Глава2. Язык программирования Python

П1. Особенности языка программирования Python

Для того, чтобы начать разговаривать о языке программирования Python, для начала нужно разобраться, что такое в целом компьютерная лингвистика, что она вообще позволяет делать и для чего её используют.

Компьютерная лингвистика – одно из научных направлений в прикладной лингвистике, направленное на изучение лингвистических основ информатики, моделирования языка и мышления в компьютерной среде с помощью компьютерных программ, а также направленное на использование различных компьютерных инструментов – программ, различных компьютерных технологий. Данное направление появилось в 1960-х годах. Компьютерная лингвистика возникла в результате сочетания лингвистики, математики, информатики и искусственного интеллекта. Одна из главных ее задач - это разработка компьютерных программ для автоматической обработки текстов на естественном языке. Данное направление достаточно широко используется в настоящее время, так как значительно упрощает работу с материалами в сфере лингвистики.

Ну а теперь о самом языке программирования Python. Python – довольно простой и понятный язык программирования, один из самых популярных языков программирования на данный момент. У него много преимуществ по сравнению с другими языками.

Главные особенности:

1. Простой
2. Лёгкий в освоении

Язык Python обладает достаточно простым синтаксисом, поэтому на нем очень легко начать программировать, даже если вы не связывались с программированием ранее.

1. Свободный и открытый
2. Язык высокого уровня

При написании программы не приходится отвлекаться на низкоуровневые детали, такие как управление памятью и т.п

1. Портируемый

Python был портирован на множество платформ, поэтому все его программы смогут запускаться с любой платформы без каких-либо изменений.

1. Интерпретируемый
2. Объектно-ориентированный
3. Расширяемый
4. Встраиваемый

Python можно встраивать в другие программы, такие как С/С++, предоставляя при этом пользователям возможности написания различных сценариев и т.д

Помимо особенностей, Python предоставляет множество различных возможностей. Вот некоторые из них:

1. Работа с xml/html файлами
2. Создание веб-сценариев
3. Робототехника
4. Работа с http запросами
5. Работа с FTP
6. Программирование математических и научных вычислений
7. GUI (графический интерфейс)
8. Работа с изображениями, аудио и видео файлами

П2. Программы

Главной моей задачей было написать программы, которая будет иметь следующие свойства: на вход каждая из программ (кроме первой) будет получать какое-то слово на немецком языке, а далее одна из программ будет выводить на экран все одиночные звуки этого слова, а последняя будет выводить звуки различных сочетаний букв. Первая программа на вход будет получать букву, а программа будет выводить на экран тип буквы (гласная, согласная или дифтонг).

Обучение программированию на языке «Python» я прошла на сайте <https://www.codecademy.com/>, где и изучила основные свойства языка.

Итак, начну с первой программы. Главной её задачей является вывод тип буквы, вводимой в программу, на экран. Для написания этой программы необходимо быть знакомым с синтаксисом языка программирования «Python». Я буду использовать условный оператор, который описывается данным образом:

*def():*

*if условие 1:*

*действие1*

 *elif условие 2:*

 *действие2*

 *else:*

 *действие 3*

Но для того, чтобы описывать каждую букву таким образом, для начала нужно прописать основные моменты, такие как типы букв. Для каждого типа я прописываю список букв, соответствующих ему. Этот список будет выглядеть так:

*glas = [ a, e, i, o, u, y]*

*sonor = [m, n, l, r]*

*gluh = [p, t, k, f, s, x, h]*

 *zvonk = [b, g, v, z, d, j]*

*diftong = [ü, ö, ä]*

После этого нужно прописать этот список в условном операторе, а затем организовать программу. Итоговая программа будет выглядеть данным образом:

*def deutsch() :*

 *print "Сейчас вы будете вводить слово"*

 *bukva = raw\_input("Введите букву: " ).lower()*

 *glas = [ a, e, i, o, u, y]*

 *sonor = [m, n, l, r]*

 *gluh = [p, t, k, f, s, x, h]*

 *zvonk = [b, g, v, z, d, j]*

 *diftong = [ü, ö, ä]*

 *if bukva == glas:*

 *print "Гласная"*

 *elif bukva == sonor:*

 *print "Согласная сонорная"*

 *elif bukva == gluh:*

 *print "Согласная глухая"*

 *elif bukva == zvonk:*

 *print "Согласная звонкая"*

 *else:*

 *print "Дифтонг"*

В итоге если ввести букву “m”, программа выведет на экран “Согласная сонорная”.

Теперь о второй программе. Её главной задачей является вывод на экран всех одиночных звуков слова. В этой программе я тоже буду пользоваться условным оператором, но только немного иным образом. Для начала я должна обозначить все одиночные звуки. Их совсем немного:

*p = /p/*

*b = /b/*

*t = /t/*

*d = /d/*

*g = /g/*

*h = /h/*

*j = /ʒ/*

*x = /ks/*

*r = /r/*

*l = /l/*

*m = /m/*

*n = /n/*

*w = /v/*

*v = /v/*

*s = /s/*

*ß = /s/*

*ä = /ε:/*

*a = /a:/*

*i = /i:/*

*e = /e:/*

*u = /u:/*

*o = /o:/*

*y = /y:/*

После описания данных букв можно приступать к формированию полноценной программы. Она будет выглядеть примерно так:

*def deutsch() :*

 *print "Сейчас вы будете вводить слово"*

 *bukva = raw\_input("Введите букву: " ).lower()*

 *if bukva == p:*

 *print ‘p’*

 *elif bukva == b:*

 *print ‘b’*

 *elif bukva == t:*

 *print ‘t’*

 *elif bukva == d:*

 *print ‘d’*

 *elif bukva == g:*

 *print ‘g’*

 *elif bukva == h:*

 *print ‘h’*

 *elif bukva == j:*

 *print ‘ʒ’*

 *elif bukva == x:*

 *print ‘ks’*

 *elif bukva == r:*

 *print ‘r’*

 *elif bukva == l:*

 *print ‘l’*

 *elif bukva == m:*

 *print ‘m’*

 *elif bukva == n:*

 *print ‘n’*

 *elif bukva == w:*

 *print ‘v’*

 *elif bukva == v:*

 *print ‘v’*

 *elif bukva == s:*

 *print ‘s’*

 *elif bukva == ß:*

 *print ‘s’*

 *elif bukva == ä:*

 *print ‘ε:’*

 *elif bukva == a:*

 *print ‘a:’*

 *elif bukva == i:*

 *print ‘i:’*

 *elif bukva == e:*

 *print ‘e:’*

 *elif bukva == u:*

 *print ‘u:’*

 *elif bukva == o:*

 *print ‘o:’*

 *else:*

 *print ‘y:’*

В итоге, если ввести слово “fremd”, то программа выведет на экран “frmd”

Теперь приступаю к написанию последней программы. Она будет немного отличаться от предыдущих двух и она будет сложнее в написании. Её главной задачей является вывод на экран звуков, обозначающих сочетание букв. Для этого, для начала я должна ввести все возможные сочетания букв и их правила чтения, то есть соответствующие звуки. Их получилось очень много, поэтому я не буду их всех прописывать, а напишу сразу программу. Программа получилась довольно большой. А вот и она:

*def deutsch() :*

 *print "Сейчас вы будете вводить слово"*

 *bukva = raw\_input("Введите букву: " ).lower()*

 *if bukva == aa:*

 *print ‘a:’*

 *elif bukva == ah:*

 *print ‘a:’*

 *elif bukva == äh:*

 *print ‘ε:’*

 *elif bukva == ee:*

 *print ‘e:’*

 *elif bukva == eh:*

 *print ‘e:’*

 *elif bukva == ieh:*

 *print ‘i:’*

 *elif bukva == ie:*

 *print ‘i:’*

 *elif bukva == ih:*

 *print ‘i:’*

 *elif bukva == oo:*

 *print ‘o:’*

 *elif bukva == oh:*

 *print ‘o:’*

 *elif bukva == öh:*

 *print ‘ø:’*

 *elif bukva == oe:*

 *print ‘ø:’*

 *elif bukva == uh:*

 *print ‘u:’*

 *elif bukva == üh:*

 *print ‘y:’*

 *elif bukva == ei:*

 *print ‘aɪ̯’*

 *elif bukva == ey:*

 *print ‘aɪ̯’*

 *elif bukva == ai:*

 *print ‘aɪ̯’*

 *elif bukva == ay:*

 *print ‘aɪ̯’*

 *elif bukva == au:*

 *print 'aʊ̯’*

 *elif bukva == eu:*

 *print ‘ɔʏ̯’*

 *elif bukva == äu:*

 *print ‘ɔʏ̯’*

 *elif bukva == am:*

 *print ‘a’*

 *elif bukva == an:*

 *print ‘a’*

 *elif bukva == al:*

 *print ‘a’*

 *elif bukva == ar:*

 *print ‘a’*

 *elif bukva == em:*

 *print ‘ε’*

 *elif bukva == en:*

 *print ‘ε’*

 *elif bukva == el:*

 *print ‘ε’*

 *elif bukva == er:*

 *print ‘ε’*

 *elif bukva == il:*

 *print ‘i’*

 *elif bukva == im:*

 *print ‘i’*

 *elif bukva == in:*

 *print ‘i’*

 *elif bukva == ir:*

 *print ‘i’*

 *elif bukva == ol:*

 *print ‘ɔ’*

 *elif bukva == om:*

 *print ‘ɔ’*

 *elif bukva == or:*

 *print ‘ɔ’*

 *elif bukva == on:*

 *print ‘ɔ’*

 *elif bukva == un:*

 *print ‘ʊ’*

 *elif bukva == um:*

 *print ‘ʊ’*

 *elif bukva == ul:*

 *print ‘ʊ’*

 *elif bukva == ur:*

 *print ‘ʊ’*

 *elif bukva == yl:*

 *print ‘ʏ’*

 *elif bukva == ym:*

 *print ‘ʏ’*

 *elif bukva == yn:*

 *print ‘ʏ’*

 *elif bukva == yr:*

 *print ‘ʏ’*

 *elif bukva == pp:*

 *print ‘p’*

 *elif bukva == rb:*

 *print ‘p’*

 *elif bukva == lb:*

 *print ‘p’*

 *elif bukva == nb:*

 *print ‘p’*

 *elif bukva == mb:*

 *print ‘p’*

 *elif bukva == bb:*

 *print ‘b’*

 *elif bukva == tt:*

 *print ‘t’*

 *elif bukva == th:*

 *print ‘t’*

 *elif bukva == rd:*

 *print ‘t’*

 *elif bukva == md:*

 *print ‘t’*

 *elif bukva == nd:*

 *print ‘t’*

 *elif bukva == ld:*

 *print ‘t’*

 *elif bukva == ck:*

 *print ‘k’*

 *elif bukva == ch:*

 *print ‘k’*

 *elif bukva == rg:*

 *print ‘k’*

 *elif bukva == mg:*

 *print ‘k’*

 *elif bukva == lg:*

 *print ‘k’*

 *elif bukva == dd:*

 *print ‘d’*

 *elif bukva == gg:*

 *print ‘g’*

 *elif bukva == ff:*

 *print ‘f’*

 *elif bukva == ph:*

 *print ‘f’*

 *elif bukva == ss:*

 *print ‘s’*

 *elif bukva == sch:*

 *print ‘ʃ’*

 *elif bukva == ch:*

 *print ‘ʃ’*

 *elif bukva == st:*

 *print ‘ʃ’*

 *elif bukva == ge:*

 *print ‘ʒ’*

 *elif bukva == ju:*

 *print ‘j’*

 *elif bukva == ye:*

 *print ‘j’*

 *elif bukva == che:*

 *print ‘ç’*

 *elif bukva == ig:*

 *print ‘ç’*

 *elif bukva == rr:*

 *print ‘r’*

 *elif bukva == rh:*

 *print ‘r’*

 *elif bukva == er:*

 *print ‘ɐ’*

 *elif bukva == ll:*

 *print ‘l’*

 *elif bukva == mm:*

 *print ‘m’*

 *elif bukva == nn:*

 *print ‘n’*

 *elif bukva == ng:*

 *print ‘ŋ’*

 *elif bukva == pf:*

 *print ‘pf’*

 *elif bukva == ze:*

 *print ‘ts’*

 *elif bukva == tz:*

 *print ‘ts’*

 *elif bukva == tsch:*

 *print ‘t͡ʃ’*

 *elif bukva == tch:*

 *print ‘t͡ʃ’*

 *elif bukva == chh:*

 *print ‘t͡ʃ’*

 *elif bukva == qu:*

 *print ‘kv’*

 *else:*

 *print ‘ks’*

В итоге, если ввести слово “lieben”, то программа выведет на экран “/i:/ /ε/”, так в этом слове присутствует только два буквенных сочетания.

Для получения полноценной программы необходимо объединить все три программы в одну, но этим я займусь в 10 классе при написании диплома.