Python.

Безусловно, изучение школьниками Python, в качестве первого языка программирования, может вызывать некоторые закономерные опасения, поэтому целесообразно основные алгоритмические конструкции изучить, например, на языке программирования КуМир.

Однако несомненны достоинства изучения Python в качестве основного языка программирования в школе. Программы на Python существенно лаконичнее Pascal, что существенно облегчают задачу знакомства с языком начинающим программистам, так как поиск ошибок и отладка требует существенно меньших временных затрат.

Сравним, например, два элемента кода программы, написанных на языках программирования Pascal и Python:

Python	Pascal
<code>a=[1]*1000</code>	<code>var a: array [1..1000] of integer; Begin for i:=1 to 1000 do a[i]:=1; End.</code>

Исходя из приведенного кода, можно увидеть, что на двух языках программирования записаны равнозначные операции, в результате выполнения которых мы получим массив из 1000 элементов, заполненных единицами. Однако на Python этот код занимает 1 строку, тогда как на Pascal 4.

Возможно, с методической точки зрения, при изучении данного раздела и решении приведенной выше задачи, школьнику необходимо объяснить, что массив является непрерывным фрагментом выделяемой памяти, и при его создании мы должны зарезервировать под него в памяти место, объявив его, а

затем проинициализировать. Однако строка $a=[1]*1000$, на наш взгляд, отображает смысл выполняемого школьником действия (нужен массив из числа 1 повторенного 1000 раз) полнее и, в конечном итоге, проще в написании.

Рассматривая вопросы обучения информатике в 7-9 классах на базовом уровне можно отметить, что вот эти начальные представления о программировании, возможно и есть порог, на котором ребенок остановится, получив общее представление о программировании и в частности о массивах, их объявлении и обработке. На углубленном же уровне в 7-9 классах, а уж тем более в профильных 10-х – 11-х классах, учитель и ученик получают в свое распоряжение достаточно универсальный, современный язык программирования, который реально применяется для разработки программного обеспечения.

Безусловно, большое количество высокоуровневых подпрограмм, встроенных в язык Python, широкий функционал языка, будет приводить к тому, что перед школьником возникнет соблазн использования этих возможностей, вместо реального изучения алгоритмов и принципов работы этих функциональных элементов.

Однако здесь необходимо рассмотреть методическую особенность изучения такого языка программирования, связанную с тем, что школьники должны первоначально изучить принципы и алгоритмы работы отдельного функционального элемента, и только затем перейти к его использованию при решении задач и как раз учитель должен контролировать этот процесс.

Например, решения следующей задачи: обмен значений переменных a и b .

Задача должна первоначально быть решена путем использования дополнительной переменной c :

$c=a$

$a=b$

$b=c$

Также можно с учениками рассмотреть и другой способ с использованием арифметических действий:

$a = a + b;$

$b = a - b;$

`a = a - b;`

И только после этого возможно использовать кортежи языка Python:

`(a,b)=(b,a).`

Таким образом, преподаватель может подойти к решению большинства задач, связанных с сортировкой массивов, поиском элементов, нахождении максимального и минимального элементов последовательности, что в итоге на профильном этапе обучения позволит школьнику решать большее количество разнообразных задач в сжатые сроки.

Рассмотрим основные особенности языка программирования Python.

1) Отступы – часть синтаксиса.

Многие учителя информатики хорошо знают, сколько сил и времени требуется для того, чтобы научить школьников правильно расставлять операторные скобки, выделяющие тело циклов или условных операторов. При программировании на Python этой проблемы не существует: отступы являются частью синтаксиса языка, то есть они обязательны.

В Паскале, например, можно написать такой (ошибочный) цикл:

```
i := 0;
while i < 100 do
  writeln ( i );
  i := i + 1;
```

– и это приведет к заикливанию, потому что оператор `i := i + 1` не входит в тело цикла. В Python такая ошибка в принципе невозможна, потому что все операторы, входящие в блок, должны иметь одинаковые отступы:

```
i = 0
while i < 100:
  print ( i )
  i = i + 1
```

Использование отступов делает ненужными использование операторных скобок.

2) Динамическая типизация

В языке Python используется динамическая типизация переменных. Это означает, что переменные не нужно объявлять. Тип переменной определяется автоматически, когда ей присваивается значение. Одна и та же переменная в разных частях программы может быть целым числом, затем вещественным числом, после этого – символьной строкой. Плохо или хорошо? Разработчик программы на Python получает полную свободу в обмен на ответственность за свои действия. Поэтому программы на Python требуют тщательного тестирования всех ветвей алгоритма.

3) Компактность кода

Одно из очевидных достоинств языка Python – компактность программного кода.

Рассмотрим на примере Алгоритма Евклида

Python	Pascal
<pre>while b: a, b = b, a % b</pre>	<pre>while b <> 0 do begin c := a mod b; a := b; b := c end;</pre>

Нужно отметить, что компактность кода говорит не о том, что Python лучше, чем Паскаль и С, а о том, что Python – это язык более высокого уровня. Он скрывает от программиста реализацию некоторых алгоритмов за счет встроенных средств.

4) Списки – это больше, чем массивы.

В языке Python нет массивов в привычном понимании этого термина, но зато есть списки, которые можно считать расширением понятия «динамический массив». Можно отдельно работать с каждым элементом списка, а можем выполнять операции со всем списком, например, добавлять и удалять элементы, копировать части списка, сортировать и т.п.

5) Кортежи

Кортеж – это неизменяемый список, он записывается с помощью круглых скобок

В отличие от списка нельзя добавить, изменить или удалить элемент кортежа, но можно построить новый кортеж и связать его с той же переменной.

При использовании кортежа, функция может вернуть несколько значений (объединив их в кортеж).

Например, пусть требуется написать функцию, которая находит минимальный элемент массива и его индекс:

```
def min_ind ( A ):
```

```
    m = min(A)
```

```
    ind = A.index(m)
```

```
    return (m, ind)
```

Оператор `return (m, ind)` вернет кортеж, составленный из значения минимального элемента и его индекса.

б) Символьные строки

В Python нет отдельного типа данных «символ», но есть тип «строка» (`string`). В отличие от многих других языков программирования в Python строки – это неизменяемые объекты, что в свою очередь может вызвать у учеников определенные сложности, поэтому при работе со строками, необходимо сразу же учеников приучать к тому, что если в результате работы алгоритма строка должна изменяться, то необходимо для этих целей использовать новую строку.

В ФГОС ОО четко перечислены задачи, которые должны уметь решать ученики. Рассмотрим, какие же это задачи на углубленном уровне:

8 класс (углубленный уровень):

- Нахождение минимума и максимума из двух, трёх и четырёх чисел.
- Решение квадратного уравнения, имеющего вещественные корни.
- Алгоритм Евклида для нахождения наибольшего общего делителя двух натуральных чисел.
- Разбиение записи натурального числа в позиционной системе с основанием, меньшим или равным 10, на отдельные цифры.
- Разложение натурального числа на простые сомножители.

- Алгоритм проверки натурального числа на простоту.
- Вычисление количества, суммы, среднего арифметического, минимального и максимального значений элементов последовательности, удовлетворяющих заданному условию.

- Посимвольная обработка строк. Подсчёт частоты появления символа в строке.

9 класс (углубленный уровень)

- Вспомогательные алгоритмы (подпрограммы, процедуры, функции).
- Рекурсия.
- Сортировка массивов.
- Двоичный поиск в упорядоченном массиве.
- Основные алгоритмы обработки двумерных массивов (матриц):
заполнение двумерного массива случайными числами и с использованием формул; вычисление суммы элементов, минимума и максимума строки, столбца, диапазона; поиск заданного значения.

- Динамическое программирование. Задачи, решаемые с помощью динамического программирования: вычисление функций, заданных рекуррентной формулой; подсчёт количества вариантов, выбор оптимального решения.

В заключении отметим основные достоинства языка программирования Python с точки зрения обучения школьников программированию:

- низкий «порог входа»; простейшая программа на Python в отличие от Паскаля и С занимает одну строчку.

- понятный синтаксис, отступы как часть синтаксиса языка;
- Python – это язык более высокого уровня, чем С и Паскаль, позволяющий решать задачу на более высоком уровне абстракции;

- развитые структуры данных: списки, словари, множества;
- компактность программ (достигается за счет встроенных средств);
- Python широко применяется в профессиональных разработках, то есть не является чисто учебным языком без перспектив применения в реальной жизни;

- большая библиотека (встроенные модули и разработки сообщества);

- возможность разработки программ с графическим интерфейсом;

- Python поддерживает различные подходы к программированию (императивный, объектно-ориентированный, функциональный).

Достоинств у Python много, он хорош для профессиональных программистов, но у учеников мы, в первую очередь, должны развивать алгоритмическое мышление, поэтому ученики должны научиться составлять основные алгоритмы без использования встроенных функций и только после того, как эти алгоритмы будут ими усвоены, можно переходить и к использованию встроенных функций.

На наш взгляд преподавание языка Python в современной школе, целесообразно и является перспективным направлением в информатике, он является достаточно простым даже для начинающих. Вместе с тем он обладает рядом особенностей, требующих пристального изучения и внимания.