Некоторые особенности изучения языка программирования Python

Г. В. Гаркавенко, email: g.garkavenko@mail.ru
И. Д. Шеметов, email: itvtnjd2003@gmail.com

Воронежский государственный педагогический университет

Аннотация. *В данной работе рассматриваются некоторые особенности программирования на языке Python, на которые надо обратить внимание даже при написании простейшей программы. В статье приводятся два примера и подробно рассматриваются особенности Python в сравнении с таким же решением на C++.*

Ключевые слова: *Python, изучение программирования, типы данных, ввод данных в Python.*

Введение

При изучении информатики в школьном курсе информатики отводится время на изучения основ какого-либо языка программирования [1]. В последнее время все большую популярность приобретает знакомство с языком Python, тем более, что его разрешено использовать при решении заданий ЕГЭ по информатике и при участии в олимпиадах по программированию [2]. Этот язык позиционируется как простой язык для изучения программирования. Но те, кто впервые знакомятся с написанием программ не всегда понимают, что надо не просто писать команды, выполняющие те или иные действия, но и понимать, как данные хранятся и обрабатываются, почему возникают ошибки, когда код написан синтаксически правильно [3]. Некоторые примеры решения задач с использованием Python, имеются в [4].

Рассмотрим некоторые особенности языка Python, которые скрываются за кодом самой простой программы на Python.

1. История возникновения и философия языка Python

Впервые язык программирования Python появился в 1991 году. Его разработчиком и создателем философии нового языка программирования стал Гвидо Ван Россум. В 1995 году появилось некоммерческое Python-сообщество, которое стало отвечать за защиту и развитие языка Python. Таким образом, в октябре 2000 года вышел Python 2.0, положивший начало линейке версий языка Python 2.x. А в декабре 2008 года появился Python 3.0, который не совместим с линейкой Python 2. И, хотя некоторые компании еще используют Python 2, все больше компаний переходят на Python 3.

Создавая свой язык программирования Гвидо Ван Россум взял некоторые идеи, которые посчитал перспективными, в таких языках программирования как ABC и Модула-3. А как известно, создателем пророделя языка Модула-3, является Никлаус Вирт, разработавший язык программирования Модула в 1975 году, и являющийся автором не безызвестного языка Паскаль.

Тим Питерс, один из членов, разработчиков Python-сообщества, написал некоторые высказывания в виде «The Zen of Python», которые можно назвать философией Питона, это – список правил, на которых основывается данный язык программирования. Посмотреть этот список можно при помощи команды «import this». Одним из правил, входящих в этот список является «простое лучше сложного»., наверное, это и привлекает школьников, которые хотят научится программировать.

По данным Википедии существует более 700 языков программирования, которые поддерживают те или иные парадигмы программирования. Большинство современных языков программирования поддерживают несколько парадигм программирования, но Python поддерживает почти все парадигмы, в том числе, наравне со структурным и объектно-ориентированным, он поддерживает парадигму функционального программирования.

В рейтинге востребованности языков программирования Python уже много лет входит в пятерку лучших.

2. О средах программирования на языке Python

Python относится к языкам высокого уровня, и аналогично Бейсику является языком-интерпретатором, то есть строки кода программы сразу интерпретируются без компиляции. Для программирования на языке Python нужно установить его на компьютер и использовать интегрированную среду разработки IDLE, которая устанавливается вместе с языком программирования. Но для разработки больших проектов она не очень удобна. При изучении программирования можно использовать более удобную среду разработки Wing 101, для профессиональной разработки часто используют среду PyCharm, или Jupyter Notebook из пакета Anaconda. Впрочем, можно не устанавливать Python на компьютер, а использовать какую-нибудь онлайн среду разработки или используя google-аккаунт программировать в Google Colab, в основе которого Jupyter Notebook, но только в онлайн-исполнении, и написанные программы будут сохранятся на google-диск в вашем аккаунте.

Сферы использования программирования на Python самые разнообразные. Но в большей степени инструменты языка предназначены для анализа и обработки данных, машинного обучения,

поэтому чаще всего используются в таких областях, как Data Scientist и Machine Learning. С его помощью были написаны такие программы как Blender, Dropbox, WikidPad и другие.

В отличие от других языков, в Python синтаксис сделан более лаконичным, благодаря чему для написания программ используется меньше кода, он проще читается и новичку достаточно просто понять и изучить его. Но несмотря на то, что количество кода меньше, программы, написанные на Python уступают в скорости работы и не подходит для задач, требующих большой объем памяти.

3. Особенности языка, примеры

Изучение любого языка программирования начинается с изучения простых типов данных и операций над ними. Уже на этом этапе мы встречаемся с особенностями языка Python. Во всех изучаемых в школе и вузе языках программирования сначала описывается тип переменной, а затем ей присваивается соответствующее значение или выражение. При выполнении кода программы, встречая задание типа, в памяти компьютера под переменную резервируется соответствующее заданному типу количество байт и туда впоследствии записывается значение переменной. Проведем некоторое исследование [5], рассмотрев пример.

Пример 1. На языке программирования C++ написан код в котором объявляются две целочисленных переменных и им присваиваются одинаковые значения. Затем печатаются значения этих переменных, адреса памяти, отведенные под хранение этих переменных и количество байт, отводимое под хранение переменной заданного типа

```
#include <iostream>
using namespace std;
int main()
{
    int a=3;
    int b=3;
    cout<<a<<" address:"<<&a<<" byte "<<sizeof(a)<<endl;
    cout<<b<<" address:"<<&b<<" byte "<<sizeof(a)<<endl;
    a=a+2;
    cout<<a<<" address:"<<&a<<" byte "<<sizeof(a)<<endl;
    return 0;
}
```

Программа выдаст следующее:

3 address:0x61ff0c byte 4
3 address:0x61ff08 byte 4
5 address:0x61ff0c byte 4

Мы видим, что значения у переменных одинаковые, а адреса в памяти разные, под каждую переменную отведен 4 байта памяти. При

изменении переменной *a* на 2, изменяется значение переменной, но адрес хранения неизменен, он отведен именно переменной.

Рассмотрим аналогичный код на языке Python:

```
a=3
b=3
print(a," address:",id(a))
print(b," address:",id(b))
a=a+2
print(a," address:",id(a))
```

Результат работы программы:

3 address: 140724136092112
3 address: 140724136092112
5 address: 140724136092176

Обратим внимания, что у разных переменных, имеющих одинаковые значение совпадают адреса в памяти компьютера, а при увеличении значения *a* на 2, изменяется не только значение, но и адрес хранения. Кроме того, мы не можем так просто как в C++ определить сколько байт выделяется под хранение числа.

В Python все числовые типы являются неизменяемыми. При встрече в коде программы числа, например, 3, создается объект с этим значением, идентификаторы *a* и *b* являются даже не переменными, а чем-то вроде «ярлычков», цепляемых к созданным объектам. При необходимости «ярлычок» переклеивают на другой объект [6].

Именно на такие вещи надо обращать внимание обучающихся, важно, чтобы человек, изучающий программирование понимал, как данные, используемые в программах хранятся в памяти компьютера.

Еще одной особенностью программирования на Python, на которую надо обратить внимание обучающихся, является то, как записывается текст программы. В большинстве языков программирования оператор оканчивается знаком ‘;’, а блок операторов заключается в так называемые операторные скобки, в Pascal begin… end, в C,C++, Java скобки {…}. В языке Python принято писать каждый оператор на отдельной строке, это связано с тем, что данный язык является интерпретируемым. Хотя точку с запятой можно поставить после оператора и ошибки не будет, этот знак используется для разделения нескольких операторов, записанных на одной строке. Читаемость и понятность такого кода ухудшатся. А в качестве операторных скобок в Python, используются отступы, и по стандарту это должно быть 4 пробела.

Далее, хотелось бы обратить внимание на ввод и печать данных, это еще одна вещь, с которой знакомят обучающихся в первую очередь. Рассмотрим простейший пример.

Пример 2. Вводятся три целых числа, высота, длина и ширина упаковочной коробки. Вычислить объем этой коробки.

На языке C++ программный код, реализующий данную задачу будет выглядеть так:

```
#include <iostream>
using namespace std;
int main()
{
    int a, b, c, V;
    cout<<"Enter 3 number:";
    cin>>a>>b>>c;
    V=a*b*c;
    cout<<"V="<<V;
    return 0;
}
```

Причем при вводе трех чисел их можно записать в одну строку, можно вводить по одному в строке, можно два числа на одной строке, а третье на другой. Все способы будут работать одинаково.

А вот на Python так не получится. Надо заранее предусмотреть то, как, в строку или столбец, будут вводится данные. Оператор ввода в данном языке всегда считывает строковый тип данных, поэтому надо еще предусмотреть преобразование прочитанной строки для нашего примера в целочисленный тип. Приведем пример программы, которая аналогична программе на C++, приведенной выше, но вводить значения можно только в столбец, то есть по одному на каждой строке.

```
print("Enter 3 number:")
a=int(input())
b=int(input())
c=int(input())
V=a*b*c
print("V=", V)
```

Причем для печати запроса на ввод трех чисел первые две строки можно заменить строкой

```
a=int(input("Enter 3 number:"))
```

Если ввод осуществляется в одну строку, то код на Python будет выглядеть следующим образом:

```
a,b,c=map(int, input("Enter 3 number:").split())
V=a*b*c
print("V=", V)
```

Возможность записи трех переменных через запятую в строку и присваивание им введенных значений связано с существованием в Python такого неизменяемого типа данных как кортежи. Не будем их подробно здесь рассматривать, отметим лишь, что с этим же типом

связана возможность поменять значения переменных («ярлычков») местами командой:

```
a, b = b, a
```

В других же языках программирования для этого надо использовать третью переменную или какую-нибудь встроенную функцию, например, swap(a, b).

Возвращаясь к нашему примеру 2, заметим, что написать на Python код, при выполнении которого вводить данные можно как угодно, по аналогии с C++, не получится.

Рассмотрим подробнее строку ввода из Листинга …. Здесь используется чудесная функция map(параметр1, параметр2), которая в качестве первого параметра принимает какую-либо функцию, в нашем случае это функция преобразования элемента в целочисленный тип данных, и эта функция применяется к каждому элементу списка, находящихся в параметр2. Но input() на вход получает строку вида '5 6 7' и сразу применить к ней функцию преобразования в целое число невозможно, для этого сначала её надо представить в виде списка строк '5', '6', '7'. Это и делает метод split(). Если у него нет параметра, то по умолчанию разделителем считается пробел.

Заключение

Мы рассмотрели только маленькую часть особенностей Python по сравнению с другими языками программирования. Но уже из этого видно, что идеология (философия) этого языка программирования отличается от той, которая присутствует в языках C++ и Pascal, которые использовались в школах и вузах для изучения программирования.

Несмотря на кажущуюся простоту, Python имеет очень много возможностей, которые надо внимательно изучать, чтобы пользоваться ими в полной мере.

Список литературы

1. Кубряков, Е. А. Алгоритмическая культура мышления / Е. А. Кубряков// Информатика: проблемы, методология, технологии. Информатика в образовании материалы XVIII Международной школы-конференции. – Воронеж, 2018. – С. 17-22.

2. Кубряков, Е. А. О подходах к решению задач ЕГЭ по информатике и ИКТ, проводимого в компьютерной форме / Е. А. Кубряков, С. О. Башарина, В. М. Дубов // Информационные технологии в образовательном процессе вуза и школы. Материалы XV Всероссийской научно-практической конференции. Редколлегия: Р.М. Чудинский (науч. ред.) [и др.]. – Воронеж, 2021. – С. 231-237.

3. Гаркавенко, Г. В. Использования ассемблера для изучения архитектуры компьютера / Г. В. Гаркавенко // Информатика: проблемы, методы, технологии. Материалы XXI Международной научно-методической конференции. – Воронеж, 2021. – С. 1665-1672.

4. Гаркавенко, Г. В. Применение математических знаний при написании программ / Г. В. Гаркавенко // Информатика: проблемы, методы, технологии. Материалы XXII Международной научно-практической конференции им. Э.К. Алгазинова. Под редакцией Д.Н. Борисова. – Воронеж, 2022. – С. 1292-1301.

5. Башарина, С. О. Исследование особенностей работы с вещественными числами / С. О. Башарина, Г. В. Гаркавенко, Е. Р. Найденкина // Информатика: проблемы, методы, технологии. Материалы XX Международной научно-методической конференции. Под редакцией А.А. Зацаринного, Д.Н. Борисова. (Воронеж, 13-14 февраля 2020 г.) – Воронеж, 2020. – С. 1791-1797.

6. Logan Jones. Pointers in Python: What's the Point? [Электронный ресурс]: статья. – Режим доступа: https://realpython.com/pointers-in-python/