**Лекция 14**

**Основы алгоритмизации. Язык программирования Python.**

**Компьютерная программа**

Прежде чем приступить к программированию, давайте сначала разберемся с компьютерными программами и тем, что они делают.

Компьютерная программа - это набор организованных инструкций, которые заставляют компьютер выполнять ту или иную задачу.

Языки программирования - это набор слов, символов и кодов, который позволяет людям общаться с компьютерами. Существуют сотни языков программирования, которые можно использовать для написания компьютерных программ, и вот некоторые из них:

* C
* C++
* Java
* Python
* PHP
* Perl
* Ruby

**Алгоритм**

Алгоритм - это подробный набор инструкций для выполнения операции или решения задачи.

Блок-схема - это графическое представление шагов, которые необходимо выполнить для решения задачи. Она состоит из набора символов. Каждый символ представляет определенное действие.

|  |  |
| --- | --- |
|  **Символ** | **Значение** |
|  | Начало / Конец |
|  | Процесс |
|  | Ввод / Вывод |
|  | Решение |
|  | Подготовка или Инициализация |
|  | Связь |
|  |  |

**Программная среда**

У программистов есть множество инструментов для улучшения процесса разработки программного обеспечения. Некоторые общие инструменты включают в себя

* Редакторы. Редактор позволяет программисту вводить исходный код программы и сохранять его в файлы. Большинство программных редакторов повышают производительность труда программистов путем применения цветового выделения языковых функций. Синтаксис языка имеет отношение к тому, как организовать языковые единицы таким образом, чтобы составить правильно сформированные предложения.
* Компиляторы. Компилятор переводит исходный код в целевой. Целевой код может быть машинным языком для конкретной платформы или встроенного устройства. Целевым кодом может быть другой исходный язык: например, самый ранний компилятор C ++ переводил C ++ в C. Компиляторы переводят содержимое исходного файла и создают файл, содержащий весь целевой код. Популярные компилированные языки включают C, C ++, Java, C #.
* Интерпретаторы. Интерпретатор похож на компилятор в том, что он переводит исходный код более высокого уровня в целевой код (машинный язык). В то время как компилятор создает исполняемую программу, которая может запускаться много раз без необходимости дополнительного перевода, интерпретатор переводит операторы исходного кода на машинный язык каждый раз, когда пользователь запускает программу. Скомпилированная программа не нуждается в повторной компиляции для того, чтобы запуститься, но интерпретируемую программу необходимо интерпретировать повторно при каждом запуске. По этой причине интерпретируемые языки часто называют языками сценариев. Популярные языки сценариев включают Python, Ruby, Perl и для веб-браузеров Javascript.
* Отладчики. Отладчик позволяет программисту легче отслеживать выполнение программы для поиска и исправления ошибок в ходе реализации этой программы.

**Введение в Python**

Гвидо ван Россум создал языки программирования Python в конце 1980-х годов. В отличие от других популярных языков, таких как C, C ++, Java и C #, Python стремится обеспечить простой, но мощный синтаксис.

Python используется для разработки программного обеспечения в таких компаниях и организациях, как Google, Yahoo, Facebook, Youtube, NASA и т.д.

Дополнительную информацию о Python, включая ссылки для загрузки последней версии для Microsoft Windows, Mac OS X и Linux, можно найти по адресу http://www.python.org.

**Основной синтаксис**

Язык Python имеет много общего с другими языками, но между ними есть определенные различия.

Идентификатор Python - это имя, которое мы присваиваем для идентификации переменной, функции, класса, модуля или другого объекта. Есть несколько правил написания идентификатора. Помните, что Python чувствителен к регистру.

1. Идентификаторы могут представлять собой комбинацию прописных и строчных букв, цифр или подчеркиваний (\_). Таким образом, myVariable, variable\_1, variable\_for\_print все являются действительными идентификаторами Python.
2. Идентификатор не может начинаться с цифры. Таким образом, хотя имя variable1 является действительным идентификатором, имя 1variable является недействительным.
3. Мы не можем использовать в нашем идентификаторе специальные символы, такие как !, #, @,%, $ и т.д.
4. Идентификатор может быть любой длины.

**Комментарии**

Python может содержать операторы, заключенные в # ... для однострочных, и ««… .. »» для многострочных комментариев. Эти комментарии используются для того, чтобы сделать программы удобными для пользователя и легкими для понимания. Что хорошо в комментариях, так это то, что они полностью игнорируются компиляторами и интерпретаторами. Таким образом, вы можете использовать любой язык, на котором вы хотите написать свои комментарии.

**Типы данных**

Одна фундаментальная вещь, которую должен уметь делать любой язык программирования, заключается в представлении элементов данных. Python обеспечивает несколько встроенных типов данных: базовые типы (целочисленные, с плавающей запятой, строковые, логические) и типы контейнеров (списки, кортеж, множество, словарь).

Целые числа (2, 4, 20) имеют тип int, а числа с дробной частью (5.0, 1.6) имеют тип float.

Помимо чисел, Python также может манипулировать строками, которые могут быть выражены несколькими способами. Они могут быть заключены в одинарные кавычки (‘…’) или двойные кавычки (“…”) с одинаковым результатом.

Python знает несколько составных типов данных, которые используются для группировки других значений. Наиболее универсальным является список, который можно записать в виде списка значений, разделенных запятыми, в квадратных скобках. Списки могут содержать элементы разных типов, но обычно все элементы имеют одинаковый тип.



Множество - это неупорядоченная подборка без повторяющихся элементов. Для создания множеств можно использовать фигурные скобки или функцию set ().



Словари иногда встречаются на других языках в форме «ассоциативной памяти» или «ассоциативных массивов». Словари индексируются по ключам. Основные операции по словарю заключаются в сохранении значения с некоторым ключом и извлечении этого значения с учетом данного ключа. Пара скобок создает пустой словарь: {}.

