МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение  
 высшего образования

«АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Колледж Алтайского государственного университета

Отделение экономики и информационных технологий

**ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

по ПМ. 03 «Участие в интеграции программных модулей»

РАЗРАБОТКА ПРОГРАММЫ ВИРТУАЛЬНОГО СОБЕСЕДНИКА

Выполнил(а) студент(ка)

3 курса 273 - а группы

Фамилия Имя

(ф.и.о.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Научный руководитель

Преподаватель

(уч. степень, звание, категория)

Ракитин Р. Ю.

(ф.и.о.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Работа защищена

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Барнаул 2020

СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc37284702)

[ВВЕДЕНИЕ 3](#_Toc37284703)

[1 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА 5](#_Toc37284704)

[1.1 Анализ требований 5](#_Toc37284705)

[1.2 Проектирование программного продукта 5](#_Toc37284706)

[2 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА 9](#_Toc37284707)

[2.1 Программирование программного продукта 9](#_Toc37284708)

[2.2 Тестирование программного продукта 13](#_Toc37284709)

[3 ДОКУМЕНТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА 16](#_Toc37284710)

[ЗАКЛЮЧЕНИЕ 17](#_Toc37284711)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ И ИСТОЧНИКОВ 18](#_Toc37284712)

# ВВЕДЕНИЕ

В наше время для получения актуальной информации об организации не обязательно общаться с реальным человеком. Всю необходимую информацию может предоставить виртуальный собеседник. Общение происходить быстро и емко, избавляя коммуникацию от «человеческого фактора». Этим и заслужили свое внимание технологии искусственного интеллекта и обработки больших данных.

Задание:

1. Постановка задачи и проектирование программного продукта;
2. Разработка программного продукта;
3. Тестирование разработанного программного продукта и устранение ошибок и недочетов;
4. Составление документации.

План работы над заданием:

1. Анализ требований;
2. Проектирование;
3. Программирование;
4. Тестирование;
5. Развертывание;
6. Документирование.

В ходе прохождения производственной практики были использованы следующие инструменты:

* Python – объектно-ориентированный язык программирования;
* SQLite – свободная реляционная система управления базами данных;
* Chatterbot – библиотека python, которая специализируется на разработке чат-ботов;
* Bash – усовершенствованная и модернизированная вариация командной оболочки Bourne shell.

Внедрение данного программного обеспечения позволит снизить затраты на операторов и осуществит быстрый доступ к необходимой информации.

# 1 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

Данная глава будет посвящена проектированию программы виртуального собеседника.

## 1.1 Анализ требований

Для начала необходимо выявить требования к программному обеспечению.

Целью данного этапа являются наиболее подробное описание того, что программа должна делать и ее спецификация.

После сбора и анализа были выявлены следующие требования:

* Программа должна потенциально интегрироваться в структуру сайта;
* Программа должна вести диалог с пользователем;
* Программа должна, по требованию пользователя, предоставлять всю необходимую информацию об организации;

## 1.2 Проектирование программного продукта

Целью проектирования является определение внутренних свойств системы и детализации её внешних свойств на основе выданных требований к ПО.

Изначально у программы должна быть схема диалогов, исходя из которых она должен понять как, что и когда отвечать в конкретный момент времени.

База знаний должна содержать наборы возможных вопросов пользователя и соответствующих им ответов. В данном случае это будет информация об организации.

На вход поступает некая строка, которую программа должна интерпретировать со строками из уже имеющихся диалогов и выдать соответствующий ответ. Если строка не имеет вхождений в базе данных, то она должна помечаться как неопределенная.

Саму архитектуру и алгоритмы программы предоставляет библиотека chatterbot, в документации к которой описывается структура программ, созданных на её основе.

ChatterBot - это библиотека Python, которая позволяет легко генерировать автоматические ответы на вводимые пользователем данные. ChatterBot использует набор алгоритмов машинного обучения для получения различных типов ответов.

Неподготовленный экземпляр начинает без знания того, как нужно общаться. Каждый раз, когда пользователь вводит выражение, библиотека сохраняет введенный текст и текст, на который оператор ответил. По мере того, как ChatterBot получает больше входных данных, количество ответов, которые он может ответить, и точность каждого ответа по отношению к оператору ввода увеличивается.

Программа выбирает ответ с наиболее близким соответствием путем поиска наиболее подходящего известного оператора, соответствующего входному значению, затем выбирает ответ из выбора известных ответов на этот оператор.



Рис. 1 Процесс работы.

Типичный логический адаптер, предназначенный для возврата ответа на оператор ввода, будет использовать два основных шага для этого. Первый шаг включает в себя поиск в базе данных известного оператора, который соответствует или близко соответствует оператору ввода. Как только совпадение выбрано, второй этап включает выбор известного ответа на выбранное совпадение. Часто существует ряд существующих операторов, которые являются ответами на известное совпадение.

Библиотека включает в себя инструменты, которые помогают упростить процесс обучения экземпляра чат-бота. Учебный процесс ChatterBot включает загрузку примера диалога в базу данных чат-бота. Это либо создает, либо строит на основе структуры данных графа, которая представляет наборы известных операторов и ответов. Когда тренеру бот-чата предоставляется набор данных, он создает необходимые записи в графе знаний чат-бота, чтобы входные данные оператора и ответы были правильно представлены.

Таким образом, если чат-бот имеет функцию самообучения, то база данных будет динамически менять свое содержимое. Будут появляться новые диалоги и возможные ответы, которые бот может выбрать, основываясь на своем предыдущем опыте.

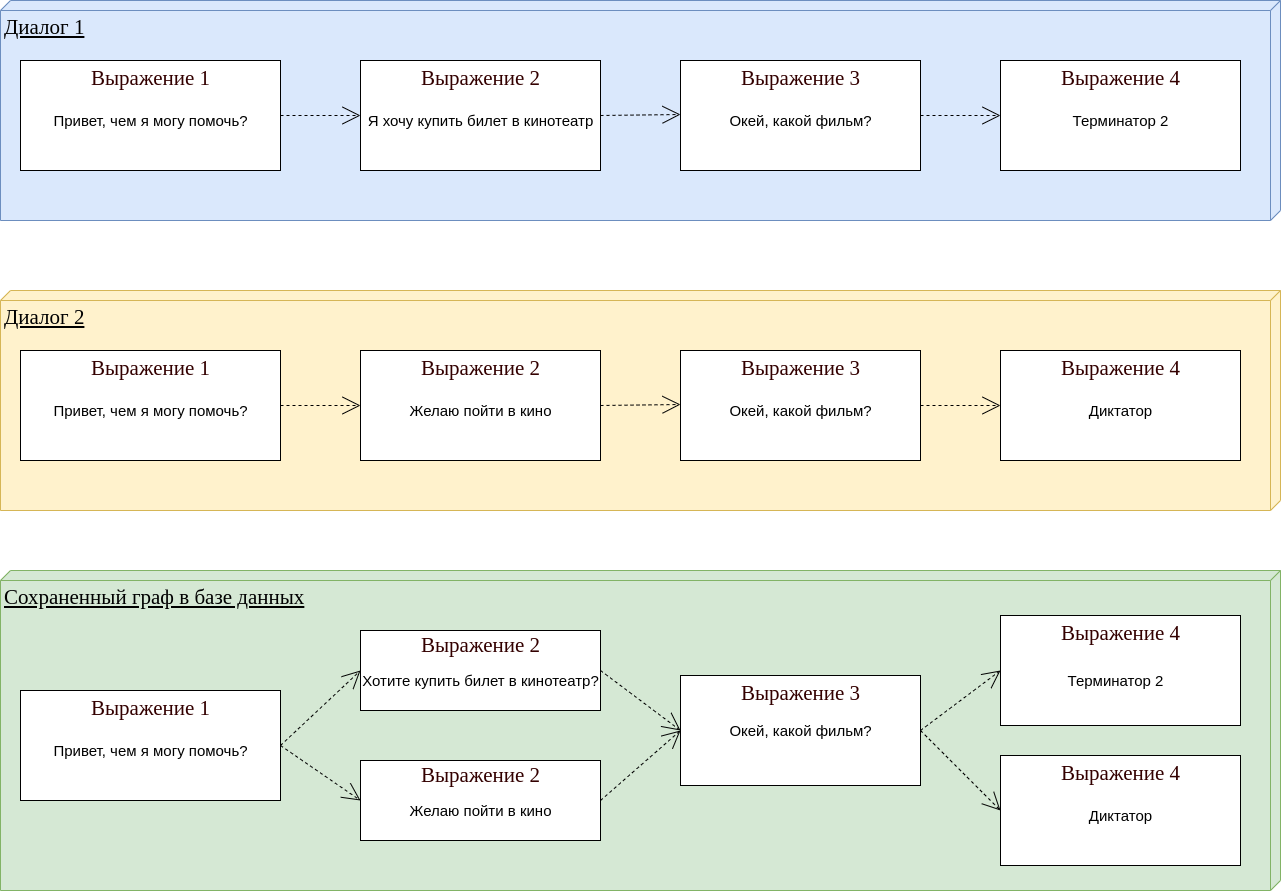


Рис. 2 Представление диалогов.

ChatterBot поддерживает возможность одновременной беседы. Это очень важно в условиях веб-сервера, когда множество клиентов взаимодействуют с одним и тем же программным обеспечением.

# 2 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

Данная глава будет посвящена программированию и тестированию программного обеспечения виртуального собеседника.

## 2.1 Программирование программного продукта

Каждый виртуальный собеседник имеет свои начальные шаблоны бесед с помощью которых он проходит свое первое обучение. Шаблоны так же отвечают на вопрос — для какой цели был создан данный чат-бот и что он должен говорить, чтобы удовлетворять информационные потребности пользователей.

Беседы будут храниться в отдельном файле в виде набора фраз, которые должны следовать друг за другом. Например, пользователь ввел фразу либо идентичную, либо очень похожую на выражение 1 из диалога 1. Значит ответом на него будет выражение 2 из диалога 1.

В данном случае, беседы будут храниться в файле json в формате массивов строк.

Данные conv.json:

{

"conv\_start" : [

"Привет",

"Привет!",

"Как ваши дела?",

"Прекрасно.",

"Рада это слышать",

"Спасибо.",

"Пожалуйста."

],

"conv\_end" : [

"Пока",

"До свидания",

"До встречи",

"Всего хорошего",

"Всего доброго",

"Пока",

"Досвидания",

"Доброго дня"

],

"phone1\_1" : [

"Можно узнать номер отдела информатизации?",

"Да, конечно. 89132453674"

],

"phone1\_2" : [

"Могу ли я узнать номер отдела информатизации?",

"Да, конечно. 89132453674"

],

"phone1\_3" : [

"Я хочу позвонить отделу информатизации?",

"Вот номер отдела. 89132453674"

],

"about" : [

"о министерстве",

"Да, конечно. Министр образования и науки Алтайского края: Костенко Максим Александрович. Адрес: 6043, г. Барнаул, ул. Ползунова, д.36. Приемная:(3852)298-6 00, (3852)298-601. Факс: (3852)298-659. Электронная почта: info@22edu.ru Сайт: www.educaltai.ru"

],

"address\_default" : [

"Какой адрес у министерства",

"6043, г. Барнаул, ул. Ползунова, д.36"

]

…

Диалог «about» является специфичным, так как не является полноценным. Это сделано для того, чтобы программного добавлять к выражению «о министерстве» некоторые популярные фразы, например, могу ли я узнать, можно получить информацию, я хочу узнать и т. п.

Исходный код находится в файле main.py.

Объект чат-бота создаются с помощью класса ChatBot параметрами которого являются: Имя (Petya), storage\_adapter (какое хранилище будет использоваться.), database\_uri (База данных. В данном случае SQLite) и logic\_adapters.

Далее необходимо прочитать содержимое conv.json с помощью кострукции with.

Для того, чтобы бот мог понимать больше вводных строк, необходимо добавить к первой строке в диалоге определенные шаблоны, о которых было написано выше.

# Добавление новых вариантов

\_words = [

"Могу ли я узнать",

"Могу ли я узнать информацию",

"Могу ли я узнать больше",

"Я хочу узнать информацию",

"Мне нужна информация",

"Мне хотелось бы узнать"

]

\_words2 = [

"Привет.",

"Здравствуйте.",

"Добрый день.",

"Доброе утро.",

"Добрый вечер.",

"Неподскажите."

]

temp\_dict = dict();

for key in conv :

if 'о' == conv[key][0][0] or "об" in conv[key][0] :

for i in range(len(\_words)) :

temp = copy.deepcopy(conv[key])

temp[0] = \_words[i] + " " + temp[0]

temp\_dict.update({

"%s%s" % (key, i) : temp

})

for i in range(len(\_words2)) :

for key in temp\_dict :

temp = copy.deepcopy(temp\_dict[key])

temp[0] = \_words2[i] + " " + temp[0]

conv.update({

"%s\_%s" % (key, i) : temp

})

Теперь чат-бот готов тренироваться. Для начала этого нужно обучить основам русских фраз с помощью класса ChatterBotCorpusTrainer, а далее провести обучение заданным диалогам с помощью класса ListTrainer.

Сам диалог будет строиться на бесконечном цикле из вводимых данных и ответа виртуального собеседника. Выйти из цикла возможно только тогда, когда пользователь введет фразу из диалога прощания - «conv\_end» или при системных исключениях.

# Диалог

print("Petya: Привет, чем я могу помочь?")

while True :

try :

print("You: ", end='')

response = bot.get\_response(input())

print("Petya:", response)

if str(response) in conv["conv\_end"] :

return

except (KeyboardInterrupt, EOFError, SystemExit) :

break

Для удобства проект может запускаться только в виртуальном окружении. Это было сделано для того, чтобы необходимые проекту зависимости присутствовали только в одной определенной версии. Так как активировать виртуальное окружение и запускать скрипт это довольно долго, был создан shell скрипт init.sh, который позволяет избежать этих трудностей.

## 2.2 Тестирование программного продукта

Любое программное обеспечение не лишено разнообразных ошибок, поэтому этап тестирование является наиболее важным в разработке. Целью тестирования является проверка соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов.

Тестирование будет осуществляться с помощью стандартного инструмента python — unittest. Через метод assertTrue класса TestUM будет осуществляться поиск ожидаемой подстроки в строке ответа чат-бота.

Содержимое файла test.py:

import sys

import json

import copy

import unittest

from chatterbot import ChatBot

def fun\_bot() :

bot = ChatBot(

"Petya",

storage\_adapter="chatterbot.storage.SQLStorageAdapter",

database\_uri="sqlite:///database.sqlite3",

# Логические адаптеры

logic\_adapters=[

{

"import\_path" : "chatterbot.logic.BestMatch",

'default\_response': 'Извините, но я вас не понимаю',

'maximum\_similarity\_threshold' : 0.70

}

]

)

return bot

class TestUM(unittest.TestCase):

def test\_phone(self):

self.assertTrue(

"89132453674" in str(fun\_bot().get\_response(

"Я хочу позвонить отделу информатизации"

))

)

def test\_address(self):

self.assertTrue(

"6043, г. Барнаул, ул. Ползунова, д.36" in str(fun\_bot().get\_response(

"По какому адресу находится министерство"

))

)

def test\_ministr(self):

self.assertTrue(

"Костенко Максим Александрович" in str(fun\_bot().get\_response(

"Я хочу узнать больше о министре"

))

)

def test\_email(self):

self.assertTrue(

"Электронная почта: info@22edu.ru" in str(fun\_bot().get\_response(

"Можно узнать адрес электронной почты?"

))

)

unittest.main()

Результат тестирования:

----------------------------------------------------------------------

Ran 4 tests in 6.223s

OK

# 3 ДОКУМЕНТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

Программное обеспечение способно корректно работать только с python 3.

Запуск осуществляется несколькими способами (Рассматриваются варианты на Linux).

Способ первый:

* Активировать виртуальное окружение - source venv/bin/activate;
* Перейти в каталог source командой cd source/;
* Запустить программу можно командой python3 main.py с аргументами и без. Если добавить опцию -t к команде, то чат-бот будет переобучаться исходя из данных в файле conv.json. Данный способ необходимо выполнять, только когда в файле conv.json появляются новые данные. Также можно использовать опции -t -c, которые позволят чат-боту переобучиться основам языка.
* Отключить виртуальное окружение командой deactivate.

Способ второй:

* Запустить файл init.sh в папке source командой ./init.sh. Опции работают точно так же как и в первом способе.

Взаимодействие с программой осуществляется следующим образом: пользователь вводит строку, а далее получает ответ. Это продолжается до тех пор пока в водимой строке не появятся слова прощания.

# ЗАКЛЮЧЕНИЕ

В процессе прохождения производственной практики были выполнены следующие задачи:

* Постановка задачи и проектирование программного продукта;
* Разработка программного продукта;
* Тестирование разработанного программного продукта и устранение ошибок и недочетов;
* Составление документации.

Прохождение производственной практики является неотъемлемой частью становления студента как специалиста своей специальности. Она позволяет определить уровень профессиональной и моральной подготовки студента в рабочей обстановке.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ И ИСТОЧНИКОВ

1. Анашкина, Н. В. Технологии и методы программирования / Н.В. Анашкина, Н.Н. Петухова, В.Ю. Смольянинов. - М.: Academia, 2012. - 384 c.
2. Анисимов, А. Е. Сборник заданий по основаниям программирования / А.Е.
3. Анисимов, В.В. Пупышев. - М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2012. - 352 c.
4. Богачев, К. Ю. Программирование. Основы параллельного программирования / К.Ю. Богачев. - М.: Бином. Лаборатория знаний, 2015. - 344 c.
5. Вендров, А. М. Практикум по проектированию программного обеспечения экономических информационных систем / А.М. Вендров. - М.: Финансы и статистика, 2011. - 192 c.
6. Гагарина, Л. Г. Разработка и эксплуатация автоматизированных информационных систем. Учебное пособие / Л.Г. Гагарина. - М.: Форум, Инфра-М, 2015. - 384 c.
7. Гагарина, Л. Г. Технология разработки программного обеспечения / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Виснадул. - М.: Форум, Инфра-М, 2011. - 400 c.
8. Гвоздева, В. А. Введение в специальность программиста / В.А. Гвоздева. - М.: Форум, Инфра-М, 2013. - 208 c.
9. Голицына, О. Л. Программное обеспечение / О.Л. Голицына, И.И. Попов, Т.Л. Партыка. - М.: Форум, 2013. - 448 c.
10. Гусятников, В. Н. Стандартизация и разработка программных систем / В.Н. Гусятников, А.И. Безруков. - М.: Финансы и статистика, Инфра-М, 2012. - 288 c.
11. Заботина, Н. Н. Проектирование информационных систем / Н.Н. Заботина. - М.: Дрофа, 2013. - 336 c.
12. Зыков, С. В. Основы современного программирования / С.В. Зыков. - М.: Горячая линия - Телеком, 2011. - 448 c.
13. Иванова, Г. С. Технология программирования / Г.С. Иванова. - М.: КноРус, 2011. - 336 c.
14. Информационные технологии / О.Л. Голицына и др. - М.: Форум, Инфра-М, 2015. - 608 c.
15. Котляров, В. П. Основы тестирования программного обеспечения / В.П. Котляров, Т.В. Коликова. - М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2012. - 288 c.
16. Милов, А. В. Основы программирования в задачах и примерах / А.В. Милов. - М.: Фолио, 2016. - 400 c.