

## **СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ: ПАСКАЛЬ, C++, PYTHON**

**Химич Анна Васильевна**

студент, Брянский государственный университет имени академика И. Г. Петровского, РФ, г. Брянск

Язык программирования Python был разработан в конце XX века, но все больше и больше набирает свою популярность не только среди опытных программистов, но и в области обучения языкам программирования.

Python – это интерпретируемый, объектно-ориентированный язык программирования высокого уровня с динамической типизацией, автоматическим управлением памятью и удобными высокоуровневыми структурами данных, такими как списки, кортежи и словари. Поддерживает классы, модули, обработку исключений, а также многопоточные вычисления. Достоинством языка является то, что он обладает простым и выразительным синтаксисом, поддерживает структурное, объектно-ориентированное, функциональное и аспектно-ориентированное программирование [2].

Язык программирования Python эффективен с точки зрения разработчика: программы делают больше, чем на многих других языках, в гораздо меньшем объеме кода. Значительно простой синтаксис позволяет легко читать код программы, возникает меньше проблем с отладкой и расширением программ. Он же является преимуществом при выборе языка начинающим программистом.

Одним из начальных понятий при изучении программирования является понятие переменной. Для успешного обучения написания оптимального программного кода очень важно понять, что такое переменная, научиться хранить в ней данные и использовать их в программах.

Понятие переменной в языке программирования C++ как и в языке программирования Паскаль тождественно: под переменной понимают поименованный участок памяти, в котором хранится значение определенного типа. Другими словами переменная – это именованная область памяти для хранения данных, которые могут изменяться в процессе выполнения программы. Исходя из определения, отличительными характеристиками данного понятия являются тип, идентификатор и значение. Имена и типы переменных указываются при описании и не могут изменяться в процессе выполнения программы. В Паскале тип переменной объявляется после служебного слова var, в разделе описания переменных.

В языке C++ переменная может быть объявлена в любом месте программы. Минусом является то, что созданная переменная, пусть в ней даже ничего не хранится, в Паскаль и C++ уже занимает определенный объем оперативной памяти компьютера. И этот объем, в свою очередь, зависит от типа переменной.

Синтаксис языка программирования C++ упрощает синтаксис языка программирования Паскаль. А Python, в свою очередь, упрощает синтаксис языка C++. Поэтому Python имеет много схожего с C++, но также решает более широкий спектр вопросов, чем его предшественники.

В языке программирования Python переменная является простейшим указателем на значение определенного типа. Поэтому она инициализируется присваиванием некоторому

идентификатору значения при помощи оператора присваивания. Таким образом, еще одно преимущество языка Python состоит в том, что переменные не занимают столько места в оперативной памяти компьютера. Тип переменных в данном языке не объявляется, он определяется автоматически при выполнении оператора присваивания.

Существуют различные классификации типов данных. Например, типы данных подразделяются на стандартные и заданные пользователем. К стандартным типам данных относятся `int` (целый), `float` (вещественный с плавающей точкой), `char` (символьный), `string` (строковый). К типам данных, заданным пользователем, относятся массивы и структуры.

Самой распространенной упрощенной структурой, реализованной практически во всех языках программирования, является массив. Рассмотрим подробнее использование массивов в языках программирования Паскаль, C++ и Python.

Массивы состоят из ограниченного числа компонент одного и того же типа. Структура массива однородна, к любой его компоненте можно обращаться произвольным образом и получить значение нужного элемента по его индексу.

В языке Паскаль тип массива задается с использованием специального слова `array` (англ. – массив), после чего указывается тип индекса массива и тип элементов массива. С помощью индекса можно обращаться с отдельным элементом любого массива, как к обычной переменной заданного типа: можно получать значение этого элемента, отдельно присваивать ему значение, использовать его в выражениях. Недостатком языка Паскаль является то, что переменные в массиве можно перебирать с интервалом равным 1 в прямом или обратном порядке с помощью стандартного цикла обхода (цикл `for` – цикл с параметром).

Доработка языка программирования C++ позволила шире использовать все возможности работы с массивами. При инициализации массива можно задать его размер, но не описывать его значения, можно задать размер и тут же описать этот массив, а можно описать массив, не задавая при этом его размера.

Важным отличием является также то, что можно перебирать элементы массива не только в прямом и обратном порядке, а так же устанавливать интервал перебора элементов с помощью того же стандартного цикла обхода. Это является значительным плюсом языка C++ в работе с массивами, а также преимуществами данной циклической конструкции `for`.

Массивы в Паскаль и C++ похожи тем, что требуют указывать точный диапазон использования элементов массива, не позволяя расширять его динамически.

Для хранения большого количества данных в Python вместо массивов используют такие структуры данных как кортежи и списки. Таким образом, понятие массивов претерпевает изменения. Классифицируются типы данных, заданных пользователем, на изменяемые и неизменяемые. К неизменяемым относятся кортежи, к изменяемым – списки и словари.

Кортеж используется для представления неизменяемой последовательности разнородных объектов. Его элементы записываются в круглых скобках. Так как кортеж – один из типов последовательностей, то всё, что можно применять к строкам в Python, верно и по отношению к кортежам. Можно найти длину кортежа, вывести все его элементы на экран с помощью цикла `for`, а также, пользуясь оператором `in`, проверить кортеж на вхождение какого-либо элемента. Кортежи можно сцеплять, создавать на них срезы.

Кортежи – хороший способ манипулировать элементами разных типов в составе одной последовательности. Но в силу того, что кортеж неизменяем, иногда возникают небольшие проблемы. Однако существуют последовательности другого типа, называемые списками. Главным преимуществом списка является то, что он изменяем. С ним можно проводить те же самые операции, что и с кортежем и даже больше: его элементы можно удалять, а также добавлять новые. Список, как и кортеж, представляет собой последовательность элементов, пронумерованных от 0, как символы в строке. Список можно задать перечислением элементов в квадратных скобках.

Кортежи и списки позволяют упорядочить данные в виде последовательностей. Со словарями дело обстоит немного иначе. В них информация представлена не как последовательность элементов, а как набор пар. Заметно сходство словарей с такой структурой данных как вложенный массив. Элементы словаря заключаются в фигурные скобки.

При работе с кортежами, словарями и списками удобно пользоваться функцией `len(s)`, где `s` последовательность. Данная функция определяет длину последовательности. Необходимо заметить, что в Паскаль и C++ такой функции нет! Чтобы в Паскале и C++ определить количество элементов в массиве, необходимо инициализировать новую переменную, присвоить ей значение 0, а затем с помощью цикла `for` перебрать все элементы массива, увеличивая созданную переменную с каждым элементом на 1 – стандартный цикл обхода массива.

Большим достоинством языка программирования Python является то, что к его изменяемым структурам данных, то есть к спискам и словарям, применяется определенный ряд встроенных методов, значительно упрощающих работу с этими типами данных.

Еще одним важным элементом сравнения для анализируемых языков программирования является запись блоков или составных операторов.

Для логически верного использования конструкций языка важно помнить, что в Паскаль блоки выделяются с помощью служебных слов `begin – end`, в C++ – с помощью фигурных скобок `{ }`. В языке программирования Python дело обстоит гораздо иначе. Здесь большую роль играют отступы. Вложенные инструкции объединяются в блоки по величине отступов. Отступ может быть любым, главное, чтобы в пределах одного вложенного блока отступ был одинаков. Но не стоит забывать про читаемость кода! Именно эта особенность Python и повышает читаемость кода только лишь спецификацией языка.

Например, при записи условного оператора в первой строке размещается условие (логическое выражение), а в блоке с отступом – практически любое действие. Если условие истинно, то выполняется код в блоке после команды `if`, а если ложно, этот код игнорируется.

Нередко в программе требуется проверять более двух возможных ситуаций. Для этого в Python предусмотрен синтаксис `if – elif – else`. Python выполняет только один блок в этой цепочке. Все условия проверяются по порядку до тех пор, пока одно из них не окажется истинным. Далее выполняется код, следующий за этим условием, а все остальные проверки Python пропускает. Для языков C++ и Паскаль реализация подобных ситуаций в коде требует использование вложенных конструкций, что усложняет читабельность кода и требует усиления внимания программиста к выполнению правил их использования («else относится к ближайшему if»).

Использование в Python конструкции `if – elif – else` похоже на использование операторов выбора `case of` и `switch()` в Паскале и C++ соответственно.

Циклы `while` с предусловием и постусловием в данных языках программирования особого отличия не имеют.

Цикл со счетчиком или, как он еще называется — цикл с параметром в Паскале служит для организации повторений с заранее известным их количеством. Оператор в теле конструкции может быть одиночным или составным оператором. Счетчик цикла или параметр цикла – это обычная переменная, которая может быть только порядкового (целочисленного, символьного, логического) или перечислимого типа. При прямом направлении цикла счетчик в каждой итерации всегда увеличивается на единицу, при обратном направлении – уменьшается на единицу. Важно знать: параметр цикла нельзя изменять внутри тела цикла! Также нельзя менять его начальное и конечное значения.

В C++ дело обстоит немного иначе. Сначала инициализируется параметр цикла, после чего ставится точка с запятой. Затем задается конечное значение счетчика цикла. Снова ставим точку с запятой. Задаем шаг цикла. Шаг цикла – это значение, на которое будет увеличиваться или уменьшаться счетчик цикла при каждом проходе.

Чтобы в Python использовать цикл for необходимо сначала написать служебное слово for, потом имя переменной, в которой предполагается сохранять значения элементов, потом in, затем имя перебираемой последовательности, двоеточие и, наконец, тело цикла. Вот и всё.

В проведенном выше анализе языков программирования высокого уровня: Паскаль, С++ и Python были выделены основные преимущества последнего и одновременно представлена линейка синтаксических усовершенствований, которые дают представления о направлениях эволюции языков программирования.

### **Список литературы:**

1. Доусон М. Програмируем на Python. - СПб.: Питер, 2017. - 416
2. Мэтиз Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. - СПб.: Питер, 2017. - 496 с.