

**Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ И.С. ТУРГЕНЕВА»
Институт приборостроения, автоматизации и информационных технологий
Кафедра программной инженерии**

ОТЧЕТ
по производственной практике

на материалах АНО «Центр Интернет-образования»

Студент _____Беликов П.Г.

Группа 81ПГ

Институт приборостроения, автоматизации
и информационных технологий

Направление 09.03.04 «Программная инженерия»

Руководитель практики

от университета _____Артемов А.В.

Руководитель практики

от профильной организации _____Забелин С.А.
М.П.

Оценка защиты

Орел, 2020

**Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ И.С. ТУРГЕНЕВА»**

**Институт приборостроения, автоматизации и информационных технологий
Кафедра программной инженерии**

Направление подготовки 09.03.04 «Программная инженерия»

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ
на производственную практику**

для обучающегося Беликова Павла Геннадьевича

2 курса очной формы обучения, группы 81-ПГ

Место прохождения практики: АНО «Центр Интернет-образования»

Срок прохождения практики с «07» июля 2020 г. по «20» июля 2020 г.

Содержание практики (вопросы, подлежащие изучению):

- разработать программу, которая будет распознавать человеческую речь и выполнять соответствующие действия;
- усовершенствовать программу, увеличив количество распознаваемых команд.

Планируемые результаты практики:

- полностью работающая программа, которая будет распознавать человеческую речь с микрофона и выполнять действия, соответствующие голосовой команде.

Руководитель практики
от университета

_____ Артемов А.В.

Согласовано:

Руководитель практики
от профильной организации

_____ Забелин С.А.

Задание принял: _____

«07» июля 2020г.

СОВМЕСТНЫЙ РАБОЧИЙ ГРАФИК (ПЛАН) **проведения производственной практики**

Обучающегося 2 курса очной формы обучения, группы 81-ПГ

Направление подготовки 09.03.04 «Программная инженерия»

Место прохождения практики АНО «Центр Интернет-образования»

Срок прохождения практики с «07» июля 2020 г. по «20» июля 2020 г.

№ п/п	Наименование этапа проведения практики*	Вид работ	Срок прохождения этапа практики	Форма отчетности	Отметка о выполнении
1	Подготовительный (организационный) этап	1. Организационное собрание для разъяснения целей, задач, содержания и порядка прохождения практики. 2. Инструктаж по технике безопасности. 3. Выдача индивидуального задания.	07.07.2020	Индивидуальное задание	
2	Основной этап	1. Ознакомление с конкретными видами деятельности в соответствии с положениями структурных подразделений и должностными инструкциями. 2. Сбор информации и материалов практики. 3. Выполнение программы практики, индивидуального задания на практику. 4. Обработка, систематизация и анализ фактического и теоретического материала.	08.07.2020 - 18.07.2020	Фактический и теоретический материал, результаты анализа	
3	Заключительный этап	Составление отчета по практике. Защита отчета по практике с представлением материалов конкретной профильной организации.	20.07.2020	Отчет по практике	

Руководитель практики
от университета

_____ Артемов А.В.

Руководитель практики
от профильной организации

_____ Забелин С.А.

С рабочим графиком (планом) ознакомлен:

Обучающийся

_____ Беликов П.Г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 ПОСТАНОВКА ЗАДАЧИ	6
2 МОДЕЛИРОВАНИЕ ПРОГРАММЫ.....	7
2.1 Диаграмма вариантов использования	7
2.2 Диаграмма состояний	8
2.3 Диаграмма деятельности	10
3 РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	12
3.1 Особенности программной реализации	12
3.2 Пример работы программы	14
ЗАКЛЮЧЕНИЕ	17
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	18
Приложение 1	19

ВВЕДЕНИЕ

На практике была поставлена задача разработки программы, слушающей голос пользователя с микрофона, распознающей текст, а затем выполняющей соответствующие распознанной команде действия. Целью практической работы является моделирование и реализация программы “Голосовой помощник”.

Задачи практической работы, необходимые к выполнению:

1. Выбор метода решения.
2. Разработка алгоритмов решения задачи.
3. Построение моделей для решения задачи.
4. Реализация программы.

1 ПОСТАНОВКА ЗАДАЧИ

Целью работы является проектирование и реализация программы, предназначенной для прослушивания и распознавания человеческого голоса. В зависимости от распознанной команды, программа должна выполнять соответствующий ряд действий.

Программа должна реализовывать следующие типы действий:

1. Произнесение голосового ответа.
2. Открытие/запуск сторонних файлов.
3. Чтение данных из текстовых файлов.
4. Выполнение каких-либо вычислений.
5. Работа с временем.
6. Взаимодействие с браузером.
7. Завершение своей работы.

Конкретные примеры реализации приведенных действий должны быть смоделированы в процессе работы.

Программа должна предусматривать варианты обращения к ней, а также определения неизвестной команды.

2 МОДЕЛИРОВАНИЕ ПРОГРАММЫ

Для построения моделей программы был использован онлайн ресурс Draw.io —это сервис, предназначенный для разработки диаграмм и схем. Сервис разделен на 3 части – меню, панель объектов и документ. [1]

С помощью веб-сервиса Draw.io можно создавать:

1. Диаграммы.
2. UML – модели.
3. Графики.
4. Блок-схемы.
5. Формы.
6. Другое.

Для разработки программы необходимо смоделировать 3 основные диаграммы:

1. Диаграмма вариантов использования.
2. Диаграмма состояний.
3. Диаграмма деятельности.

2.1 Диаграмма вариантов использования

Диаграмма вариантов использования — диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. [2]

Исходя из диаграммы вариантов использования, представленной на Рисунке 1, мы можем сделать вывод, что у нас есть два актёра (пользователь и голосовой помощник). Пользователь произносит голосовую команду, а помощник записывает её, переводит в строку, обрезает и выполняет соответствующую ей команду. Выполнение команды зависит от условия

обращения пользователя: он должен начать свою фразу с обращения к помощнику.

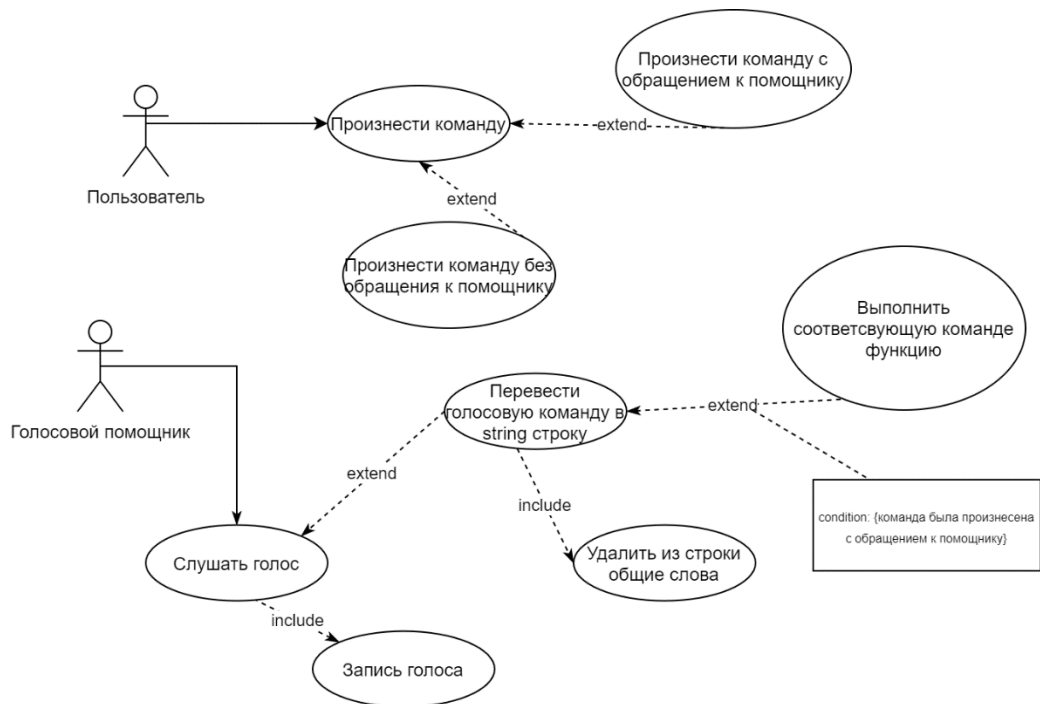


Рисунок 1 – Диаграмма вариантов использования.

2.2 Диаграмма состояний

Диаграммы состояний являются хорошо известным средством описания поведения систем. Они определяют все возможные состояния, в которых может находиться конкретный объект, а также процесс смены состояний объектов результате влияния некоторых событий. [3]

После начала работы программа приветствует пользователя в соответствии с временем суток, затем переходит в режим ожидания голосовой команды. Программа слушает звук с выбранного устройства ввода звука, пока та не будет произнесена. После программа распознает команду, и, если распознанная команда совпадает с одной из имеющихся, то программа выполняет соответствующие ей действия. Среди команд существует особая команда завершения, завершающая работу программы. Если распознанная команда не известна, то после распознавания программа снова вернется в состояние ожидания голосовой команды.



Рисунок 2 – Диаграмма состояний.

Ниже представлен полный список действий, которые может выполнить голосовой помощник:

- Сказать текущее время.
- Открыть проводник.
- Произвести вычисления на калькуляторе (только с двумя числами).
- Запустить игру (игра написана на языке Python).
- Выполнить перевод валюты (доступно RUB, EUR, USD).
- Рассказать шутку.
- Рассказать о своих возможностях.
- Открыть сайт в браузере (доступно google, firefox, youtube, vk, aliexpress, gmail, oreluniver).

- Вызвать собственные настройки для изменения.
- Включить музыку.
- Поставить секундомер.
- Выключить секундомер.
- Завершить работу.

2.3 Диаграмма деятельности

Диаграмма активностей (видов деятельности) - один из доступных видов диаграмм. Она, как и диаграмма состояний, отражает динамические аспекты поведения системы. По существу, эта диаграмма представляет собой блок-схему, которая наглядно показывает, как поток управления переходит от одной деятельности к другой. [4]

В начальном состоянии программа будет пассивно слушать звук с доступного устройства ввода. Когда пользователь произносит команду, помощник распознает её. После этого, команда проверяется на наличия слова обращения к помощнику. Если обращение отсутствует, то программа возвращается в изначальное состояние. Если обращение обнаружено, то производится следующая проверка: если распознана обычная команда, то программа выполняет соответствующие ей действия; если распознана команда завершения работы, то программа завершает свою работу; если распознанная команда не соответствует ни одной из заданных, то программа возвращается в начальное состояние.

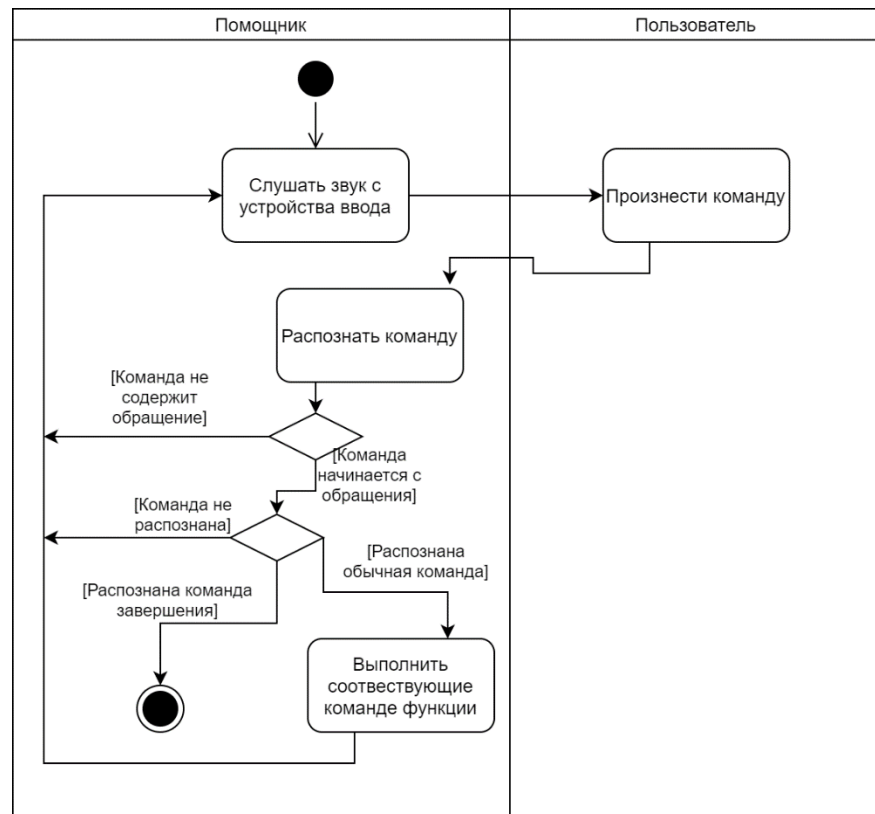


Рисунок 3 – Диаграмма деятельности.

3 РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1 Особенности программной реализации

Программа реализована на языке программирования Python в виде консольного приложения.

Чтобы голосовой ассистент начал работать, необходимо решить следующие вопросы:

1. Считывание речи с устройства ввода.
2. Распознавание речи.
3. Выполнение соответствующих команде функций.
4. Вывод голосового ответа (при необходимости).

Для считывания речи с устройства ввода используется модуль PyAudio. Он позволяет считывать речь с микрофона в реальном времени. [5]

Для распознавания речи был выбран модуль SpeechRecognition. В отличие от своего основного конкурента в лице модуля rocketsphinx, выбранный модуль лучше распознает русскую речь, а также не грузит свою БД в оперативную память компьютера, что существенно сказывается на производительности программы. Но, к сожалению, выбранный модуль требует подключения к интернету для работы. SpeechRecognition осуществляет распознавание речи через Google API. [6]

Для преобразования текста в речь используется модуль pyttsx3. В отличие от альтернативных библиотек, он работает в автономном режиме. Данный модуль позволяет использовать для вывода как стандартный голос Windows, так и многие другие, установленные дополнительно. [7]

В программе описан словарь “options”, в котором хранятся основные настройки программы: слова обращения, общие слова и слова для распознавания команд. В программе описан словарь функций “exec”, в котором по ключам расположены функции, соответствующие каждой голосовой команде

Логика работы программы реализована 4 основными функциями:

1. Функция “HelperSay” – получает в качестве аргумента строку и выводит её голосовой вариант с помощью преобразования текста в речь.

2. Функция “ListenUserVoice” – получает в качестве аргумента считанное микрофоном аудио, распознает его и преобразовывает в текст. Из текстового варианта команды удаляются слова обращения и общие слова, получая команду в чистом виде. Затем использует функции “RecognizeCommand” и “ExecuteCommand”.

3. Функция “RecognizeCommand” – получает в качестве аргумента преобразованный текст команды из функции “ListenUserVoice” и распознает, к какой функции относится данная команда.

4. Функция “ExecuteCommand” – получает в качестве аргумента от функции “RecognizeCommand” ключ команды. С помощью этого ключа она обращается к словарю “ехес”, от которого получает необходимую для исполнения функцию и запускает её.

Программа выполняет функцию “ListenUserVoice” до того момента, пока не будет произнесена команда завершения работы.

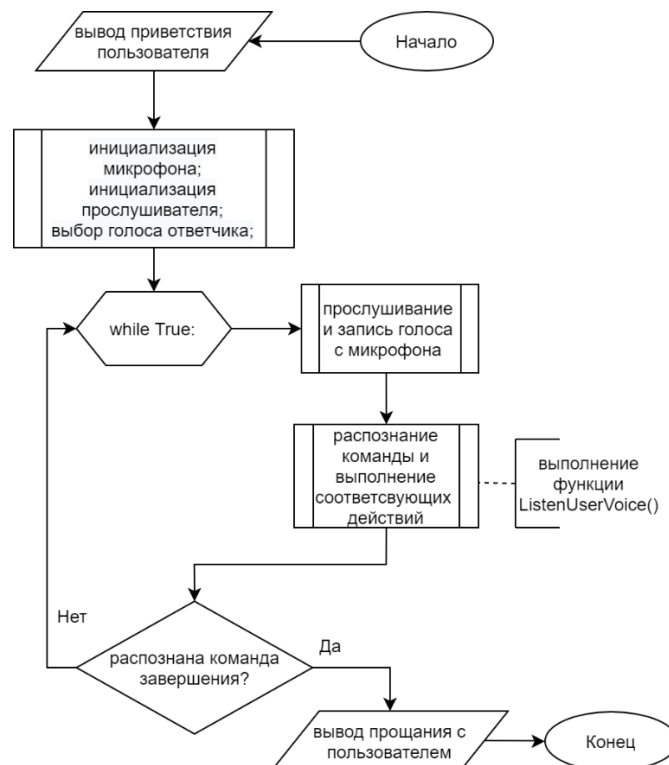


Рисунок 4 – алгоритм работы программы.

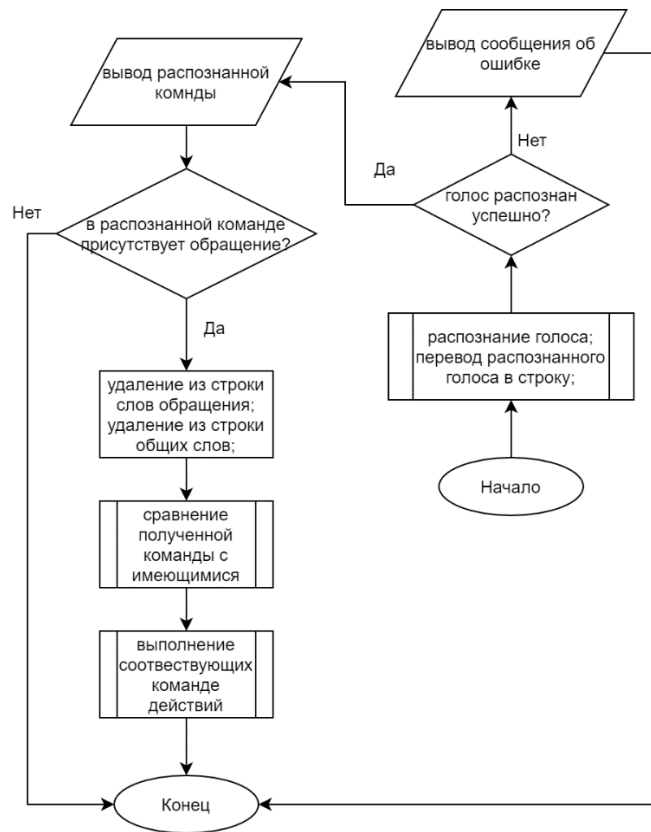


Рисунок 5 – алгоритм функции “ListenUserVoice”.

3.2 Пример работы программы

Сразу после запуска, программа приветствует пользователя.

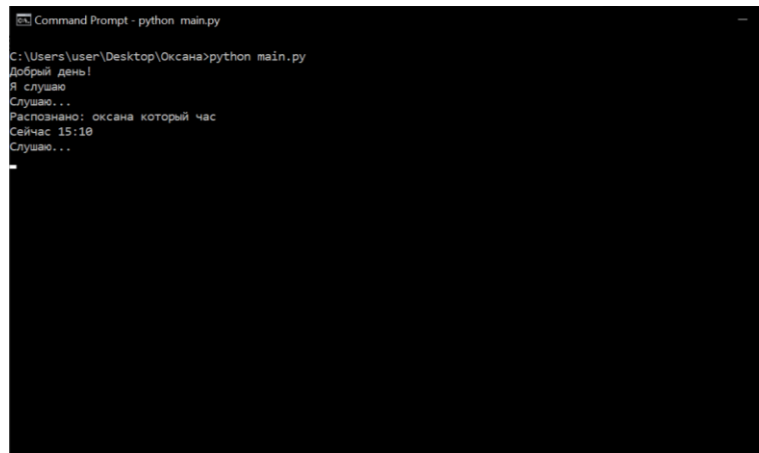
```

Command Prompt - python main.py
C:\Users\user\Desktop\Оксана>python main.py
Добрый день!
Я слушаю
Слушаю...
  
```

The screenshot shows a Windows Command Prompt window titled 'Command Prompt - python main.py'. The user has executed the command 'python main.py' from the directory 'C:\Users\user\Desktop\Оксана'. The program's output is displayed on the next lines: 'Добрый день!' (Good day!), 'Я слушаю' (I am listening), and 'Слушаю...' (Listening...).

Рисунок 6 – Начало работы.

При поступлении команды с обращением от пользователя, программа выводит распознанную речь на экран, а затем выполняет команду, выводя её результат на экран. В качестве обращения используется “Оксана”.



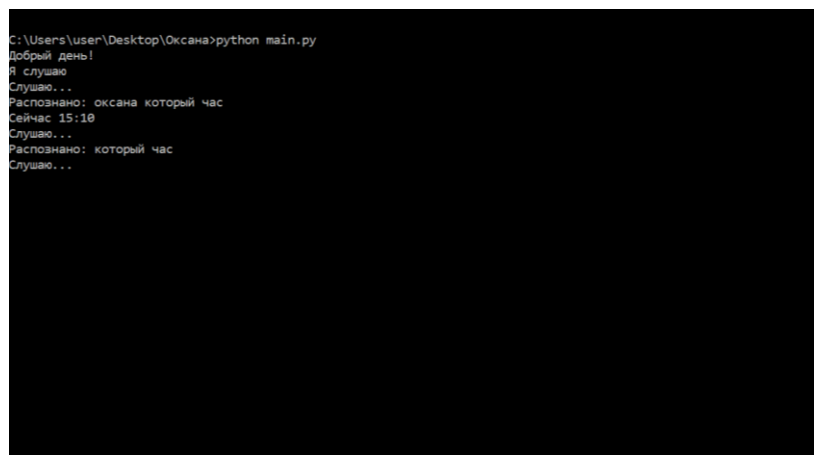
```

Command Prompt - python main.py
C:\Users\user\Desktop\Оксана>python main.py
Добрый день!
Я слушаю
Слушаю...
Распознано: оксана который час
Сейчас 15:10
Слушаю...

```

Рисунок 7 – команда от пользователя с обращением.

При поступлении команды без обращения, программа распознает её, но не выполняет никаких действий, возвращаясь в режим прослушивания.



```

C:\Users\user\Desktop\Оксана>python main.py
Добрый день!
Я слушаю
Слушаю...
Распознано: оксана который час
Сейчас 15:10
Слушаю...
Распознано: который час
Слушаю...

```

Рисунок 8 – команда от пользователя без обращения.

Ниже приведен пример команд пользователя для выполнения каждой функции программы (слова обращения в примере опущены).

- Сказать текущее время – “Который час”, “сколько времени”.
- Открыть проводник – “Открой проводник”, “Открой мой компьютер”.
- Произвести вычисления на калькуляторе (только с двумя числами)
- “Сколько будет три умножить на четыре”.

- Запустить игру (игра написана на языке Python) – “Запусти игру”.
- Выполнить перевод валюты (доступно RUB, EUR, USD) - “50 евро в рубли”.
- Рассказать шутку – “Расскажи шутку”.
- Рассказать о своих возможностях - “Расскажи о себе”.
- Открыть сайт в браузере (доступно google, firefox, youtube, vk, aliexpress, gmail, oreluniver) - “Открой vk”.
- Вызвать собственные настройки для изменения - “открой настройки”.
- Включить музыку - “Включи музыку”, “Воспроизведи музыку”.
- Поставить секундомер - “Поставь секундомер”, “Засеки время”.
- Выключить секундомер - “Останови секундомер”.
- Завершить работу. - “До свидания”, “Пока”.

ЗАКЛЮЧЕНИЕ

В ходе выполнения практической работы мы выбрали метод решения поставленной задачи, разработали алгоритмы программы, смоделировали её с помощью диаграмм UML, реализовали программу на языке программирования высокого уровня Python. Таким образом, мы достигли поставленной цели.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. StartPack [Электронный ресурс]. – Режим доступа: <https://startpack.ru/application/draw-io> (дата обращения 12.07.2020)
2. Свободная энциклопедия Википедия. Диаграмма прецедентов [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/> (дата обращения 12.07.2020).
3. ITteach. Диаграмма состояний [Электронный ресурс]. – Режим доступа: <https://itteach.ru/rational-rose/diagrammi-sostoyaniy> (дата обращения (12.07.2020)).
4. Flexberry. Диаграмма видов деятельности [Электронный ресурс]. – Режим доступа: https://flexberry.github.io/ru/fd_activity-diagram.html Дата обращения (14.07.2020).
5. PyPI. PyAudio [Электронный ресурс]. – Режим доступа: <https://pypi.org/project/PyAudio/> Дата обращения (15.07.2020)
6. PyPI. SpeechRecognition [Электронный ресурс]. – Режим доступа: <https://pypi.org/project/SpeechRecognition/> Дата обращения (15.07.2020)
7. PyPI. pyttsx3 [Электронный ресурс]. – Режим доступа: <https://pypi.org/project/pyttsx3/> Дата обращения (16.07.2020)

Листинг кода программы

main.py

```
import os
import time
import speech_recognition as sr
from fuzzywuzzy import fuzz
import pyttsx3
import datetime
import webbrowser
import random
import platform
import requests
from mutagen.mp3 import MP3
from googletrans import Translator
import threading
from currency_converter import CurrencyConverter

sites = { }

# настройки
options = {
    "alias": ('оксана', 'окс', 'окси'),
    "tbr": (
        'скажи', 'расскажи', 'покажи', 'сколько', 'произнеси', 'запусти', 'открой', 'как', 'поставь',
        'переведи', 'засеки',
        'сколько будет'),
    "cmds": {
        "ctime": ('текущее время', 'сейчас времени', 'который час', 'сколько времени'),
        "explorer": ('мой компьютер', 'проводник'),
        "calc": ('прибавить', 'умножить', 'разделить', 'степень', 'вычесть', 'поделить', 'x', '+', '-', '/'),
        "game_start": ('игрушку', 'игру'),
        "conv": ('валюта', 'конвертер', 'доллар', 'руб', 'евро'),
        "joke": ('расскажи анекдот', 'рассмеши меня', 'ты знаешь анекдоты'),
```

```

"user_help": ('что ты можешь', 'что ты умеешь'),
"internet": ("vk", "vk", "контакт", "гугл", "google", "google chrome", "хром", "youtube"
             "сайт", 'вконтакте', "ютуб",
             "алиэкспресс", "aliexpress", "wikipedia", "wiki",
             "википедия", "вики", "орел универ", "универ", "огу", "агу", "gmail"),
"user_settings": ('пользовательские настройки', 'настройки'), # не прописаны
"translator": ("переводчик", "translate"),
"music": ("музыку", "музыка", "воспроизведи музыку"),
"startStopwatch": ('запусти секундомер', "включи секундомер", "засеки время"),
"stopStopwatch": ('останови секундомер', "выключи секундомер", "останови"),
"goodbye": ('прощай', 'до свидания', 'до скорого', 'аривидерчи', 'пока')
}
}

```

```
#####funcs#####
```

```
def HelperSay(whatSpeak):
```

```

    print(whatSpeak)
    speak_engine.say(whatSpeak)
    speak_engine.runAndWait()
    speak_engine.stop()

```

```
def ListenUserVoice(recognizer, audio):
```

```

    try:
        global voice
        voice = recognizer.recognize_google(audio, language="ru-RU").lower()
        print("Распознано: " + voice)
        if voice.startswith(options["alias"]): # обращение к помощнику
            cmd = voice
            for i in options['alias']:
                cmd = cmd.replace(i, "").strip()
            for i in options['tbr']:
                cmd = cmd.replace(i, "").strip()

```

```

    cmd = RecognizeCommand(cmd)
    ExecuteCommand(cmd['cmd'])

except sr.UnknownValueError:
    pass
except sr.RequestError as e:
    print("[log] Неизвестная ошибка, проверьте интернет!")

def RecognizeCommand(cmd):
    RC = {'cmd': '', 'percent': 0}
    for c, v in options['cmds'].items():
        for x in v:
            vrt = fuzz.ratio(cmd, x)
            if vrt > RC['percent']:
                RC['cmd'] = c
                RC['percent'] = vrt
    return RC

def convertation():
    class CurrencyError(Exception):
        pass
    c = CurrencyConverter()
    money = None
    from_currency = None
    to_currency = None
    list_of_conv = voice.split()
    if len(list_of_conv) > 4:
        list_of_conv = list_of_conv[1:]
    else:
        print()
    while money is None:
        try:

```

```

    money = list_of_conv[0]
except ValueError:
    HelperSay("Скажите, к примеру: 50 долларов в рубли")
    break
while from_currency is None:
    try:
        list_of_conv[0] = int(list_of_conv[0])
    except ValueError:
        HelperSay("Скажите, к примеру: 50 долларов в рубли")
        break
    try:
        if "руб" in list_of_conv[1]:
            from_currency = "RUB"
        elif "дол" in list_of_conv[1]:
            from_currency = "USD"
        elif "евр" in list_of_conv[1]:
            from_currency = "EUR"
        if from_currency not in c.currencies:
            raise CurrencyError

    except (CurrencyError, IndexError):
        from_currency = None
        HelperSay("Скажите, например: 50 долларов в рубли")
        break

while to_currency is None:
    try:
        list_of_conv[0] = int(list_of_conv[0])
    except ValueError:
        return None
    try:
        if "руб" in list_of_conv[3]:
            to_currency = "RUB"
        elif "дол" in list_of_conv[3]:
            to_currency = "USD"

```

```

elif "евр" in list_of_conv[3]:
    to_currency = "EUR"
if to_currency not in c.currencies:
    raise CurrencyError

except (CurrencyError, IndexError):
    to_currency = None
    HelperSay("Скажите, например: 50 долларов в рубли")
    break
while True:
    try:
        HelperSay(f'{money} {from_currency} в {to_currency} - '
                  f'{round(c.convert(money, from_currency, to_currency), 2)}')
        break
    except ValueError:
        HelperSay("Скажите, например: 50 долларов в рубли")
        break

def ctime():
    now1 = datetime.datetime.now()
    HelperSay("Сейчас " + str(now1.hour) + ":" + str(now1.minute))

def explorer():
    HelperSay("Открываю проводник")
    os.system(f'start {os.path.realpath("C:/")}')

def game_start():
    HelperSay("Запускаю игру")
    os.system(f'start RacerGame.py')

def user_help():

```

```

HelperSay("Вы обращаетесь ко мне по имени а потом "
          "говорите команду и я её выполняю. Вот список того, что я могу: ")
print("Сказать время\n Рассказать анекдот\n"
      " Запустить сайт из перечисленных (Google, Firefox, Aliexpress, vk, oreluniver)\n "
      "Открыть проводник\n Посчитать на калькуляторе\n"
      " Выполнить перевод валюты (доступно RUB, USD, EUR)\n Запустить музыку из папки\n"
      " Запустить встроенную игру\n Поставить секундомер\n")

def user_settings():
    HelperSay("Открываю настройки")
    UserSettings()

def music():
    thread2 = threading.Thread(target=music_play())
    thread2.start()
    thread2.join()

def startStopwatch():
    HelperSay("Секундомер запущен")
    global startTime
    startTime = time.time()

def stopStopwatch():
    global startTime
    if startTime != 0:
        Time = time.time() - startTime
        HelperSay(f"Прошло {round(Time // 3600)} часов {round(Time // 60)} минут {round(Time % 60, 2)} секунд")
        startTime = 0
    else:
        HelperSay("Секундомер не включен")

```



```

def gb():
    HelperSay("До свиданья!")
    os.abort()

def UserSettings():
    if platform.system() == 'Windows':
        os.system('cls') # For Windows
    else:
        os.system('clear') # For Linux/OS X
    loop = True
    while loop: ## While loop which will keep going until loop = False
        print(30 * "-", "MENU", 30 * "-")
        print("1. Добавить сайт (нет)")
        print("2. Изменить голос ассистента (нет)")
        print("3. Изменить устройство ввода (нет)")
        print("4. Указать путь к пользовательской игре (нет)")
        print("5. Exit")
        print(67 * "-") ## Displays menu
        choice = input("Enter your choice [1-5]: ")

        if choice == 1:
            print("Menu 1 has been selected")
            ## Код
        elif choice == 2:
            print("Menu 2 has been selected")
            ## Код
        elif choice == 3:
            print("Menu 3 has been selected")
            ## Код
        elif choice == 4:
            print("Menu 4 has been selected")
            ## Код

```

```

elif choice == 5:
    print("Menu 5 has been selected")
    ## Код
    loop = False # This will make the while loop to end as not value of loop is set to False
else:
    # Any integer inputs other than values 1-5 we print an error message
    print("Wrong option selection. Enter any key to try again..")

```

```

def GetJocke():
    file = open('Jockes.txt', 'r', encoding='utf-8')
    for line in file:
        jockesToUser.append(line)
    HelperSay(random.choice(jockesToUser))

```

```

def music_play():
    import pygame
    DIR = 'Music/'
    filename = random.choice(os.listdir(DIR))
    f = MP3('Music/' + filename)
    song = pygame.media.load(os.path.join(DIR, filename))
    song.play()
    pygame.app.run()

```

```

def translate():
    url = 'https://translate.yandex.net/api/v1.5/tr.json/translate?'
    key =
    'trnsl.1.1.20190227T075339Z.1b02a9ab6d4a47cc.f37d50831b51374ee600fd6aa0259419fd7ecd9
7'
    text = voice.split()[1:]
    lang = 'en-ru'
    r = requests.post(url, data={'key': key, 'text': text, 'lang': lang}).json()
    try:

```

```
HelperSay(r["text"])
```

```
except:
```

```
    HelperSay("Обратитесь к переводчику, начиная со слова 'Переводчик'")
```

```
def calculator():
```

```
    try:
```

```
        list_of_nums = voice.split()
```

```
        num_1, num_2 = int((list_of_nums[-3]).strip()), int((list_of_nums[-1]).strip())
```

```
        opers = [list_of_nums[0].strip(), list_of_nums[-2].strip()]
```

```
        for i in opers:
```

```
            if 'дел' in i or 'множ' in i or 'лож' in i or 'приба' in i or 'выч' in i or i == 'x' or i == '/' or i == '+' or i == '-' or i == '*' or "степен" in i or "в степен" in i:
```

```
                oper = i
```

```
                break
```

```
            else:
```

```
                oper = opers[1]
```

```
        if oper == "+" or 'слож' in oper:
```

```
            ans = num_1 + num_2
```

```
        elif oper == "-" or 'выче' in oper:
```

```
            ans = num_1 - num_2
```

```
        elif oper == "x" or 'множ' in oper:
```

```
            ans = num_1 * num_2
```

```
        elif oper == "/" or 'дел' in oper:
```

```
            if num_2 != 0:
```

```
                ans = num_1 / num_2
```

```
            else:
```

```
                HelperSay("Делить на ноль невозможно")
```

```
        elif "степен" in oper or oper == "**":
```

```
            ans = num_1 ** num_2
```

```
        HelperSay("Будет {0}".format(ans))
```

```
    except:
```

```
        HelperSay("Скажите, например: Сколько будет 2+2?")
```

```

def browser():
    global sites
    sites = {"https://vk.com": ["vk", "vk"], 'https://www.youtube.com/': ['youtube', 'ютуб'],
            'https://ru.wikipedia.org': ["вики", "wiki"],
            'https://ru.aliexpress.com': ['али', 'ali', 'aliexpress', 'алиэспресс'],
            'http://google.com': ['гугл', 'google'],
            'https://www.amazon.com': ['амазон', 'amazon'],
            'https://www.apple.com/ru': ['apple', 'эпл'],
            'http://www.oreluniver.ru/': ['ору', 'ару', 'орел универ'],
            'https://mail.google.com/mail/u/0/': ['gmail']}

    site = voice.split()[-1]
    for k, v in sites.items():
        for i in v:
            if i not in site.lower():
                open_tab = None
            else:
                open_tab = webbrowser.open_new_tab(k)
                break
        if open_tab is not None:
            break

    exec = {
        "ctime": ctime,
        "explorer": explorer,
        "calc": calculator,
        "game_start": game_start,
        "conv": conversation,
        "joke": GetJoke,
        "user_help": user_help,
        "internet": browser,
        "user_settings": user_settings,
        "translator": translate,
        "music": music,
        "startStopwatch": startStopwatch,
        "stopStopwatch": stopStopwatch,
    }

```

```

    "goodbye": gb,
}

def ExecuteCommand(cmd):
    global exec
    if cmd not in exec.keys():
        HelperSay("Я еще так не умею.")
    else:
        exec[cmd]()

#####

jockesToUser = []
voice = "str"
startTime = 0

speak_engine = pyttsx3.init()

voices = speak_engine.getProperty('voices')
speak_engine.setProperty('voice', voices[0].id)

now = datetime.datetime.now()
if now.hour >= 6 and now.hour < 12:
    HelperSay("Доброе утро!")
elif now.hour >= 12 and now.hour < 18:
    HelperSay("Добрый день!")
elif now.hour >= 18 and now.hour < 23:
    HelperSay("Добрый вечер!")
else:
    HelperSay("Доброй ночи!")
HelperSay("Я слушаю")

# Запуск программы

```

```

if platform.system() == 'Linux':
    chrome_path = '/usr/bin/google-chrome %s'
elif platform.system() == 'Windows':
    chrome_path = 'C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s'

r = sr.Recognizer()
m = sr.Microphone(device_index=0)

while True:
    with m as source:
        print("Слушаю...")
        r.adjust_for_ambient_noise(source, duration=0.1)
        audio = r.listen(source)
        ListenUserVoice(r, audio)

```

RacerGame.py

```

import pygame
import time
import random

pygame.init()

WidthDisplay = 800
HeightDisplay = 600
carWidth = 60
gameDisplay = pygame.display.set_mode((WidthDisplay, HeightDisplay))
pygame.display.set_caption('Racer')

clock = pygame.time.Clock()
crashed = False
carImage = pygame.image.load('Car.png')

def things_dodged(count):

```

```

font = pygame.font.Font('17668.otf', 20)
text = font.render("Dodged: "+str(count), True, (0, 0, 0))
gameDisplay.blit(text,(0,0))

def things(thingX, thingY, thingW, thingH, color):
    pygame.draw.rect(gameDisplay, color, [thingX, thingY, thingW, thingH])

def car(x,y):
    gameDisplay.blit(carImage,(x,y))

def text_objects(text, font):
    textSurface = font.render(text, True, (181, 20, 20))
    return textSurface, textSurface.get_rect()

def message_display(text):
    largeText = pygame.font.Font('17666.otf',115)
    TextSurf, TextRect = text_objects(text, largeText)
    TextRect.center = ((WidthDisplay / 2),(HeightDisplay / 2))
    gameDisplay.blit(TextSurf, TextRect)

    pygame.display.update()

    time.sleep(2)

    gameLoop()

def crash():
    message_display('You crashed!')

def gameLoop():
    x = WidthDisplay * 0.45
    y = HeightDisplay * 0.8
    gameExit = False
    xChange = 0

```

```

thing_startx = random.randrange(0, WidthDisplay)
thing_starty = -600
thing_speed = 7
thing_width = 100
thing_height = 100
dodged = 0

while not gameExit:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            gameExit = True
#####
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                xChange = -5
            elif event.key == pygame.K_RIGHT:
                xChange = 5
            if event.type == pygame.KEYUP:
                if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
                    xChange = 0
#####
            x += xChange
#####
            #there is our environment
            gameDisplay.fill((255, 255, 255))

            things(thing_startx, thing_starty, thing_width, thing_height, (0, 0, 0))
            thing_starty += thing_speed
            car(x, y)
            things_dodged(dodged)

            if x > WidthDisplay - carWidth:
                x = WidthDisplay - carWidth

            if x < 0:

```



```
x = 0
```

```
if thing_starty > HeightDisplay:
```

```
    thing_starty = 0 - thing_height
```

```
    thing_startx = random.randrange(0, WidthDisplay)
```

```
    dodged += 1
```

```
    thing_speed += 1
```

```
    thing_width += (dodged * 1.2)
```

```
if y < thing_starty + thing_height:
```

```
    if x > thing_startx and x < thing_startx + thing_width or x + carWidth >= thing_startx and
    x + carWidth <= thing_startx + thing_width: #ERROR! FIX IT!
```

```
        crash()
```

```
pygame.display.update()
```

```
clock.tick(60)
```

```
gameLoop()
```

```
pygame.quit()
```

```
quit()
```

**Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ И.С. ТУРГЕНЕВА»**

Институт приборостроения, автоматизации и информационных технологий

ДНЕВНИК

производственной практики студента

Беликов Павел Геннадьевич

Курс 2

Группа 81ПГ

Место прохождения практики АНО «Центр Интернет-образования»

Руководитель практики
от университета

_____ Артемов А.В.

Руководитель практики
от профильной организации

_____ Забелин С.А.
М.П.

Начало практики

«07» июля 2020 года

Окончание практики

«20» июля 2020 года

График
прохождения практики

№ п/п	Дата	Работа, выполненная студентом	Примечание
1	07.07.2020	Получение индивидуального задания.	
2	08.07.2020	Анализ задачи.	
3	09.07.2020	Анализ задачи.	
4	10.07.2020	Анализ задачи.	
5	11.07.2020	Разработка диаграммы вариантов использования.	
6	13.07.2020	Разработка диаграммы состояний.	
7	14.07.2020	Разработка диаграммы деятельности.	
8	15.07.2020	Разработка стартового варианта программы и тестирование.	
9	16.07.2020	Разработка расширенного варианта программы.	
10	17.07.2020	Отладка и исправление ошибок.	
11	18.07.2020	Оптимизация программы.	
12	20.07.2020	Составление отчета по практике. Защита отчета по практике с предоставлением материалов профильной организации.	

Руководитель практики
от университета

_____ Артемов А.В.

Руководитель практики
от профильной организации

_____ Забелин С.А.

Отзыв руководителя практики от профильной организации

Студент Беликов Павел Геннадьевич

Направление 09.03.04 «Программная инженерия»

Место прохождения практики: АНО «Центр Интернет-образования»

Сроки проведения практики: с «07» июля 2020 г. по «20» июля 2020 г.

Отчет по практике соответствует всем требованиям поставленного студенту задания. Все вопросы разрешены и освещены в полной степени. Отношение практиканта к выполняемой работе хорошее, степень выполнения поручений отличная, подготовленность студента к самостоятельному выполнению отдельных заданий хорошая. Грамотность изложения отчета по практике соответствует требованиям, объем и степень использования отечественной и иностранной литературы и материалов, справочников достаточный.

Дисциплина студента во время практики удовлетворительная. Случаев нарушения трудовой дисциплины не было. Наличие отрицательных черт, действий, проявлений, характеризующих студента с негативной стороны в период прохождения практики не выявлено.

Оценка выполнения программы практики и индивидуального задания в целом: зачтено. Все пункты программы выполнены.

Руководитель практики от АНО «Центр Интернет-образования»

инженер-программист, Забелин С.А.

«20» июля 2020г.

М.П.