ГБОУ гимназия №1505

«Московская городская педагогическая гимназия-лаборатория»

Реферат на тему

«Анализ немецкого языка компьютерными методами»

Ученицы 9В класса Захаренковой Дарьи

Научный руководитель:

к.ф.н Евдокимова А.А

Москва 2016

Содержание: стр.

Введение …………………………………………………………………………… 3

Глава 1. Фонетика и словообразование немецкого языка:

§1. Общие правила фонетики …………………………………………….. 4 - 7

§2. Словообразование в немецком языке ………………………………... 8 - 9

§3. Ударение в словах …………………………………………………….. 9

Глава 2. Язык программирования «Python»:

§1. Особенности языка программирования «Python» …………………... 10 - 11

§2. Программы …………………………………………………………….. 11 - 23

Заключение ………………………………………………………………………... 24

Список источников ……………………………………………………………….. 25

Введение

Тема моего реферата – анализ немецкого языка компьютерными методами. Компьютерный анализ – один из методов прикладной лингвистики. Его часто используют в компьютерной лингвистике. Компьютерная лингвистика – направление в областях математического и компьютерного исследования. Разные иностранные языки имеют различные методы их описания. Мне интересно рассмотреть немецкий язык с точки зрения компьютерной лингвистики и выяснить, какие особенности в компьютерной лингвистике, как именно он описывается и т.д. Но для начала следует изучить какой – нибудь язык программирования. Для изучения я возьму язык «Python». Он довольно прост и понятен.

Итак, целью моего реферата можно считать изучение особенностей немецкого языка в компьютерной лингвистике. Поэтому можно поставить себе несколько основных задач:

• Знакомство с языком «Python»

• Изучение его особенностей

• Изучение принципов описания языков

• Анализ немецкого языка с точки зрения фонетики и словообразования

• Написание программ на языке программирования «Python»

Объект исследования: анализ особенностей фонетики и словообразования немецкого языка методами компьютерной лингвистики.

Предмет исследования: особенности фонетики и словообразования немецкого языка.

В своей работе я использовала различные книги, а также несколько статей из интернета. Все их названия можно просмотреть в разделе «Источники».

Глава 1. Фонетика и словообразование немецкого языка

§1. Общие правила фонетики

Немецкий язык - один из самых простых языков по своей грамматике. Фонетика немецкого языка, по сравнению с другими языками, тоже довольно проста. Тем не менее в ней присутствуют некие сложности.

В немецкой фонетике в общей сложности насчитывается 44 различных звука, из которых 16 гласных звуков, 22 согласных звука, 3 дифтонга и 3 аффриката. Дифтонг – сложный гласный звук, состоящий из слогового и неслогового элементов. Аффрикат – согласные звуки, представляющие собой слитное сочетание взрывного согласного звука с фрикативными.

Гласные звуки немецкого языка делятся на краткие и долгие, переднего и заднего плана. Длительность гласных обычно зависит от особенностей слога, в образовании которого они участвуют.

Ниже приведена таблица гласных звуков и дифтонгов в немецком языке, с их фонетическими характеристиками и соответствующими буквами и буквосочетаниями.

|  |  |  |
| --- | --- | --- |
| **Звуки** | **Соответствующие буквы и буквосочетания** | **Примеры** |
| /a:/ | a, aa, ah | Rat, Tag, Tat |
| /a/ | а | dann, Mann, wann |
| /ε/ | e, ä | Hemd, fremd, Zwerg |
| /ε:/ | ä, äh | gären, Bär, fähig |
| /ə/ | е | wagen, sagen, fragen, jagen |
| /e:/ | e, ee, eh | wehen, leben, leer |
| /ı/ | i | blind, Wind, Kind |
| /i:/ | i, ieh, ie, ih | wiegen, geliehen, Kino |
| /ɔ/ | о | sollen, Zorn, wollen, Horn |
| /o:/ | o, oo, oh | wohl, bohren, Kohl, Bote |
| /œ/ | ö | löffeln, Wölfe, Gewölbe |
| /ø:/ | ö, öh, oe | Löhne, Höhle, Röhre, böse |
| /ʊ/ | u | Bund, Hund, Burg |
| /u:/ | u, uh | Blut, gut, Rute |
| /ʏ/ | ü, y | fünfzig, Gewürze, schützen |
| /y:/ | ü, üh, y | Führer, Lüge, bügeln, Rügen |
| /aɪ̯/ | ei, ey, ai, ay | leiten, Bayer, Keim, laisieren |
| /aʊ̯/ | au | saugen, Maus, rauchen, Haus |
| /ɔʏ̯/ | eu, äu | Säure, Beutel, deutlich, Räume |

Согласные звуки в немецком языке тоже делятся на несколько групп: звонкие, глухие и сонорные, а также аффрикаты.

Ниже приведена таблица согласных звуков аффрикатов в немецком языке, с их фонетическими характеристиками и соответствующими буквами и буквосочетаниями.

|  |  |  |
| --- | --- | --- |
| **Звуки** | **Соответствующие буквы и буквосочетания** | **Примеры** |
| /p/ | p, pp, b | prüfen, Post, Polster, Korb |
| /b/ | b, bb | Bier, basteln, knabbern |
| /t/ | t, tt, th, d | Ratte, Lot, Theorie, Bund |
| /d/ | d, dd | daraus, addieren, Diele |
| /k/ | k, ck, ch, g | Kino, Wecker, Chor, Burg |
| /g/ | g, gg | Bagger, Wiege, tragen |
| /f/ | f, ff, v, ph | fahren, fabelhaft, Neffe, Vetter, Phase |
| /v/ | w, v | Wetter, Wasser, Wurm, warnen, Klavier |
| /s/ | s, ss, ß | besser, Gras, süß |
| /z/ | s | sauber, Sieb, suchen, Leser |
| /ʃ/ | sch, s, ch | wischen, stehen, Stunde |
| /ʒ/ | g, j | Ingenieur, Genie |
| /j/ | j, y | juristisch, Bayern, Projekt |
| /ç/ | ch, g | wichtig, listig, Chemie |
| /x/ | ch | Suche, Lauch, kochen |
| /h/ | h | Honig, daheim, Hauch |
| /r/ | r, rr, rh | Rom, Riemen, Karren, Rhythmus |
| /ɐ/ | r, er | Fühler, hierher |
| /l/ | l, ll | lieben, lachen, bellen |
| /m/ | m, mm | Mücke, Komma, sammeln |
| /n/ | n, nn | nah, Nebel, Kanne, binden |
| /ŋ/ | ng, n | Klang, danke, Banken |
| /pf/ | pf | Pfiff, pflücken, zapfen |
| /ts/ | tz, z, c, t | duzen, platzen, Cäsar |
| /t͡ʃ/ | tsch, ch, tch | lutschen, Strechhose, Tschess |
| /kv/ | qu | Quote, quer, quellen, Quadrat |
| /ks/ | x, chs | Lachs, sechzig, Xenon |

§2. Словообразование в немецком языке

Словообразование в любом языке – создание новых слов при помощи специальных средств и способов. В немецком языке словообразование делится на три основных группы: словообразование существительных, прилагательных и глаголов.

Существует три способа словообразования существительных: с помощью словообразовательных элементов, а именно суффиксов и префиксов, путем словосложения и путем субстантивации, то есть перехода другой части речи в разряд существительных. Для каждого рода существительных характерны свои суффиксы. Для женского рода: -in, -ung, -heit/-keit, -schaft, -ie, -tion, -ik, -ei, - tät. Для мужского рода: -er/-ler/-ner, -e, -at, -ant, -ent, -ist, -or, -eur. Для среднего рода: -tum, -chen/-lein, -um.

Глаголы бывают простыми, производными и сложными. Простыми называются глаголы, состоящие из корня и суффикса неопределенной формы. Производные глаголы образуются при помощи приставок или суффиксов. В немецком языке у глаголов могут быть два вида приставок: отделяемые и неотделяемые. К неотделяемым приставкам относят: be-, ge-, er-, zer-, miß-, emp-, ent-, ver-. Эти приставки безударны. К отделяем приставкам относят: an-, auf-, aus-, zu-, vor-, mit-, bei- и другие. Отделяемые приставки всегда стоят под ударением.

Приставки durch-, hinter-, über-, um-, unter-, wieder- могут быть как отделяемыми, так и неотделяемыми.

Прилагательные, так же как и другие части речи, образуются с помощью приставок и суффиксов, путем словосложения. Наиболее часто употребляются данные суффиксы: -lich, -ig, -isch, -bar, -sam, -er, -los, -haft. Наиболее часто употребляются данные приставки: miß-, un-, ur-.

Но больше всего меня интересует именно словосложение. Словосложение является одним из самых распространенных способов словосложения. В результате словосложения получаются сложные слова. Наиболее распространены сложные существительные. Сложное существительное может состоять как из двух, так и из трех и более основ. Определяемым словом всегда является последнее слово. Оно, как правило, является существительным. По определяемому слову определяется форма множественного числа, род и тип склонения. Такие сложные существительные в немецком языке образуются несколькими способами словосложения:

1. Словосложение существительного с существительным
2. Словосложение существительного с прилагательным
3. Словосложение существительного с числительным
4. Словосложение существительного с глаголом
5. Словосложение существительного с предлогом

При словосложении между словами могут стоять соединительные элементы, такие как: (e)n, (e)s, e.

§3. Ударение в словах

Ударение в словах в немецком языке – довольно простая и понятная тема. Расставлять ударения совершенно несложно, потому что практически во всех словах ударение ставится одинаково. В сложных словах ударение практически всегда падает на корень определяющего слова, стоящего на первом месте, но иногда, как и во всех языках мира, есть свои исключения. Бывают случаи, когда корень определяющего слова перемещается на второе место, главное ударение при этом сохраняется на нем. В сложных существительных, состоящих из трех основ, часто может происходить сдвиг ударения, обусловленный ритмическими причинами, или может наблюдаться перенос ударения с первого слова на второе.

Глава 2. Язык программирования «Python»

§1. 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

§2. 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.

Заключение

Исследуя особенности морфологии немецкого языка и закономерности языка «Python», его логику, мне удалось создать несколько компьютерных программ, умеющих определять тип буквы, звуки и различные буквенные сочетания. Все задачи, поставленные мной в начале исследования, выполнены. Изучены теоретические основы компьютерной лингвистики, пройден курс по программированию в «Python», изучены такие особенности словообразования и фонетики немецкого языка как ударения в словах, правила словообразования и др. А также достигнута главная цель исследования – созданы три программы. На данный момент программы еще не до конца усовершенствованы, они будет дополняться другой различной информацией, чтобы в итоге получить одну полноценную компьютерную программу, которая будет выполнять несколько функций сразу.

Список источников:

1. <https://www.codecademy.com/>
2. <http://yazykoznanie.ru/>
3. <http://crazylink.ru/germangrammar/gruppen-schwacher-verben.html>
4. <http://www.krugosvet.ru/>
5. <http://deutsch135.blogspot.ru/p/>
6. <http://abc.cvsw.ru/regeln.html>
7. <http://deutsch-online.ru/>
8. <http://ideone.com/>
9. [prezi.com/mce7q34lc2hw/python/](http://text.ru/rd/aHR0cHM6Ly9wcmV6aS5jb20vbWNlN3EzNGxjMmh3L3B5dGhvbi8%3D)
10. "Компьютерная лингвистика для всех: Мифы. Алгоритмы. Язык" Автор: Анисимов А.
11. <http://www.dialog-21.ru/>