

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН  
Некоммерческое акционерное общество  
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ  
им. ГУМАРБЕКА ДАУКЕЕВА»  
Кафедра IT – инжиниринг

«ДОПУЩЕНА К ЗАЩИТЕ»  
Зав. Кафедрой, PhD, доцент Досжанова А.А  
\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2020

**ДИПЛОМНЫЙ ПРОЕКТ**

На тему: Разработка системы распознавания казахской речи  
Специальность 5B070400 – «Вычислительная техника и программное  
обеспечение»

Выполнила: Абдрахым Э.Б.                      Группа: ВТ-16-2  
Научный руководитель: д.т.н., профессор Казиев Г.З.

Консультанты:

по экономической части: к.э.н., профессор Габелашвили К.Р.  
(учёная степень, звание, Ф.И.О.)  
\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2020 г.

по безопасности жизнедеятельности: доцент Приходько Н.Г.  
(учёная степень, звание, Ф.И.О.)  
\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2020 г.

по программному обеспечению: ст.преп. Майкотов М.Н.  
(учёная степень, звание, Ф.И.О.)  
\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2020 г.

Нормоконтролер: ст.преп. Абсатарова Б.Р.  
(учёная степень, звание, Ф.И.О.)  
\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2020 г.

Рецензент: \_\_\_\_\_  
(учёная степень, звание, Ф.И.О.)  
\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2020 г.

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН  
Некоммерческое акционерное общество  
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ  
им. ГУМАРБЕКА ДАУКЕЕВА»

Институт систем управления и информационных технологий

Специальность 5В070400 – «Вычислительная техника и программное обеспечение»

Кафедра IT-инжиниринг

**ЗАДАНИЕ**

на выполнение дипломного проекта

Студенту Абдрахым Эленоре Бакытжановне

Тема проекта: Разработка системы распознавания казахской речи

Утверждена приказом по университету № \_\_\_ от «\_\_\_» \_\_\_\_\_ 2020 г.

Срок сдачи законченного проекта «\_\_\_» \_\_\_\_\_ 2020 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): Python – язык программирования, PyCharm – интегрированная среда разработки.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:

- а) изучение системы распознавания речи;
- б) разработка графической схемы;
- в) программная реализация;
- г) экономическая эффективность проекта;
- д) вопросы безопасности жизнедеятельности и охраны труда.

Перечень графического материала (с точным указанием обязательных чертежей): представлены 14 таблиц, 37 иллюстрации.

Основная рекомендуемая литература:

- 1 Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.:Символ-Плюс, 2011. – 992 с.
- 2 Джоши П. Искусственный интеллект на Python, 2019г.

Консультация по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Габелашвили К.Р.	21.04.2020	
Безопасности жизнедеятельности	Приходько Н.Г.	04.05.2020	
Программная часть	Майкотов М.Н.	15.05.2020	
Нормоконтролер	Абсатарова Б.Р.	13.05.2020 – 18.05.2020	

ГРАФИК  
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечания
Изучение системы распознавания речи	13.01.2020 – 23.01.2020	
Разработка графической схемы	24.01.2020 – 03.02.2020	
Программная реализация	04.02.2020 – 20.04.2020	

Дата выдачи задания « \_\_\_\_ » \_\_\_\_\_ 2020 г.

Заведующий кафедрой \_\_\_\_\_ А.А.Досжанова  
(подпись)

Научный руководитель проекта \_\_\_\_\_ Г.З. Казиев  
(подпись)

Задание принял к исполнению студент \_\_\_\_\_ Э.Б.Абдрахым  
(подпись)

## Аңдатпа

Дипломдық жобаның тақырыбы: «Қазақ тілін тану жүйесін құру».

Бұл дипломдық жоба қазақ тілін, атап айтқанда дауыстық көмекшіні тану жүйесін дамытуды көздейді.

Дипломдық жоба PyCharm кешенді даму ортасында жүзеге асырылды. Әзірленген дипломдық жобаның мақсатына жету үшін Python бағдарламалаудың жоғары деңгейлі жалпы мақсаттағы бағдарламасы қолданылды.

Бағдарламаны әзірлеуге жұмсалған шығындар мен шығындардың экономикалық есебі, әзірленген жобаның экономикалық мақсаттылығы бағаланды.

«Тіршілік қауіпсіздігі» бөлімінде дербес компьютердің қорғаныс жерге тұйықталуы және өрт қаупі туралы есеп жасалды.

## **Аннотация**

Тема дипломного проекта: «Разработка системы распознавания казахской речи».

В данном дипломном проекте реализована разработка системы распознавания казахской речи, а именно, голосового помощника.

Дипломный проект был выполнен в интегрированной среде разработки PyCharm. Для достижения цели разрабатываемого дипломного проекта был применен высокоуровневый язык программирования общего назначения Python.

Также был проведен экономический расчет затрат и стоимости разработки программы, оценка экономической целесообразности разрабатываемого проекта.

В разделе «Безопасность жизнедеятельности» был произведен расчет защитного зануления ПЭВМ и пожароопасности.

## **Annotation**

The theme of the thesis project: “Development of a system Kazakh speech recognition system”.

This diploma project implements the development of a system for recognizing Kazakh speech, namely, a voice assistant. The graduation project was carried out in the PyCharm integrated development environment. To achieve the goal of the developed graduation project, a high-level general-purpose programming language Python was used.

An economic calculation of the costs and costs of developing the program, an assessment of the economic feasibility of the developed project was also carried out.

In the section "Life Safety" a calculation was made of the protective grounding of the personal computer and fire hazard.

## Содержание

Введение	8
1 Аналитическая часть	9
1.1 Голосовые интерфейсы, компоненты, вилы и задачи решаемые разрабатываемой системой	9
1.2 Исторический обзор системы распознавания речи	11
1.3 Степень разработанности проблемы	14
1.4 Выбор и обоснование средства разработки	14
2 Программная часть	18
3 Экспериментальная часть	28
3.1 Назначение программы	28
3.2 Установка и настройка программы	32
4 Техничко-экономическое обоснование проекта	40
4.1 Определение объема и трудоемкости разработки программного обеспечения	40
4.2 Расчет затрат на разработку информационной технологий	40
4.3 Сравнительный анализ эксплуатационных затрат до и после внедрения	44
5 Безопасность жизнедеятельности	49
5.1 Анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал	49
5.1.1 Электромагнитные поля	51
5.1.2 Правила техники безопасности и гигиены при работе на ПК	52
5.2 Расчет показателей, удовлетворяющих требованиям техники безопасности и охраны труда при работе с проектируемой системой	55
5.2.1 Пожароопасность	55
5.2.2 Электробезопасность. Расчет защитного зануления ПЭВМ	56

Заключение			
61			
Список			литературы
62			
Приложение	А	Листинг	программы
63			
Приложение Б Акт внедрения			71

## Введение

Сегодняшние речевые технологии (распознавания и синтез речи) шагнули вперед большими темпами за последние десятки лет, начиная с простых дикторо-зависимых приложений до дикторо-независимых систем автоматической транскрипции новостей, телефонных разговоров, лекций и прочих. Речевые технологии встраиваются в операционные системы или используются отдельно для организации голосового управления компьютером или озвучивания текстов в телекоммуникационных компаниях, интернет-браузерах, мобильных и бытовых устройствах. Однако эти разработки в основном ориентированы на языки развитых стран (прежде всего, английский, японский, русский, китайский), что оставляет казахский язык в стороне от этих достижений компьютерной лингвистики.

**Актуальность темы.** Казахский язык является государственным языком Республики Казахстан и, помимо этого, является родным для казахов, проживающих в разных странах мира. Казахстан, интегрируясь в мировое сообщество, должен обеспечивать достаточный уровень развития государственного языка. С другой стороны, при предоставлении казахоязычному населению тех или иных электронных услуг в рамках электронного правительства необходима организация речевого пользовательского интерфейса на казахском языке. Эти факторы обосновывают актуальность темы проекта.

**Цель дипломного проекта** провести исследование существующих систем распознавания речи казахского языка и создать свою систему для тестирования.

**Объектом исследования** дипломной работы является система распознавания казахского языка, а именно, голосовой помощник.

**Научная новизна** проекта заключается в том, что на рынке мобильных приложений подобных проектов очень мало. В магазинах приложений для мобильных устройств, а именно: Google Play, Apple App Store, Amazon Appstore и др. не найдены приложения, распознающие казахскую речь.

## **1 Аналитическая часть**

### **1.1 Голосовые интерфейсы, компоненты, виды и задачи системы распознавания речи**

Основной задачей дипломной работы является разработка системы распознавания казахской речи. В данной дипломной работе используются последние технологии в сфере разработки веб-приложения, которые позволяют создать масштабируемое, безопасное и гибкое приложения с минимальными затратами.

Требования при разработке информационной системы (приложения):

- приложение должно обладать удобным и понятным интерфейсом для пользователей;
- исходный код приложения должен комментироваться на этапе разработки для будущих изменений или дополнений.

Должен выполнять следующие команды:

- осуществлять поиск в интернете по запросу;
- открывать внешние приложения;
- выводить на экран список доступных команд;
- переводить слова;
- преобразовывать речь в текст;
- выводить и очищать историю запросов.

Интерфейс - способ взаимодействия компьютерной системы (программы) с пользователями и устройствами.

На основе интерфейса реализуется взаимодействие всех современных информационных систем. Под интерфейсом понимается набор средств, правил и методов, за счет которых осуществляется коммуникация между элементами системы, различными программами и устройствами[1].

Преимущества голосового интерфейса:

- оперативность и естественность;
- минимум специальной подготовки пользователя;
- возможность управления объектами в темноте, за пределами его визуальной видимости (в частности, с использованием существующей телефонной сети);
- возможность использования одновременно ручного ввода (с помощью клавиатуры) и голосового ввода информации;
- обеспечение мобильности оператора при управлении.

К основным классам задач голосового интерфейса следует отнести:

Синтез речи - эта задача включает в себя комплекс подзадач и заключается в обеспечении возможности произнесение речи компьютером на основе произвольного орфографического текста;

Анализ и распознавание речи - комплекс задач, включающих запись, оцифровку и анализ речи для распознавания полученного речевого сообщения компьютерной системой;

Понимание (интерпретация) речи - это комплекс задач, связанных с анализом смысла речевых сообщений и формированием реакции (ответа) компьютерной системы;

Распознавание голоса - комплекс задач, включающих анализ особенностей голоса говорящего с целью выявления каких-либо его индивидуальных (личностных) особенностей и качеств;

Компьютерное клонирование голоса и дикции - это создание близкой копии, но не биологической, а компьютерной, и не всего существа в целом (в данном случае человека), а только одной из его интеллектуальных функций: чтение произвольного орфографического текста.

Общая структура голосового интерфейса включает два основных компонента:

- синтез речи;
- распознавание речи.

Синтез речи по правилам с использованием предварительно запомненных отрезков естественного языка – это разновидность синтеза речи по правилам, которая получила распространение в связи с появлением возможностей манипулирования речевым сигналом в оцифрованной форме.

Процесс синтеза речи выглядит следующим образом. Уровни выходов нейронов эффекторного слоя нейросети при помощи карты эффекторов преобразуются в значения выбранных параметров модели синтеза. Карта эффекторов определяет соответствие между каждым нейроном эффекторного слоя и конкретным параметром модели синтеза, а также предельные значения каждого параметра. Число эффекторов и число параметров модели может не совпадать. Если параметру не соответствует ни один эффектор, используется некоторое фиксированное значение (значение по умолчанию) [2].

Каждая из задач голосового интерфейса является достаточно сложной, то в соответствие указанным компонентам ставятся два отдельных класса систем:

- системы синтеза речи;
- системы распознавания речи.

Реализация речевого диалога происходит посредством диалога, при котором запрос и ответ со стороны пользователя ведется на языке, близком к естественному. Пользователь свободно формулирует задачу, но с набором установленных программной средой слов, фраз и синтаксиса языка. Разновидностью интерактивного естественного диалога является речевое общение с компьютерной системой. В этом случае человеческий голос может преобразовываться, например, в текст, или использоваться для интерактивного управления системой, или для идентификации личности. В основе данных процессов лежит технология и решение задачи распознавания речи.

Речь в физическом смысле - это акустический сигнал, генерируемый артикуляционными органами человека, передающийся через физическую среду, воспринимаемый ухом человека. При естественной или искусственной

генерации речи в акустическом сигнале изменяются физические параметры. Эти изменения воздействуют на мембрану уха, создают траектории звуковых образов, понимаемых человеком как соответствующие звуки данного языка, или иначе говоря, при произнесении слов человек генерирует звуки (фонемы), которые несут информацию о тех символах, с помощью которых эти слова могут быть записаны в виде текста. Математическую модель генерации звука можно представить в виде возбуждающих генераторов тонового и белого шума, группы резонаторов, модуляторов и ключей (рот, нос, язык, губы), обеспечивающих формирование ощущения определенного звука.

Системы распознавания речи - это системы, анализирующие акустический сигнал алгоритмами, основанными на разнообразных теориях, предполагающих, какие характеристики речевого сигнала создают ощущения звуков данного языка, и математических методах, с той или иной точностью выделяющих значащие параметры акустического сигнала и преобразующие его в различной полноте в необходимую форму.

Системы распознавания речи состоят из двух частей - акустической и лингвистической. В общем случае могут включать в себя фонетическую, фонологическую, морфологическую, лексическую, синтаксическую и семантическую модели языка.

Акустическая - отвечает за представление речевого сигнала, за его преобразование в некоторую форму, в которой в более явном виде присутствует информация в содержании речевого сообщения.

Лингвистическая - интерпретирует информацию, получаемую от акустической модели, и отвечает за представление результата распознавания потребителю.

Задачи распознавания речи - автоматическое восстановление текста произносимых человеком слов, фраз или предложений на естественном языке и проблемы идентификации, шумоочистки, распознавания языков, оценки психофизического состояния человека. При решении задачи распознавания слитной речи человек применяет свои знания о естественном языке, а также смысл произносимого для устранения неоднозначности при восстановлении текста предложения.

## **1.2 Исторический обзор системы распознавания речи**

Попытки научить компьютеры общаться с людьми при помощи естественного голосового интерфейса предпринимались с первых лет истории компьютерной техники.

Рассмотрим мировую историю развития области распознавания речи и голосовых интерфейсов.

Технология распознавания голоса была разработана задолго до того, как Apple Siri дебютировала в 2011 году. На Всемирной выставке в Сиэтле в 1962 году IBM представила инструмент под названием Shoebox. Он был размером с обувную коробку и мог выполнять математические функции и распознавать

16 произнесенных слов, а также цифры 0-9. Shoebox представлен на рисунке 1.1.



Рисунок 1.1 – Shoebox

В 1970-х годах ученые из Университета Карнеги-Меллона в Питтсбурге, штат Пенсильвания, при существенной поддержке Министерства обороны США и его Агентства перспективных исследований в области обороны (DARPA) - создали Narгу. Он мог распознать 1 011 слов.

Как только организации придумали изобретения, способные распознавать последовательности слов, компании начали создавать приложения для этой технологии. Кукла Джули из игрушечной компании Worlds of Wonder появилась в 1987 году и могла распознавать голос ребенка и реагировать на него.

В течение 1990-х годов такие компании, как IBM, Apple и другие, создавали продукты, использующие распознавание голоса. Apple начала встраивать функции распознавания речи в свои компьютеры Macintosh с помощью PlainTalk в 1993 году. В апреле 1997 года компания Dragon выпустила Dragon NaturallySpeaking, который был первым продуктом непрерывной диктовки. Он может понимать около 100 слов в минуту и превращать его в текст. Устройства для медицинской диктовки были одними из первых, кто внедрил технологию распознавания голоса.

Siri от Apple стала первым цифровым виртуальным помощником, который стал стандартным на смартфонах Apple, когда 4 октября 2011 года вышел iPhone 4s. Siri перешла в мир интеллектуальных динамиков, когда HomePod дебютировал в феврале 2018 года.

Затем последовал Google Now (который стал Google Assistant) на платформе Android. Он также работает на iOS от Apple, но имеет ограниченную функциональность. Затем появились умные динамики, "Alexa", "Hey Google". Они стали частью многих домашних распознавателей. Google

Assistant - это часть Google Home. Раньше Facebook имел Facebook M, но его использование в приложении Facebook Messenger закончилось в январе 2018 года. Facebook Messenger представлен на рисунке 1.2.

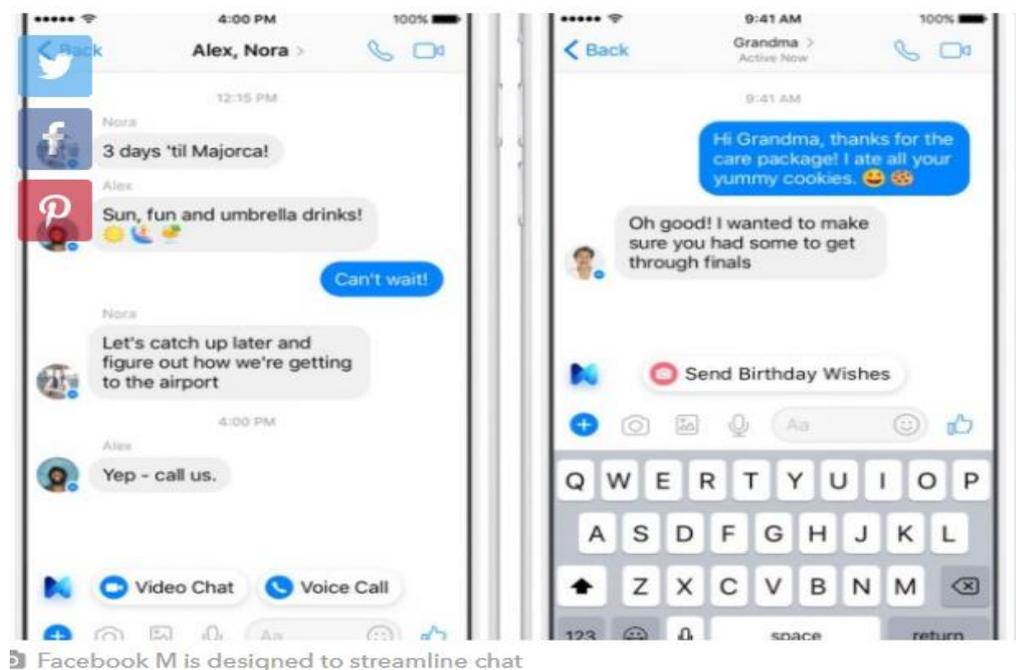


Рисунок 1.2 - Facebook Messenger

Исследование проблемы распознавания речи уже более 50 лет занимаются специалисты нескольких научных областей. Методы и алгоритмы, которые используются могут быть разделены на четыре больших класса:

- методы дискриминантного анализа, которые основаны на Байесской дискриминации;
- скрытые модели Маркова;
- искусственные нейронные сети;
- динамическое программирование, также их называют временными динамическими алгоритмами.

Следует отметить ряд преимуществ, к которым стремятся при разработке систем распознавания речи:

а) непрерывная речь - возможность, позволяющая пользователям говорить естественно (непрерывно), не делая паузы между словами (дискретный ввод речи);

б) большие словари - способность обрабатывать большое количество слов как общей, так и специальной категории из технических и предметных областей знаний с целью увеличения мощности и эффективности систем распознавания голоса;

в) независимость от диктора – способность системы распознать слова без персональной настройки компьютера путем повторения одного и того же

речевого сигнала. Для распознавания выбирают различные базовые единицы: фонемы, аллофоны, дифоны и трифоны и.т.д. Для распознавания отдельных слов все же эффективны временные динамические алгоритмы.

### **1.3 Степень разработанности проблемы**

В настоящее время вопросами познания казахского языка в нашей стране занимаются ученые Евразийского национального университета им.Л. Н. Гумилева Шарипбай, Г. Т. Бекманова, У. А. Тукеев, Д. Р. Рахимова, ученые Института информационных и вычислительных технологий КН МОН РК Амиргалиев Е. Н., Р. Р. Мусабаев и др.

А. Шарипбай со своими учениками представил математическую теорию, основывающуюся на закономерностях формальной фонетики казахского языка, морфологический и синтаксический режим, синтез и анализ слов, алгоритм и приложение речевого распознавания.

В 2018 году в Казахстане разработали свою версию голосового помощника - Айбота. Программа общается только на русском языке, но по заверениям разработчиков, Айбота должна была заговорить на казахском. Это один из проектов программы «Method PRO». «Method PRO» - ежегодная летняя стажировка, на которой в течение 2 месяцев юные программисты имеют возможность получить профессиональное обучение от лучших специалистов индустрии.

Но в магазинах приложений для мобильных устройств, а именно: Google Play, Apple App Store, Amazon Appstore и др. не найдены приложения, распознающие казахскую речь.

### **1.4 Выбор и обоснование средства разработки**

В качестве инструмента для создания голосового помощника была выбрана интегрированная среда разработки PyCharm и высокоуровневый язык программирования общего назначения - Python.

Python - это популярный язык программирования. Он был создан Гвидо ван Россумом и выпущен в 1991 году.

Возможности Python :

- python можно использовать на сервере для создания веб-приложений;
- python можно использовать вместе с программным обеспечением для создания рабочих процессов;
- python может подключаться к системам баз данных. Он также может читать и изменять файлы;
- python можно использовать для обработки больших данных и для выполнения сложных математических операций;
- python можно использовать для быстрого прототипирования или для разработки готового к производству программного обеспечения[3].

Сайт DOU.UA представил результаты собственного опроса по языкам программирования (в этот раз собрали и обработали 9747 анкет). По

результатам опроса можно посмотреть, что Python является популярным языком программирования [4].

Ниже приведены результаты опроса рейтинга языков программирования на 2020 год:

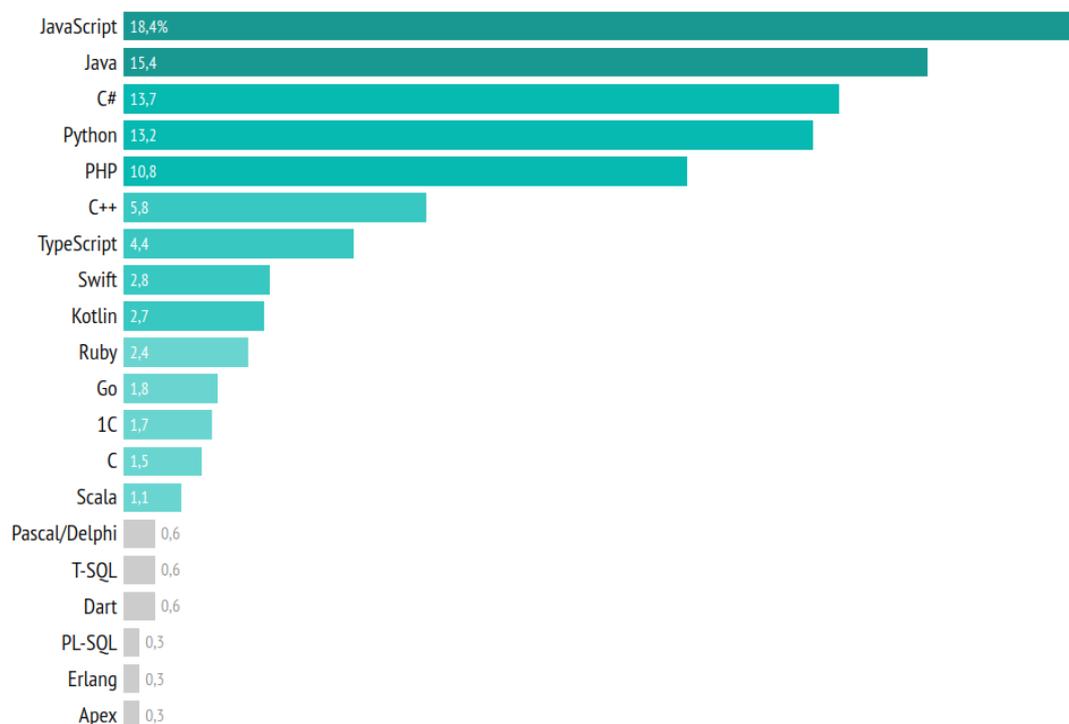


Рисунок 1.3 - Коммерческое использование

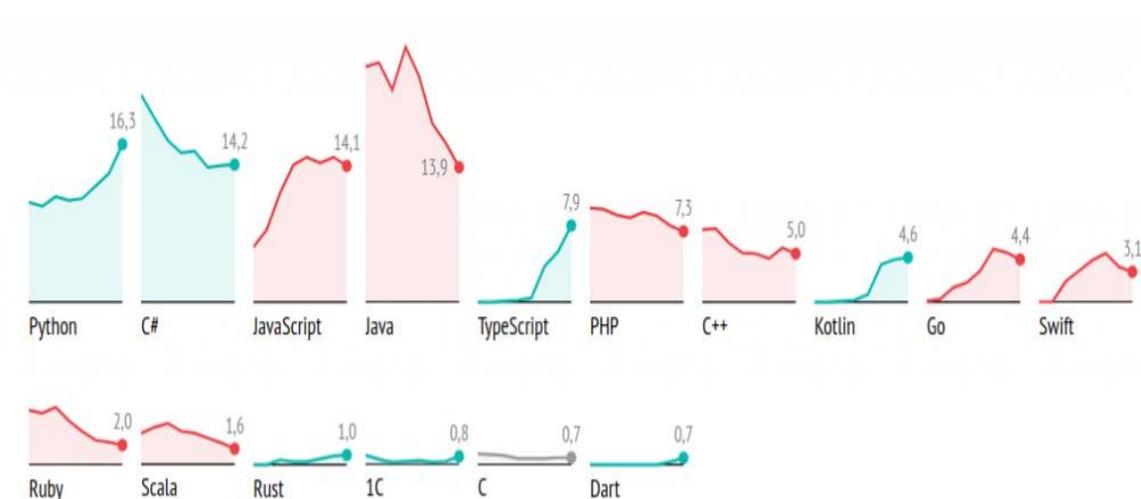


Рисунок 1.4 - Личные предпочтения

На рисунке 1.4 можно увидеть, что сейчас фаворитом является Python, далее идут почти на одном уровне C#, JavaScript и Java, а дальше - TypeScript.

На рисунке 1.5 можно увидеть, что 20,6 % анкетированных изучают или же планируют изучить Python. По этим категориям вопросов Python занимает первое место.

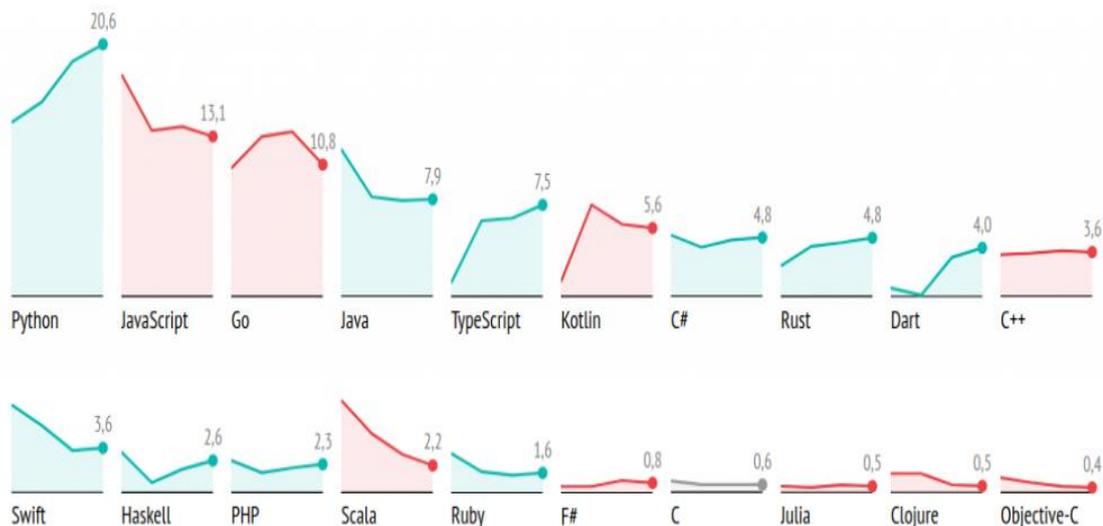


Рисунок 1.5 - Изучение новых языков

В соответствии с рисунком 1.6 можно сказать, что многие пользователи пишут свои проекты на языке Python.



Рисунок 1.6 - Свои проекты

Судя по результатам опросов, можно сказать, что на сегодняшний день высокоуровневый язык программирования Python, является современным и популярным языком.

PyCharm от разработчика JetBrains inc – интегрированная среда разработки для Python. Полный набор инструментов позволяет анализировать

код, запускать графический отладчик и юнит-тесты. Кроме того, IDE обеспечивает возможность для профессионального создания веб-страниц на Django. PyCharm является передовой средой разработки, ориентированной на повышение производительности работы программиста за счет сокращения времени написания кода.

## 2 Программная часть

Система распознавания речи, как правило, состоит из следующих частей:

- запись сообщения от абонента;
- распознавание голоса и прием текстовых данных от сервиса;
- анализ полученной информации и выполнение необходимых действий.

При создании голосового помощника использовала технологию распознавания голоса от компании Google. Google Cloud Speech API позволяет преобразовывать аудио информацию в текст, применяя модели нейронных сетей в простой для использования API. Google Cloud Speech API поддерживает более 110 языков (в том числе казахский). Использование нейронных сетей позволяет быстро преобразовывать голосовую модель в текст. На рисунке 2.1 представлена блок-схема подсистемы выполнения команд.

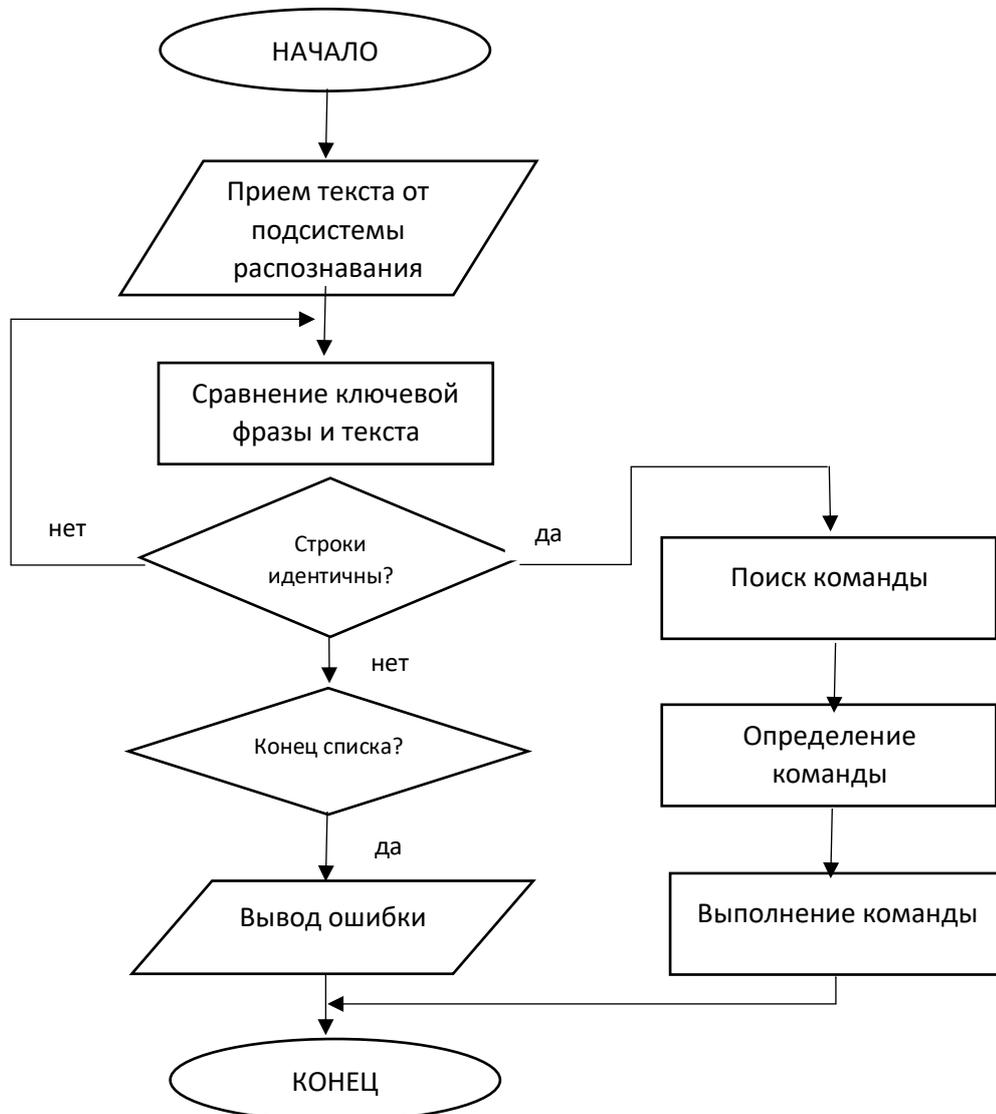


Рисунок 2.1 - Блок-схема подсистемы выполнения команд

Рассмотрим основные библиотеки, которые применялись при создании проекта.

PyAudio обеспечивает привязки Python для PortAudio. Эта библиотека используется для аудио ввода-вывода. С помощью PyAudio мы можем легко использовать Python для воспроизведения и записи звука на различных платформах. На рисунке 2.2 представлен фрагмент кода реализации библиотеки PyAudio.

```
14
15
16 # no stupidity allowed
17 if not (input or output):
18     raise ValueError("Must specify an input or output " + "stream.")
19
20 # remember parent
21 self._parent = PA_manager
22
23 # remember if we are an: input, output (or both)
24 self._is_input = input
25 self._is_output = output
26
27 # are we running?
28 self._is_running = start
29
30 # remember some parameters
31 self._rate = rate
32 self._channels = channels
33 self._format = format
34 self._frames_per_buffer = frames_per_buffer
35
36 arguments = {
37     'rate' : rate,
38     'channels' : channels,
39     'format' : format,
40     'input' : input,
41     'output' : output,
42     'input_device_index' : input_device_index,
43     'output_device_index' : output_device_index,
44     'frames_per_buffer' : frames_per_buffer}
45
46 if input_host_api_specific_stream_info:
47     _l = input_host_api_specific_stream_info
48     arguments[
49         'input_host_api_specific_stream_info'
50     ] = _l._get_host_api_stream_object()
51
52 if output_host_api_specific_stream_info:
53     _l = output_host_api_specific_stream_info
54     arguments[
55         'output_host_api_specific_stream_info'
56     ] = _l._get_host_api_stream_object()
57
58 if stream_callback:
59     arguments['stream_callback'] = stream_callback
60
```

Рисунок 2.2 - Фрагмент кода PyAudio

В режиме обратного вызова PyAudio вызывает указанную функцию обратного вызова тогда, когда ему для воспроизведения нужны новые аудиоданные и когда имеются новые записанные аудиоданные. PyAudio вызывает функцию обратного вызова в отдельном потоке. Функция имеет следующий callback(<input\_data>, <frame\_count>, <time\_info>, <status\_flag>).

Pygame не входит в стандартную библиотеку Python, то есть не поставляется с установочным пакетом, а требует отдельной установки.

Ниже приведен фрагмент кода Pygame.

```
from pygame.base import *
from pygame.constants import *
from pygame.version import *
from pygame.rect import Rect
from pygame.compat import PY_MAJOR_VERSION
from pygame.rwobject import encode_string, encode_file_path
import pygame.surflock
import pygame.color
Color = color.Color
import pygame.bufferproxy
BufferProxy = bufferproxy.BufferProxy
import pygame.math
Vector2 = pygame.math.Vector2
Vector3 = pygame.math.Vector3
__version__ = ver

# next, the "standard" modules
# we still allow them to be missing for stripped down pygame distributions
if get_sdl_version() < (2, 0, 0):
# cdrom only available for SDL 1.2.X
try:
import pygame.cdrom
except (ImportError, IOError):
cdrom = MissingModule("cdrom", urgent=1)
try:
import pygame.cursors
except (ImportError, IOError):
cursors = MissingModule("cursors", urgent=1)
try:
import pygame.display
except (ImportError, IOError):
display = MissingModule("display", urgent=1)
try:
import pygame.draw
except (ImportError, IOError):
```

```

draw = MissingModule("draw", urgent=1)
try:
import pygame.event
except (ImportError, IOError):
event = MissingModule("event", urgent=1)
try:
import pygame.image
except (ImportError, IOError):
image = MissingModule("image", urgent=1)
try:
import pygame.joystick
except (ImportError, IOError):
joystick = MissingModule("joystick", urgent=1)
try:
import pygame.key
except (ImportError, IOError):
key = MissingModule("key", urgent=1)
try:
import pygame.mouse
except (ImportError, IOError):
mouse = MissingModule("mouse", urgent=1)
try:
import pygame.sprite
except (ImportError, IOError):
sprite = MissingModule("sprite", urgent=1)
try:
import pygame.threads
except (ImportError, IOError):
threads = MissingModule("threads", urgent=1)
try:
import pygame.pixelcopy
except (ImportError, IOError):

pixelcopy = MissingModule("pixelcopy", urgent=1)
def warn_unwanted_files():
    """warn about unneeded old files"""
    py_files = ["%s%s" % (x, py_ext)
for py_ext in [".py", ".pyc", ".pyo"]
for x in py_to_remove]

files = py_files + extension_files
unwanted_files = []
or f in files:
unwanted_files.append(os.path.join(install_path, f))

```

```

ask_remove = []
for f in unwanted_files:
    if os.path.exists(f):
        ask_remove.append(f)
if ask_remove:
    message = "Detected old file(s). Please remove the old files:\n"
    for f in ask_remove:
        message += "%s " % f
    message += "\nLeaving them there might break pygame. Cheers!\n\n"
    try:
        import warnings
        level = 4
        warnings.warn(message, RuntimeWarning, level)
    except ImportError:
        print (message)

```

Pandas - структурирует данные, применяется для любого сайта. Эта библиотека создана для анализа и обработки данных. Pandas позволяет формировать запросы к данным и многое другое. На рисунке 2.3 рассмотрим реализацию библиотеки Pandas.

```

from pandas.io.json import _json_normalize as json_normalize

from pandas.util._tester import test
import pandas.testing
import pandas.arrays

# use the closest tagged version if possible
from ._version import get_versions

v = get_versions()
__version__ = v.get("closest-tag", v["version"])
__git_version__ = v.get("full-revisionid")
del get_versions, v

```

Рисунок 2.3 - Реализация библиотеки Pandas

### Продолжение рисунка 2.3

```
+ TODO: remove Panel compat in 1.0
if pandas.compat.PY37:

    def __getattr__(name):
        import warnings

        if name == "Panel":

            warnings.warn(
                "The Panel class is removed from pandas. Accessing it "
                "from the top-level namespace will also be removed in "
                "the next version",
                FutureWarning,
                stacklevel=2,
            )

            class Panel:
                pass

            return Panel

        elif name == "datetime":
            warnings.warn(
                "The pandas.datetime class is deprecated "
                "and will be removed from pandas in a future version. "
                "Import from datetime module instead.",
                FutureWarning,
                stacklevel=2,
            )

            from datetime import datetime as dt

            return dt

        elif name == "np":

            warnings.warn(
                "The pandas.np module is deprecated "
                "and will be removed from pandas in a future version. "
                "Import numpy directly instead",
                FutureWarning,
                stacklevel=2,
            )
            import numpy as np

            return np
```

Рассмотрим код программы в виде блок-схемы. Блок-схема выполнялась в программе Visustin.

Visustin позволяет преобразовать исходный код в блок-схемы и UML - диаграммы (Activity Diagram). Visustin поддерживает 49 языков (Python, Java, C/C++, PHP, C#, JavaScript, HTML, Ruby, Shell, Pascal, АВАР, COBOL, Fortlan и другие. С помощью этой программы мы можем увидеть логику нашей программы. На рисунке 2.4 изображен интерфейс программы Visustin.

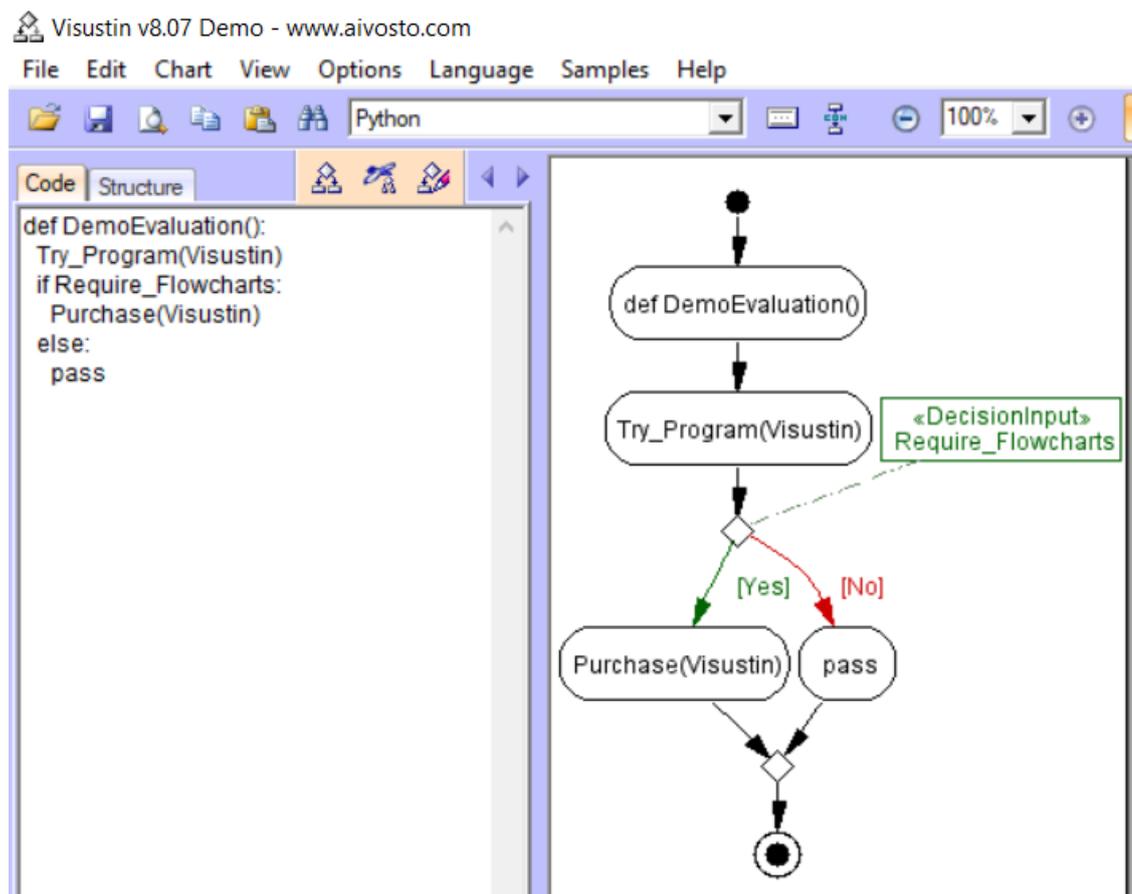


Рисунок 2.4 - Интерфейс программы Visustin

Рассмотрим работу некоторых команд программы в виде блок-схемы.

На рисунке 2.5 можем увидеть логику выполнения команд программы. При запуске программы задаем команду голосовому помощнику. Далее, получаем один из голосовых ответов «Бір минут», «Қазір орындаймын», «Орындап жатырмын». После этого совершается поиск команды. Если, голосовой помощник находит заданную команду, то получаем результат. Иначе, получаем голосовой ответ «Команда табылған жоқ» и «Тағы тапсырмаңыз барма?». После этого можно повторить команду, либо завершаем работу.

На рисунке 2.5 представлена блок - схема выполнения команд.

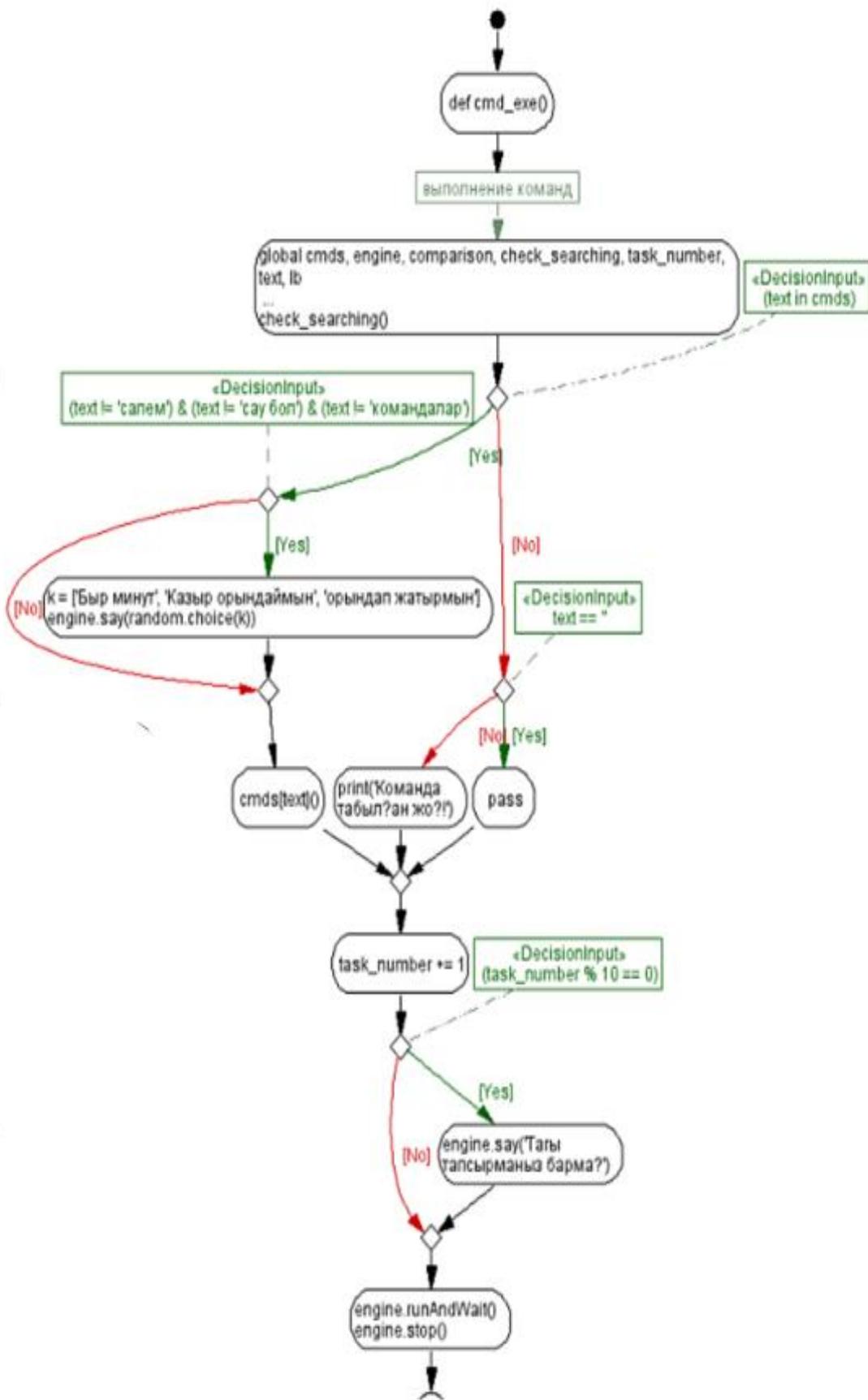


Рисунок 2.5 - Блок - схема выполнения команд

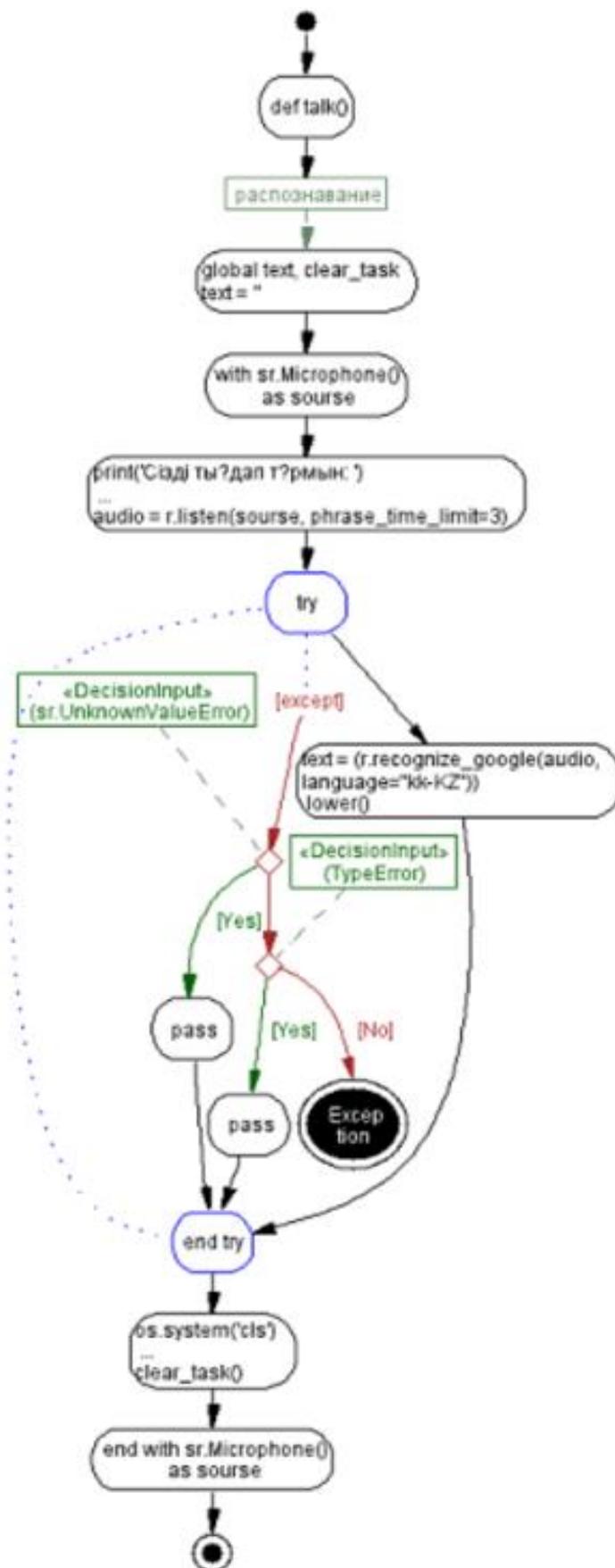


Рисунок 2.6 - Блок - схема распознавания

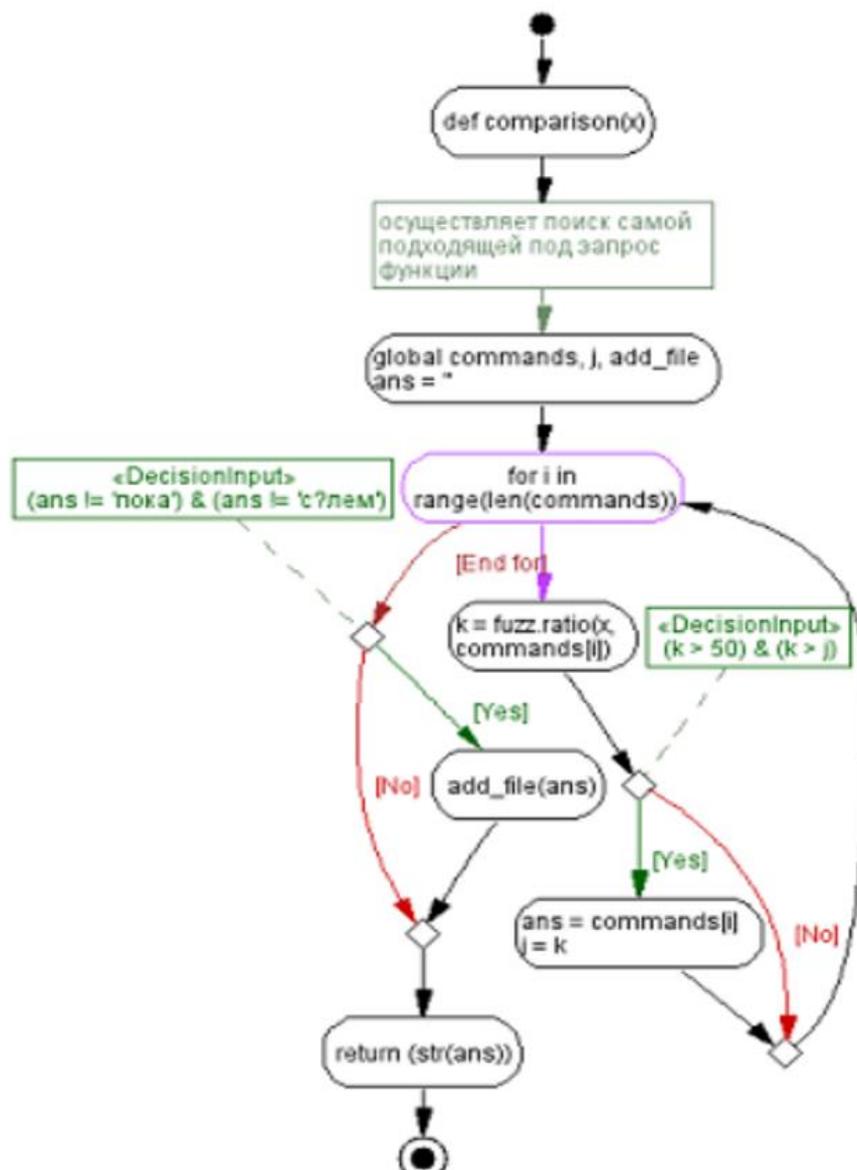


Рисунок 2.7 - Блок - схема функции

### 3 Экспериментальная часть

#### 3.1 Назначение программы

Интерфейс программы состоит из одного основного окна, показанного на рисунке 3.1.



Рисунок 3.1 - Интерфейс программы

Имя голосового помощника «Айсұлу». На интерфейсе голосового помощника «Айсұлу» расположены две кнопки «Тыңдау», «Шығу» и поле вывода. После запуска программы кликаем на кнопку «Тыңдау» и задаем команду. На поле вывода отображается наша команда в виде текста, а также используется при преобразований речи в текст.

Голосовой помощник может выполнять следующие команды:

- осуществлять поиск в интернете по запросу;
- открывать внешние приложения;
- выводить на экран список доступных команд;
- переводить слова и предложения;
- преобразовывать речь в текст;
- диалог с пользователем.

С помощью кнопки «Шығу» осуществляется выход.

Выводим список доступных команд:

- Сізді тыңдап тұрмын:  
командалар
- 1) сәлемдесу
  - 2) сыртқы командаларды ашу
  - 3) компьютерды өшіру
  - 4) ғаламтордан іздеу
  - 5) қоштасу
  - 6) файлды ашу
  - 7) командаларды көрсету
  - 8) сөз және сөйлемдерді аудару



Рисунок 3.2 - Список доступных команд

Проверим работу команд голосового помощника. На рисунках 3.2-3.8 представлены скриншоты выполнения команд.

Открывает внешние приложения.

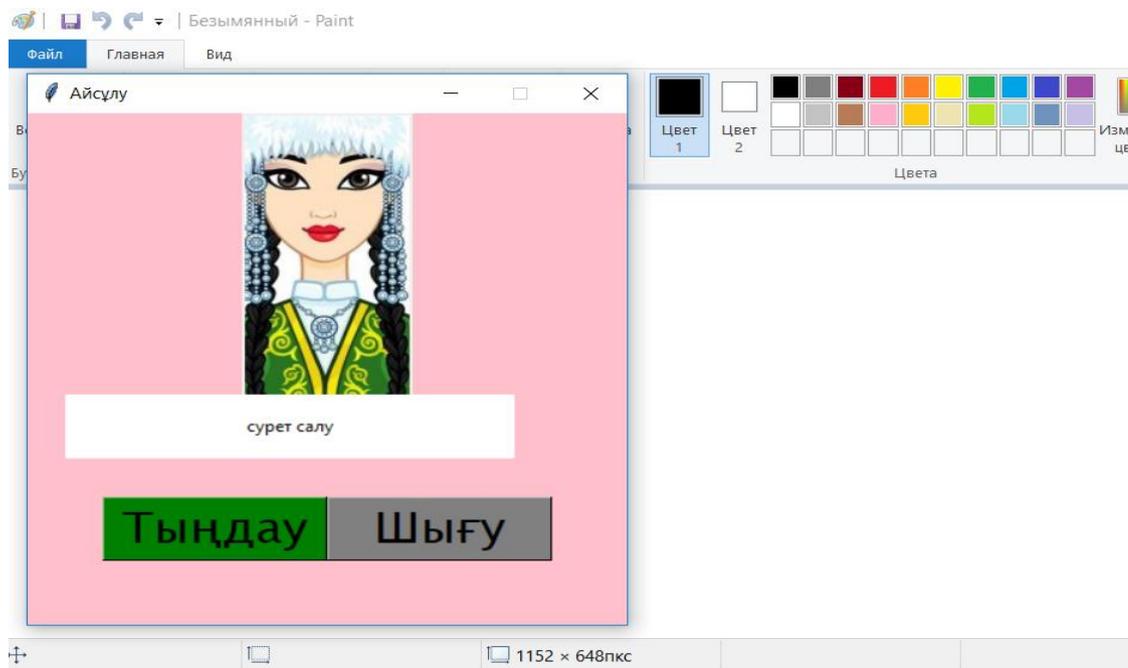


Рисунок 3.3 - Команда «Сурет салу»

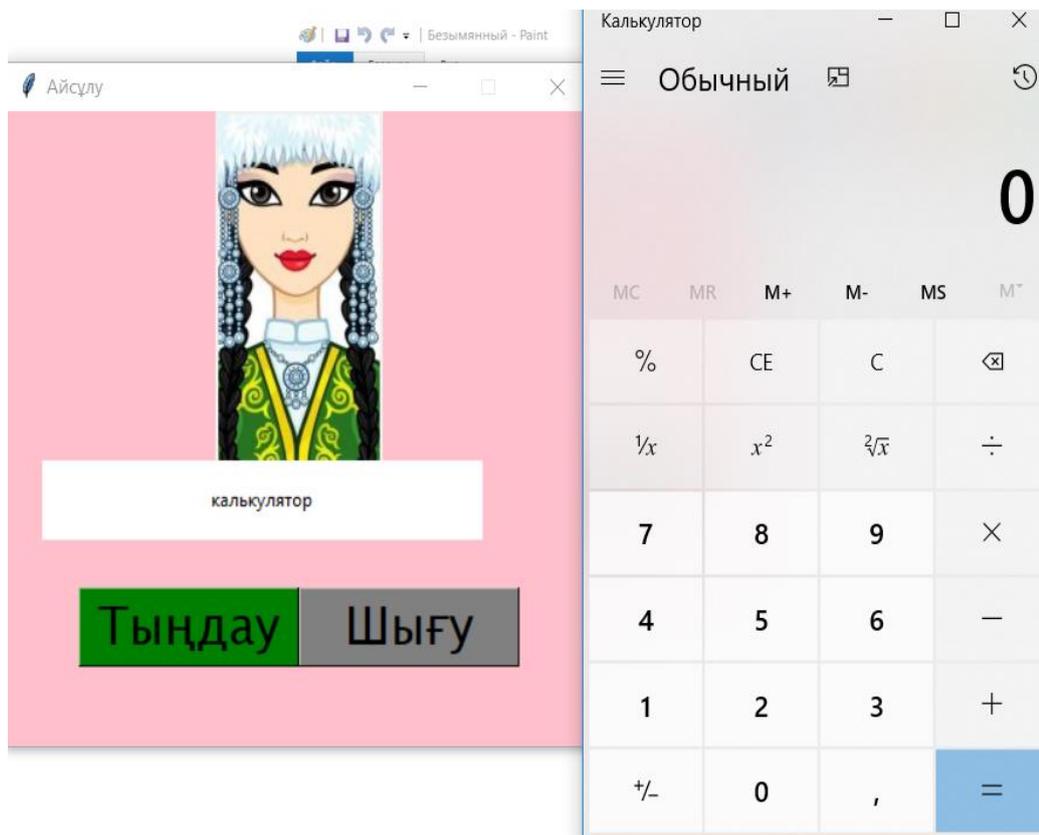


Рисунок 3.4 - Команда «Калькулятор»

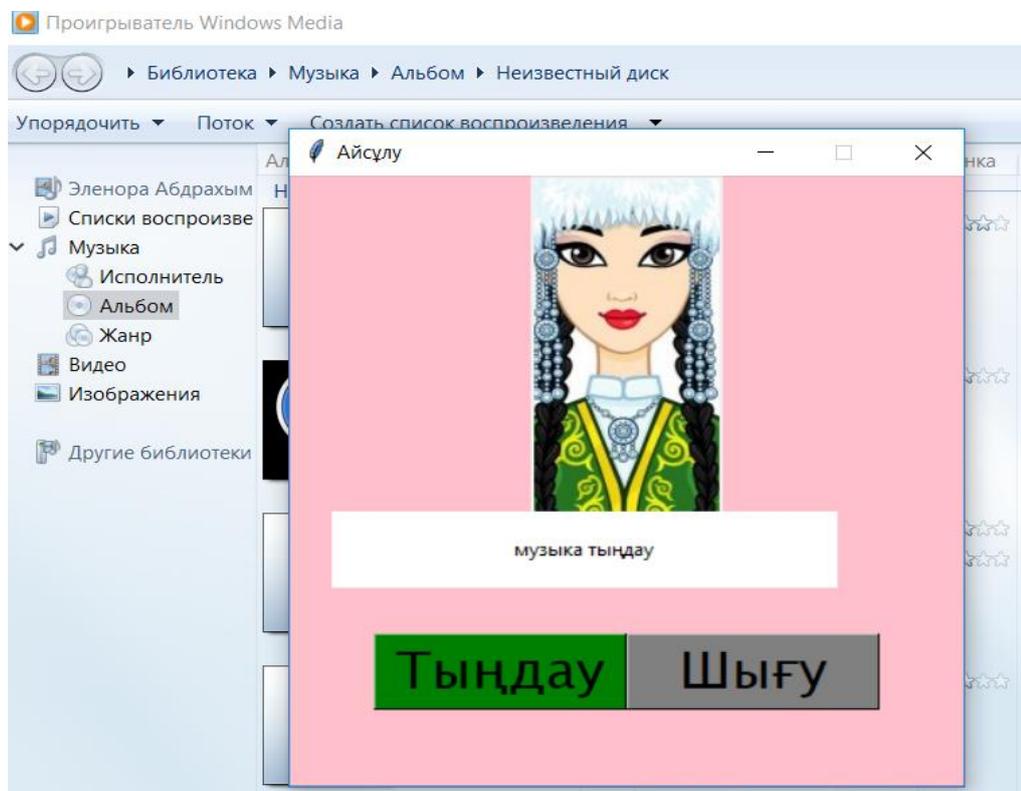


Рисунок 3.5 - Команда «Музыка тыңдау»

Переводит слова и предложения.

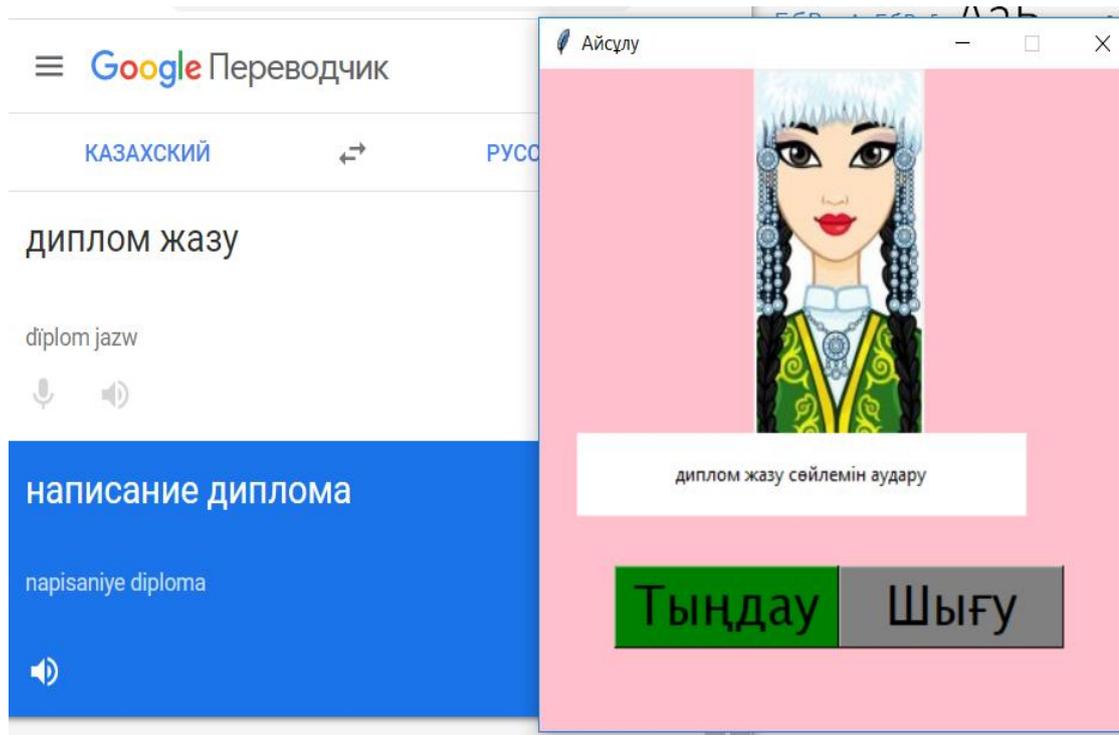


Рисунок 3.6 - Команда «Аудармашы»

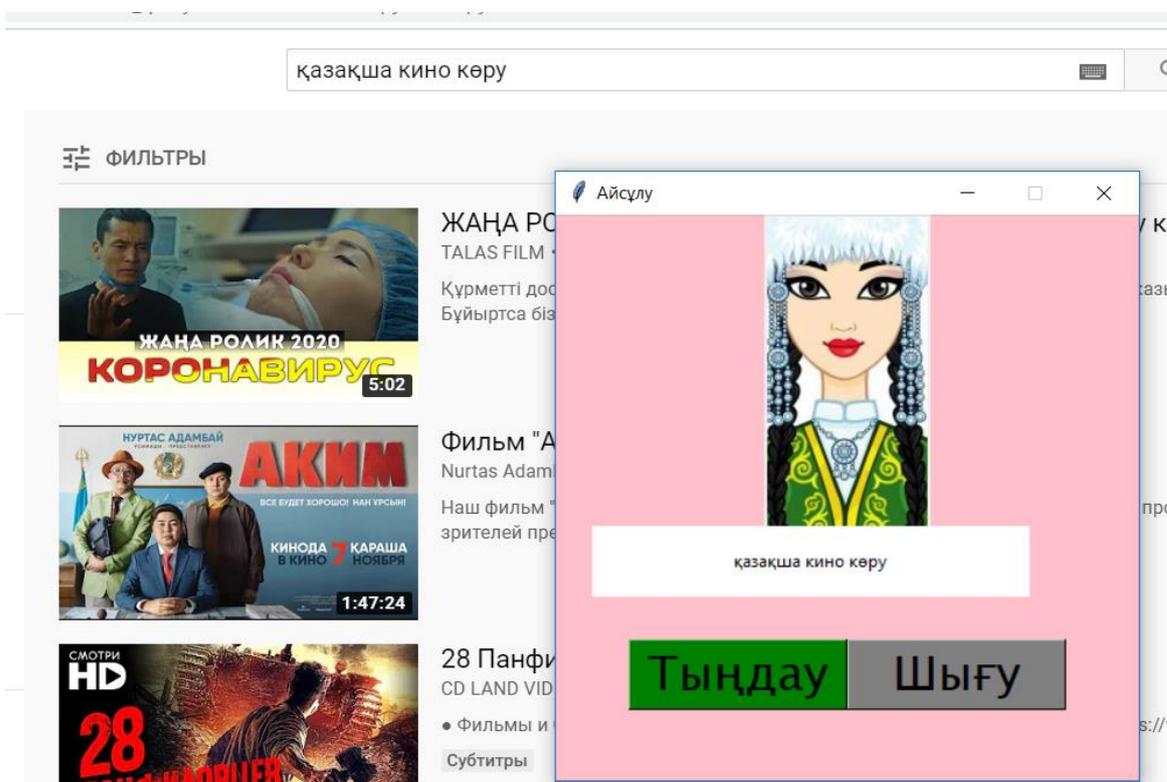


Рисунок 3.7 - Команда «Қазақша кино көру»  
С помощью команды «файлды аш» выводится история запросов.

13)	командалар	11
14)	файлды аш	9
15)	музыканы аш	1
16)	vk қос	2
17)	статистиканы көрсет	1
18)	аудар	2
19)	командаларды көрсету	2
20)	сәлемдесу	15
21)	компьютерды өшіру	1
22)	музыка қос	1
23)	ғаламторды аш	1
24)	калькулятор	4
25)	сурет салу	3
26)	сау бол	4
27)	сағат	1
28)	ютуб	1
29)	кино көру	1
30)	сенің атың кім	1
31)	іздеу	4

Рисунок 3.8 - Команда «Файлды аш»

Файл состоит из трех столбцов.

- 1) нумерация команд;
- 2) название команд;
- 3) количество вывода данной команды.

Также, можно очистить файл командой «файлды тазала».

### 3.2 Установка и настройка программы

Перед началом работы, нужно установить:

- python;
- текстовый редактор PyCharm;
- пакетный менеджер Pip.

С официального сайта <https://python.org/> скачала и установила Python 3.6.

Понадобится 32-разрядная версия так как, достаточно тяжело найти бинарники для 64-разрядной версии, а это значит, что некоторые сторонние модули могут установиться неправильно на 64-разрядной версии Python.

Проверяем установку Python 3.6.0. В командной строке вводим python.

```

Microsoft Windows [Version 10.0.17134.1246]
(c) Корпорация Майкрософт (Microsoft Corporation), 2018. Все права защищены.

C:\Users\user>python
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>

```

Рисунок 3.9 - Проверка установки Python

Пакетный менеджер Pip, устанавливается автоматический, либо через терминал, командой `pip install pip`.

```

Usage:
  pip <command> [options]

Commands:
  install             Install packages.
  download            Download packages.
  uninstall           Uninstall packages.
  freeze              Output installed packages in requirements format.
  list                List installed packages.
  show                Show information about installed packages.
  check               Verify installed packages have compatible dependencies.
  config              Manage local and global configuration.
  search              Search PyPI for packages.
  wheel               Build wheels from your requirements.
  hash                Compute hashes of package archives.
  completion          A helper command used for command completion.
  debug               Show information useful for debugging.
  help                Show help for commands.

General Options:
  -h, --help          Show help.
  --isolated           Run pip in an isolated mode, ignoring environment variables and user configuration.
  -v, --verbose        Give more output. Option is additive, and can be used up to 3 times.
  -V, --version        Show version and exit.
  -q, --quiet          Give less output. Option is additive, and can be used up to 3 times (corresponding to
WARNING, ERROR, and CRITICAL logging levels).
  --log <path>        Path to a verbose appending log.
  --proxy <proxy>      Specify a proxy in the form [user:passwd@]proxy.server:port.
  --retries <retries> Maximum number of retries each connection should attempt (default 5 times).
  --timeout <sec>     Set the socket timeout (default 15 seconds).
  --exists-action <action> Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup,
(a)abort.
  --trusted-host <hostname> Mark this host or host:port pair as trusted, even though it does not have valid or any
HTTPS.
  --cert <path>        Path to alternate CA bundle.
  --client-cert <path> Path to SSL client certificate, a single file containing the private key and the
certificate in PEM format.
  --cache-dir <dir>   Store the cache data in <dir>.
  --no-cache-dir      Disable the cache.
  --disable-pip-version-check Don't periodically check PyPI to determine whether a new version of pip is available for

```

Рисунок 3.10 - Пакетный менеджер pip

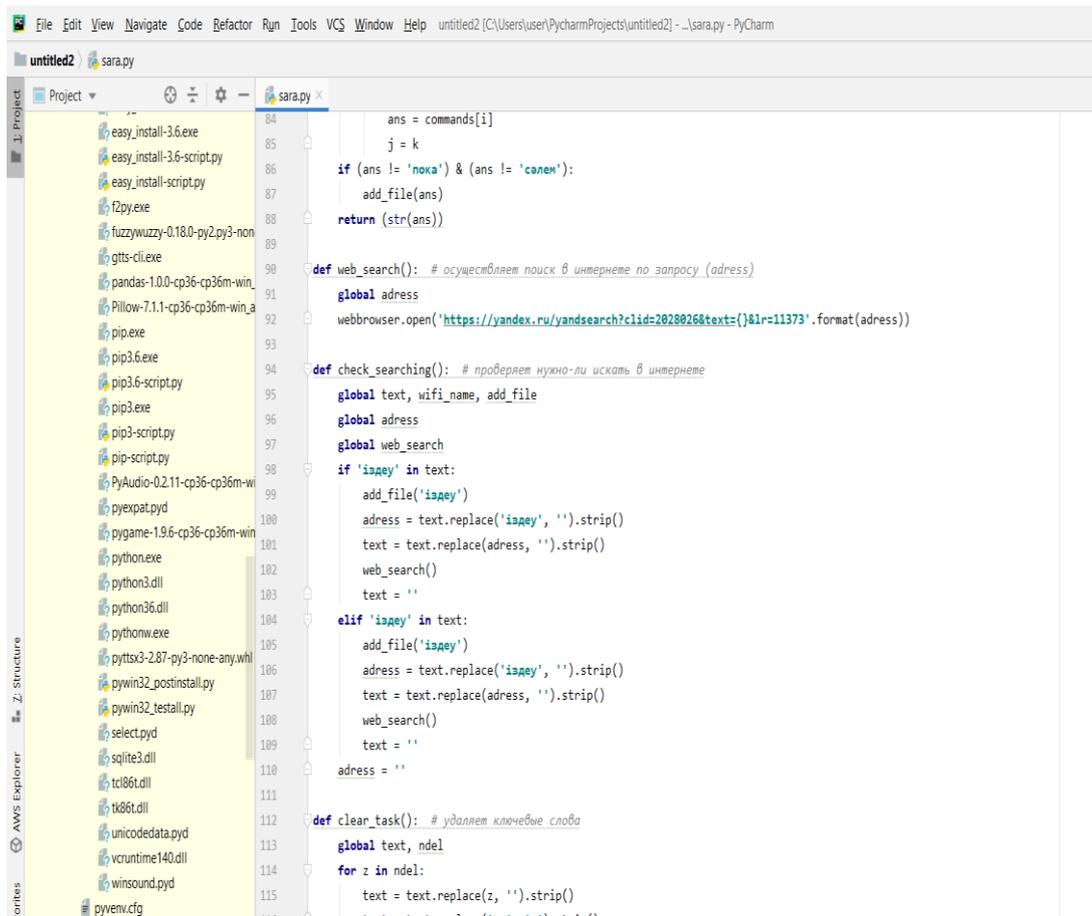


Рисунок 3.11 - Текстовый редактор PyCharm

На Рисунке- 3.12 представлен список установленных библиотек. Все библиотеки устанавливаем через терминал.

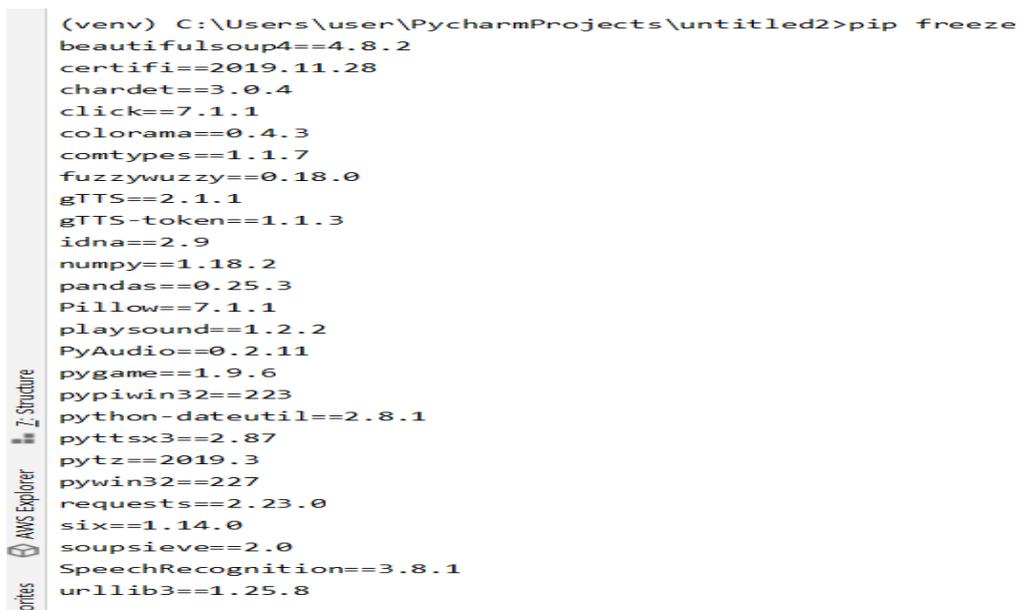


Рисунок 3.12 - Список библиотек

На рисунках 3.13- приведены скриншоты фрагментов кода программы. Полный листинг программы представлен в Приложении А.

```
4 def pri_com(): # выводит на экран историю запросов
5     z = {}
6     mas = []
7     mas2 = []
8     mas3 = []
9     mas4 = []
0     file = open('commands.txt', 'r', encoding='UTF-8')
1     k = file.readlines()
2     for i in range(len(k)):
3         line = str(k[i].replace('\n', ' ').strip())
4         mas.append(line)
5     file.close()
6     for i in range(len(mas)):
7         x = mas[i]
8         if x in z:
9             z[x] += 1
0         if not (x in z):
1             b = {x: 1}
2             z.update(b)
3         if not (x in mas2):
4             mas2.append(x)
5     for i in mas2:
6         mas3.append(z[i])
7     for i in range(1, len(mas3) + 1):
8         mas4.append(str(i) + ' ')
9     list = pd.DataFrame({
0         'command': mas2,
1         'count': mas3
2     }, index=mas4)
3     list.index.name = '№'
4     print(list)
```

Рисунок 3.13 - История запросов

```

def clear_analis(): # очистка файла с историей запросов
    global engine
    file = open('commands.txt', 'w', encoding='UTF-8')
    file.close()
    engine.say('Файл тазаланды')

def add_file(x):
    file = open('commands.txt', 'a', encoding='UTF-8')
    if x != '':
        file.write(x + '\n')
    file.close()

def comparison(x): # осуществляет поиск самой подходящей под запрос функции
    global commands, j, add_file
    ans = ''
    for i in range(len(commands)):
        k = fuzz.ratio(x, commands[i])
        if (k > 50) & (k > j):
            ans = commands[i]
            j = k
    if (ans != 'пока') & (ans != 'сәлем'):
        add_file(ans)
    return (str(ans))

```

Рисунок 3.14 - Очистка файла с историей запросов

```

def hello(): # функция приветствия
    global engine
    z = ["Сызге калай комектесе аламын?", "Кандай комек керек?", 'Салем Сызды тындап турмын']
    x = random.choice(z)
    engine.say(x)

def name(): # диалог
    global engine
    z = ["Менын атым Айсулу"]
    x = random.choice(z)
    engine.say(x)

def quit(): # функция выхода из программы
    global engine
    x = ['тагы кездесемыз деп ойлаймын', 'комектескениме куаныштымын', 'сау болыңыз']
    engine.say(random.choice(x))
    engine.runAndWait()
    engine.stop()
    os.system('cls')
    exit(0)

```

Рисунок 3.15 - Функции

```

# распознавание
def talk():
    global text, clear_task
    text = ''
    with sr.Microphone() as source:
        print('Сізді тыңдап тұрмын: ')
        r.adjust_for_ambient_noise(source)
        audio = r.listen(source, phrase_time_limit=3)
        try:
            text = (r.recognize_google(audio, language="kk-KZ")).lower()
        except (sr.UnknownValueError):
            pass
        except (TypeError):
            pass
    os.system('cls')
    lb['text'] = text
    clear_task()

```

Рисунок 3.16 - Распознавание голоса

```

def cmd_exe():
    global cmds, engine, comparison, check_searching, task_number, text, lb
    check_translate()
    text = comparison(text)
    print(text)
    check_searching()
    if (text in cmds):
        if (text != 'салем') & (text != 'сау бол') & (text != 'командалар'):
            k = ['Быр минут', 'Казыр орындаймын', 'орындап жатырмын']
            engine.say(random.choice(k))
            cmds[text]()
    elif text == '':
        pass
    else:
        print('Команда табылған жоқ!')
    task_number += 1
    if (task_number % 10 == 0):
        engine.say('Тағы тапсырманыз барма?')
    engine.runAndWait()
    engine.stop()

```

Рисунок 3.17 - Распознавание команд

```

283 # раздел создания интерфейса
284 root = Tk()
285
286 root.geometry('400x400')
287 canvas = Canvas(root, width=110, height=265)
288 canvas.pack()
289 pilImage = Image.open("kaz.jpg")
290 image = ImageTk.PhotoImage(pilImage)
291 imagesprite = canvas.create_image(50, 40, image=image)
292 root.configure(bg='pink')
293 root.title('Айсұлу')
294 root.resizable(False, False)
295
296 lb = Label(root, text=text)
297 lb.configure(bg='white')
298 lb.place(x=25, y=220, height=50, width=300)
299
300 but1 = Button(root, text='Тыңдау', command=main)
301 but1.configure(bd=1, font=('Castellar', 25), bg='green')
302 but1.place(x=200, anchor="se", y=350, height=50, width=150)
303
304 but2 = Button(root, text='Шығу', command=quit)
305 but2.configure(bd=1, font=('Castellar', 25), bg='gray')
306 but2.place(x=200, anchor="sw", y=350, height=50, width=150)
307
308 root.mainloop()
309
310 while True:
311     main()
312

```

Рисунок 3.18 - Создание интерфейса

```

cmds = {
    'сәлем': hello, 'калькулятор': calc,
    'компьютерді өшір': shut,
    'сау бол': quit, 'командалар': show_cmds,
    'vk қос': ovk, 'ғаламторды аш': brows,
    'ютуб': youtube, 'музыканы өшір': musik, 'vkны аш': ovk,
    'музыка қос': musik, 'файлды тазала': clear_analis,
    'файлды көрсет': pri_com, 'файлды аш': pri_com, 'музыка': musik,
    'аудар': check_translate, 'кино көру': cinema,
    'сәлемдесу': hello, 'компьютерды өшіру': shut, 'ғаламтордан іздеу': brows,
    'қоштасу': hello, 'командаларды көрсету': pri_com,
    'аудармашы': check_translate, 'сурет салу': paint, 'атың кім': name, 'іздеу': check_searching
}

```

Рисунок 3.19 - Список команд

```

def show_cmds(): # выводит на экран список доступных команд
    my_com = ['1) сәлемдесу', '2) сыртқы командаларды ашу', '3) компьютерды өшіру',
              |'4) ғаламтордан іздеу', '5) қоштасу', '6) файлды ашу', '7) командаларды көрсету',
              |'8) сөз және сөйлемдерді аудару']
    for i in my_com:
        print(i)
        time.sleep(2)

def brows(): # открывает браузер
    webbrowser.open('https://google.ru')

def ovk(): # открывает вк
    webbrowser.open('https://vk.com/feed')

def youtube(): # открывает ютюб
    webbrowser.open('https://www.youtube.com')

def calc(): # открывает калькулятор
    webbrowser.open('calc')

def paint(): # открывает ютюб
    webbrowser.open('C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessories\paint')

def shut(): # выключает компьютер
    global quit
    os.system('shutdown /s /f /t 10')
    quit()

def musik(): # включает музыку
    webbrowser.open('C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessories')

```

Рисунок 3.20 - Функции выполнения команд

## 4 Технико-экономическое обоснование проекта

Цель экономической части дипломной работы - при помощи всесторонней оценки, анализа и расчетов доказать экономическую целесообразность и техническую возможность разработки и реализации конкретной системы автоматизации.

### 4.1 Определение объема и трудоемкости разработки программного обеспечения

Таблица 4.1 - Основные этапы разработки проекта

№ п/п	Наименование работ	Длительность работы в марте 2020 года	
		в днях	в часах
1	2	4	5
1	Изучение и описание задачи	2	16
2	Основной этап разработки	15	120
3	Отладка программы	6	48
5	Ввод в действие	5	40
	Итого	28	224

Таким образом, на разработку голосового помощника было затрачено **28** дней в марте 2020 года, или **224** человеко-часа (при восьмичасовом рабочем дне).

### 4.2 Расчет затрат на разработку информационной технологий

Расчет полных затрат на разработку проектного решения осуществляется по формуле 4.1 [7].

$$C_{ni} = Z_{фот} + Z_{сзи} + M_i + P_{ci} + P_{mi} + P_{нки} + P_{zi} + P_{ni}, \quad (4.1)$$

где  $Z_{фот}$  – общий фонд оплаты труда разработчиков, тенге;  
 $Z_{сзи}$  – отчисления по социальному налогу, тенге;  
 $M_i$  – затраты на материалы, тенге;  
 $P_{ci}$  – затраты на специальные программные средства, необходимые для разработки проектного решения, тенге;  
 $P_{mi}$  – затраты, связанные с эксплуатацией техники, тенге;  
 $P_{нки}$  – затраты на научные командировки, тенге;  
 $P_{zi}$  – прочие затраты, тенге;  
 $P_{ni}$  – накладные расходы, тенге.

Размер фонда оплаты труда разработчика ( $Z_{фот}$ ) рассчитывается по формуле:

$$Z_{фот} = Z_{oi} + Z_{di}, \quad (4.2)$$

где  $Z_{oi}$  – основная заработная плата, тенге;  
 $Z_{di}$  – дополнительная заработная плата, тенге.

Время на разработку проекта составляет 28 дней. Месячная заработная плата руководителя – 200 000 тенге программиста – 150 000 тенге.

Заработная плата за один час рассчитаем по формуле 4.3:

$$(4.3) \quad D = \frac{ЗП_m}{D_p \times Ч_p},$$

где  $ЗП_m$  – ежемесячный размер заработной платы;  
 $D_p$  – количество рабочих дней в месяц (21 рабочих дня);  
 $Ч_p$  – количество часов рабочего дня (при 8 часовом рабочем дне).

По формуле (4.3) заработная плата руководителя за один час составляет:

$$D = \frac{200000}{21 \times 8} = 1190 \frac{\text{тенге}}{\text{час}}$$

По формуле (4.3) заработная плата программиста за один час составляет:

$$D = \frac{150000}{21 \times 8} = 892 \frac{\text{тенге}}{\text{час}}$$

Таблица 4.2 - Затраты на оплату труда

Категория работника	Квалификация	Трудоемкость разработки ПП, чел.×час	Часовая ставка, тг/ч	Сумма, тг
руководитель	руководитель	64	1190	76 160
специалист	программист	160	892	142 720
<b>ИТОГО на затраты оплаты труда</b>				<b>218 880</b>

Дополнительная заработная плата составляет 10 % от основной заработной платы и рассчитывается по формуле 4.4 [7].

$$З_{доп} = Z_{осн} \times 10\%, \quad (4.4)$$

$$З_{доп} = 218\,880 \times 0,1 = 21\,888 \text{ тенге.}$$

Таким образом, суммарный фонд оплаты труда составит

$$ФОТ = 218\,880 + 21\,888 = 240\,768 \text{ тенге.}$$

Затраты на оплаты труда- 240 768Т

Таблица 4.3 - Уплаченные налоги

Уплаченные юридическим лицом налоги	Ставка налога	161 145	
СО (Социальные отчисления)	3,5	(ЗП - ОПВ)*3,5%	7584,19
ВОСМСЮ (Отчисления на ВОСМСЮ)	2,0	ЗП*2%	4815,36
СН (Социальный налог)	9,5	(ЗП-ОПВ-ВОСМС)* *9,5%-СО	12 772,74
Всего уплаченные налоги			<b>25 172,29</b>

Расчет затрат на материальные ресурсы описан в таблице 4.4, которая представлена ниже.

Таблица 4.4 - Затраты на материальные ресурсы

Наименование материального ресурса	Единица измерения	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
Блокнот(А4)	Штук	1	3200	3200
Карандаш	Штук	2	60	120
Ручки	Штук	2	120	240
ИТОГО затраты на материальные ресурсы				<b>3560</b>

Затраты на материальные ресурсы рассчитаем по формуле 4.5:

$$Z_{\text{общее}} = Z_{\text{блокнот}} + Z_{\text{карандаши}} + Z_{\text{ручки}} , \quad (4.5)$$

$$Z_{\text{общее}} = 3200 + 120 + 240 = 3560 \text{ тенге}$$

Определяем общую сумму затрат на различные материальные ресурсы ( $Z_M$ ), по формуле 4.5, которая представлена ниже:

$$Z_M = \sum_{i=1}^n P_i * C_i, \quad (4.6)$$

где  $P_i$  - расход  $i$ -го вида материального ресурса, натуральные единицы;

$C_i$  - цена за единицу  $i$ -го вида материального ресурса, тг;

$i$  - вид материального ресурса;

n - количество видов материальных ресурсов.

Таблица 4.5 - Расчет затрат на оборудование и ПО, необходимое для проекта

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Ноутбук	Ноутбук HP PAVILION 15-ab060ur	Штук	1	135000	135000
Компьютерная мышь беспроводная	Logitech M185	Штук	1	7 080	7 080
Подписка Play Market			1	10 668	10668
<b>Итого</b>					<b>152 748</b>

$$Z_{\text{общее}} = 135000 + 7080 + 10668 = 152\,748$$

Так как для разработки программного продукта используются техника, которая потребляет электроэнергию, то необходимо рассчитать затраты на электроэнергию, которые представлены в таблице 4.6.

Таблица 4.6 - Затраты на электроэнергию

Наименование оборудования	Паспортная мощность, кВт	Коэффициент использования мощности	Время работы оборудования для разработки ПП, ч	Цена электроэнергии, тг/кВт*ч	Сумма, тг
Ноутбук HP Pavilion 15-ab060ur	0,6	0,7	224	19,17	1932
<b>ИТОГО</b> затраты на электроэнергию					<b>1972,3</b>

Общая сумма затрат на электроэнергию ( $Z_9$ ) рассчитывается по формуле 4.7 [8]:

$$Z_9 = \sum_{i=1}^n M_i * K_i * T_i * Ц, \quad (4.7)$$

где  $M_i$  - паспортная мощность  $i$ -го электрооборудования, кВт;

$K_i$  - коэффициент использования мощности  $i$ -го электрооборудования (принимается  $K_i=0.7, 0.9$ );

$T_i$  - время работы  $i$ -го оборудования за весь период разработки ПП  
ч;

$\Pi$  - цена электроэнергии, тг/кВт×ч;

$i$  - вид электрооборудования;

$n$  - количество электрооборудования.

$$Z_{\text{общие}} = (0,6 \times 0,7 \times 224 \times 19,17) + (0,5 \times 0,7 \times 6 \times 19,17) = 1843,77 \text{ тенге}$$

Расходы по статье «Прочие затраты» на конкретное ПО включают затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы [7]. Рассчитывается по формуле 4.8.

$$\Pi z_i = Z_{oi} \times \Pi_{пз} / 100, \quad (4.8)$$

где  $\Pi_{пз}$  – норматив прочих затрат в целом по организации в (%), в дипломной работе нужно брать 20%.

$$\Pi z_i = 146496 \times 20 / 100 = 29\ 299$$

Таблица 4.7 - Смета затрат на разработку программы

Статьи затрат	Сумма, тг
$Z_{\text{фот}}$ – общий фонд оплаты труда разработчиков, тенге;	161 145
$Z_{\text{сзи}}$ - отчисления по социальному налогу, тенге;	25 172,29
$M_i$ - затраты на материалы, тенге;	3560
$P_{\text{си}}$ - затраты на специальные программные средства, необходимые для разработки проектного решения, тенге;	152748
$Z_{\text{э}}$ -затраты на электроэнергию	1972,3
$\Pi z_i$ -прочие затраты, тенге;	29 299
<b>ИТОГО по смете</b>	<b>373 896,59</b>

### 4.3 Сравнительный анализ эксплуатационных затрат до и после внедрения

Статьи затрат после внедрения программного обеспечения:

- основная и дополнительная заработная плата программиста с отчисляемыми налогами;
- износ ПЭВМ;
- накладные расходы.

Статьи затрат до внедрения:

- основная и дополнительная заработная плата программиста;
- расходуемые материалы;
- накладные расходы.

*Расчет затрат после внедрения ПО*

Таблица заработной платы и отчисляемых налогов представлена в таблице 4.8

Таблица 4.8 - Затраты на заработную плату

Сотрудник	Кол-во, чел	Заработная плата в месяц, тг	Заработная плата в год, тг	Социальное отчисление	Социальный налог	ВОСМС
Программист	1	150 000	1 800 000	56 700	95 490	36 000
Итого	1	150 000	1 800 000	56 700	95 490	36 000
К оплате	<b>1 988 190</b>					

Износ ПЭВМ рассчитывается исходя из 25 % амортизационных отчислений за год [7]. Занесем данные в таблицу 4.9.

Таблица 4.9 - Амортизационные отчисления

Оборудование	Количество	Стоимость, тенге	Отчисления
Ноутбук HP PAVILION 15-ab060ur	1	135000	33 750
Компьютерная мышь беспроводная	1	7 080	1 770
Итого	2	142 080	35 520

Общие накладные расходы рассчитываются по формуле

$$Z_{н.р.} = 1\,988\,190 * 0,2 = 397\,638 \text{ тенге.} \quad (4.9)$$

Итого, общие эксплуатационные расходы после внедрения программного продукта будут равны,

$$Z = 1\,988\,190 + 35\,520 + 397\,638 = 2\,421\,348 \text{ тенге.}$$

*Расчет затрат до внедрения ПО*

Таблица заработной платы и отчисляемых налогов представлена в таблице 4.10.

Таблица 4.10 - Затраты на заработную плату

Сотрудник	Кол-во, чел	Заработная плата в месяц одного, тг	Заработная плата в год, тг	Социально отчисления	Социальный налог	ВОСМС
Руководитель	1	200 000	2 400 000	75 600	127 320	48 000
Программист	1	150 000	1 800 000	56 700	95 490	36 000
Итого	2	350 000	4 200 000	132 300	222 810	84 000
К оплате						<b>4 639 110</b>

Износ ПЭВМ рассчитывается исходя из 25 % амортизационных отчислений за год [7]. Занесем данные в таблицу 4.11

Таблица 4.11 - Амортизационные отчисления

Оборудование	Количество	Стоимость, тенге	Отчисления
Ноутбук HP PAVILION 15-ab060ur	1	135000	33 750
Компьютерная мышь беспроводная	1	7 080	1 770
Итого	2	142 080	35 520

Затраты на расходуемые материалы занесем в таблицу 4.12

Таблица 4.12 - Затраты на расходуемые материалы

Наименование материального ресурса	Единица измерения	Количество израсходованного материала	Цена за единицу, тг
Блокнот(А4)	Штук	1	3200
Карандаш	Штук	2	60
Ручки	Штук	2	120
<b>ИТОГО</b> затраты на материальные ресурсы -			<b>3560</b>

Общие накладные расходы рассчитываются по формуле (4.10):

$$Z_{н.р.} = 4\,639\,110 * 0,2 = 927\,822 \text{ тенге.}$$

(4.10)

Итого, общие эксплуатационные расходы до внедрения программного продукта будут равны,

$$З = 4\,639\,110 + 35\,520 + 3560 + 927\,822 = 5\,606\,012 \text{ тенге.}$$

Для наглядности сведем все данные в аналитическую таблицу 4.13.

Таблица 4.13 - Годовые эксплуатационные затраты

Статьи	До внедрения ПП	После внедрения ПП
Годовая заработная плата	4 639 110	1 988 190
Расходуемые материалы	3560	-
Износ ПЭВМ	35 520	35 520
Накладные расходы	927 822	397 638
Всего	5 606 012	2 421 348

Годовая экономия денежных средств будет составлять:

$$5\,606\,012 - 2\,421\,348 = 3\,184\,664 \text{ тенге.}$$

Необходимо определить экономический эффект от внедрения программного продукта.

Технико-экономические показатели позволяют определить целесообразность проведения разработки и ее внедрения, а также оценить реальную выгоду, как для разработчика системы, так и для ее пользователя.

Необходимо определить срок окупаемости программного продукта в месяцах по формуле

$$P_{\text{окуп}} = 12 * \frac{Z_{\text{разработка}}}{Э_{\text{годовая}}},$$

(4.11)

где  $P_{\text{окуп}}$  – период окупаемости в месяцах;

$Z_{\text{разр.}}$  – затраты на разработку программы;

$Э_{\text{годовая}}$  – годовая экономия.

$$P_{\text{окуп.}} = 12 \cdot \frac{373\,896,59}{3\,184\,664} = 12 \cdot 0,117 = 1,4 \text{ мес.}$$

Зная годовую экономию и годовые эксплуатационные затраты с внедрением программного продукта можно определить коэффициент эффективности ( $K_{эф.}$ ) по формуле (4.12):

Расчетный коэффициент экономической эффективности капитальных вложений составляет:

$$E_p = \frac{\mathcal{E}_{уг}}{K}, \quad (4.12)$$

$$E_p = \frac{3\,184\,664}{373\,896,59} * 100\% = 854\% \text{ или } 8,54 \text{ тенге.}$$

Коэффициент эффективности показывает сколько экономии в тенге приходится на каждый тенге затрат. Следовательно, на каждый тенге затрат приходится 8,54 тенге экономии.

Сравнивая данные показатели видно, что внедрение программного продукта отлично отразится на прибыльности предприятия, поскольку внедрение способствует уменьшению кадров, что в свою очередь уменьшит выплаты заработной платы и налогов.

## **5 Безопасность жизнедеятельности**

Главная цель нашего общества - превращение труда в первоочередные жизненные потребности человека. Для эффективной работы необходимо создать необходимые условия для трудящихся, сохранить их здоровье, обеспечить безопасность условий труда, избежать профессиональных заболеваний и повреждений на производстве. Наука по охране труда занимается этими вопросами, то есть проблемами создания безвредных, безопасных условий труда.

По данным Международной организации труда в мире ежегодно происходят в среднем 50 миллионов несчастных случаев или 160 тысяч несчастных случаев в день.

Таким образом, охрана труда представляет собой систему обеспечения безопасности жизни и здоровья работников в процессе трудовой деятельности, включающую правовые, социально-экономические, организационно-технические, санитарно-эпидемиологические, лечебно-профилактические, реабилитационные и иные мероприятия и средства. Наука по охране труда выявляет причины несчастных случаев на производстве, профессиональных заболеваний, взрывов и пожаров и разрабатывает необходимые меры по их предотвращению.

### **5.1 Анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал**

Недавно зарубежные ученые провели исследование влияния воздуха в офисе на здоровье человека, в результате чего показали негативное влияние на жизненный возраст и ухудшение работы головного мозга в помещении, где сконцентрированы компьютеры. Как известно, 80% жителей города работают в основном в закрытых учреждениях. Сегодня офис оснащен современной техникой, изготовленным из комбинированных строительных материалов. А ядовитые вещества, выделяемые из этой техники и искусственных строительных материалов, наносят огромный ущерб организму персонала.

Рассмотрим опасные и вредные факторы в офисе, воздействующие на персонал.

Опасные и вредные производственные факторы подразделяются по природе действия на следующие группы [9]:

- физические;
- химические;
- биологические;
- психофизиологические.

Физические опасные и вредные производственные факторы подразделяются на:

- движущиеся - машины и механизмы, подвижные части производственного оборудования, передвигающиеся изделия, заготовки, материалы, разрушающиеся конструкции, обрушивающиеся горные породы;
- повышенная запыленность и загазованность воздуха рабочей зоны;
- повышенная или пониженная температура поверхностей оборудования, материалов;
- повышенная или пониженная температура воздуха рабочей зоны;
- повышенный уровень шума на рабочем месте;
- повышенный уровень вибрации;
- повышенный уровень инфразвуковых колебаний;
- повышенный уровень ультразвука;
- повышенное или пониженное барометрическое давление в рабочей зоне и его резкое изменение;
- повышенная или пониженная влажность воздуха;
- повышенная или пониженная подвижность воздуха;
- повышенная или пониженная ионизация воздуха;
- повышенный уровень ионизирующих излучений в рабочей зоне;
- повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека;
- повышенный уровень статического электричества;
- повышенный уровень электромагнитных излучений;
- повышенная напряженность электрического поля;
- повышенная напряженность магнитного поля;
- отсутствие или недостаток естественного света;
- недостаточная освещенность рабочей зоны;
- повышенная яркость света;
- пониженная контрастность;
- прямая и отраженная блескость;
- повышенная пульсация светового потока;
- повышенный уровень ультрафиолетовой радиации;
- повышенный уровень инфракрасной радиации;
- острые кромки, заусенцы и шероховатость на поверхностях заготовок, инструментов и оборудования;
- расположение рабочего места на значительной высоте относительно поверхности земли (пола);
- невесомость.

Химические опасные и вредные производственные факторы подразделяются [9]:

по характеру воздействия на организм человека на:

- токсические;
- раздражающие;
- сенсibiliзирующие;
- канцерогенные;
- мутагенные;

- влияющие на репродуктивную функцию;
- по пути проникания в организм человека через:
- органы дыхания;
- желудочно-кишечный тракт;
- кожные покровы и слизистые оболочки.

Биологические опасные и вредные производственные факторы включают следующие биологические объекты: патогенные микроорганизмы (бактерии, вирусы, риккетсии, спирохеты, грибы, простейшие) и продукты их жизнедеятельности.

Психофизиологические опасные и вредные производственные факторы по характеру действия подразделяются на следующие [9]:

- а) физические перегрузки;
- б) нервно-психические перегрузки.

Физические перегрузки подразделяются на:

- статические;
- динамические.

Нервно-психические перегрузки подразделяются на:

- умственное перенапряжение;
- перенапряжение анализаторов;
- монотонность труда;
- эмоциональные перегрузки.

### **5.1.1 Электромагнитные поля**

К числу факторов, ухудшающих состояние здоровья работников, использующих в качестве рабочего инструмента средства ПЭВМ, относятся электромагнитное и электростатическое поля, а также рентгеновское излучение.

Электромагнитное поле электронно-лучевого монитора вредно для здоровья, так как распространяется в стороны и в особенности – назад от монитора на расстояние до 1-1,5 м. В связи с чем, желательно размещать монитор таким образом, чтобы излучение не достигало активных участков помещения. Особенно это касается беременных женщин, поскольку некоторые ученые считают, что электромагнитное поле может вредно влиять на эмбрион, начиная с момента зачатия и на протяжении первых месяцев беременности. Жидкокристаллический монитор тоже создает электромагнитное поле, однако оно не столь интенсивно и вредно, в отличие от ЭЛТ мониторов. Поэтому, в плане безопасности жидкокристаллический монитор предпочтительнее. Однако, если все-таки, по каким бы то ни было причинам, пользователю предстоит производить работу на электроннолучевом мониторе, то, первую очередь, требуется приобрести монитор, соответствующий стандартам ТСО-95, ТСО-99 (ТСО (Шведская конфедерация профессиональных рабочих) – является шведским национальным законодательным органом в области ввода стандартов по электрическим и магнитным помехам, излучаемым компьютерной техникой)

либо более поздним. Такие мониторы уже имеют встроенную защиту от электромагнитных излучений (встроенный защитный экран, специальная фольга внутри корпуса), поэтому они почти безопасны. Однако, чтобы данная защита работала, ЭЛТ - монитор должен быть правильно заземлен.

Работа на ПЭВМ сопровождается постоянным и значительным напряжением функций зрительного анализатора. Одной из основных особенностей является иной принцип чтения информации, чем при обычном чтении. При обычном чтении текст на бумаге, расположенный горизонтально на столе, считывается работником с наклоненной головой при падении светового потока на текст. При работе на ПК оператор считывает текст, почти не наклоняя голову, глаза смотрят прямо или почти прямо вперед, текст формируется по другую сторону экрана, поэтому пользователь не считывает отраженный текст, а смотрит непосредственно на источник света, что вынуждает глаза и орган зрения в целом работать в несвойственном ему стрессовом режиме длительное время.

Расстройство органов зрения резко увеличивается при работе более четырех часов в день. Всемирная организация здравоохранения (ВОЗ) ввела понятие “компьютерный зрительный синдром” (КЗС), типовыми симптомами которого являются жжение в глазах, покраснение век и конъюнктивы, чувство инородного тела или песка под веками, боли в области глазниц и лба, затуманивание зрения, замедленная пере фокусировка с ближних объектов на дальние.

При продолжительной работе перед монитором желательно каждый час давать глазам отдохнуть: на протяжении десяти-пятнадцати минут покидать рабочее место, повторяя массаж глаз и различные зрительные тренировки, что позволит снизить напряжение и стресс и позволит восстановить кровообращение, так как при долгом нахождении в одном положении тело начинает затекать, что впоследствии может привести к осложнениям.

### **5.1.2 Правила техники безопасности и гигиены при работе на ПК**

Общие требования технической безопасности:

- быть дисциплинированным, внимательным;
- приступить к работе только в том случае, если знаете способ выполнения. Обратиться к работодателю при подозрении;
- не отвлекаться и не отвлекать своих коллег;
- не работать при плохой освещенности, также при отсутствиях освещения;
- не оставлять включенный компьютер;
- не приступать к работе влажными и мокрыми руками;
- не трогать руками сетевые соединители и провода;
- не затрагивать установки без разрешения;
- убрать посторонние предметы с рабочего стола;
- соблюдать чистоту на рабочем месте и в помещении.

Специальные требования технической безопасности:

1) перед началом работы:

- упорядочение рабочей одежды;
- удаление излишков на рабочем месте.

2) при работе:

- подключите компьютер и убедитесь, что неисправность отсутствует;
- при наличии неисправностей, отсоединить компьютер и сообщить мастеру;
- выполнение упражнений для пальцев во время технических перерывов;
- запрещается работать при открытом системном блоке, самостоятельно исправлять неисправности.

3) по окончании работы:

- правильно выключить компьютер;
- уборка личных вещей с документами;
- упорядочение рабочего места.

4) при пожаре:

- выключить компьютер;
- сообщить ситуацию администрации или мастеру;
- оперативное использование углекислого огнетушителя;
- позвонить на 101.

Правила работы на компьютере:

Большое значение имеет правильное размещение монитора и клавиатуры по высоте. В случае неудобства монитора может возникнуть головная боль, вызванная нагрузкой мышц спины и шеи. Существует одно простое правило: верхний край монитора должен быть визуально на одном уровне, а угол между этим уровнем и центром должен составлять 15 градусов.

На рисунке 5.1 представлено правильное и неправильное расположение монитора.

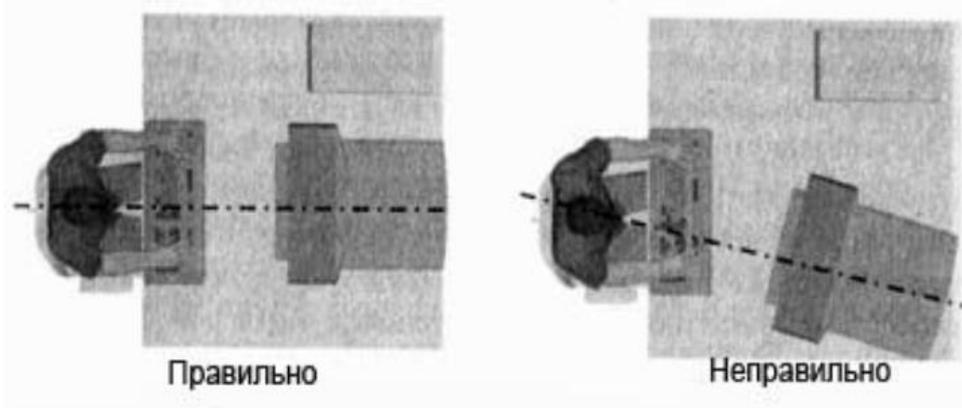


Рисунок 5.1 - Расположение монитора

1) держите спину прямо;

- 2) держите плечи свободными, локоть держат согнутым под прямым углом.
- 3) держите голову прямо, наклонив голову чуть вперед;

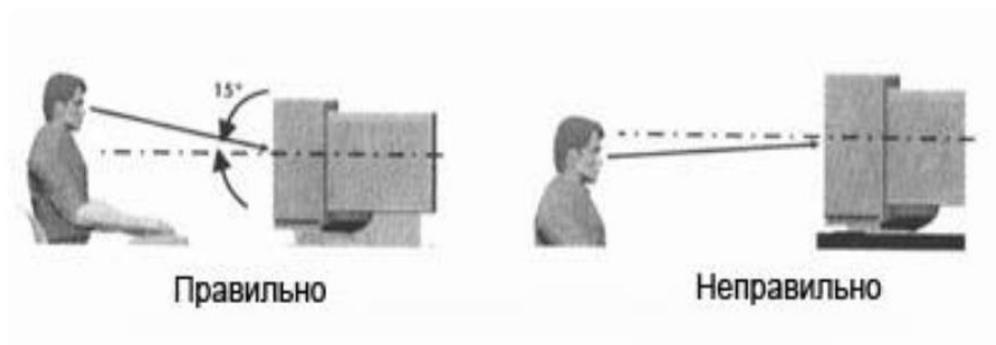


Рисунок 5.2 - Расположение высоты монитора

- 4) сиденье для сиденья должно быть мягким, колени, согнутые под прямым углом, стопы должны быть полностью прижаты к полу;
- 5) соблюдайте расстояние. Расстояние между глазом и экраном монитора должно быть не менее 70 см.

Ниже на рисунке 5.3 можно увидеть правильную схему работы за компьютером.

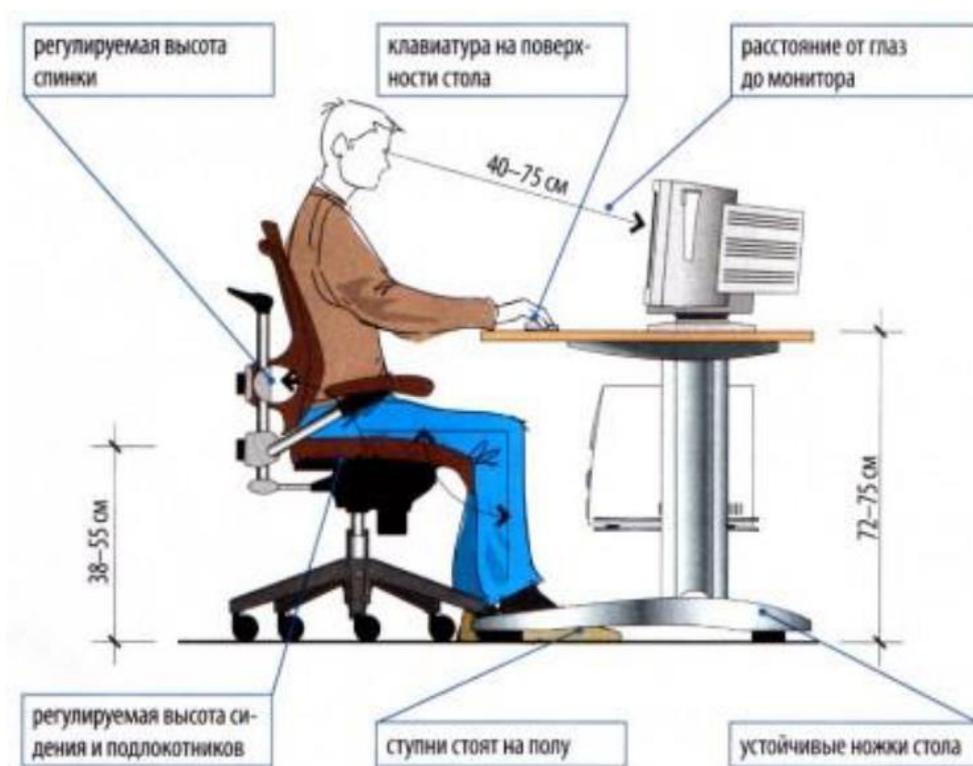


Рисунок 5.3 - Схема работы за компьютером

Необходимо правильно организовать рабочее время, придерживаясь некоторых правил:

- не допускается работать за компьютером более 5 часов в день;
- каждый час можно сделать упражнения для глаз, изменить вид деятельности, выпить чай, воду, стоя на месте. Если вы работаете "экстренно", чтобы не стоять, то, по крайней мере, вращайте голову, сгибайте мышцы;
- при перерыве проветрите вашу комнату.

## **5.2 Расчет показателей, удовлетворяющих требованиям техники безопасности и охраны труда при работе с проектируемой системой**

### **5.2.1 Пожароопасность**

Перечислим причины возникновения пожара:

- курение в запрещенных местах;
- неисправность электроприводов и установок;
- из-за перегрева проводов, установок, перенапряжения электрических сетей сверх норм;
- нарушение правил правильного хранения и использования легковоспламеняющихся жидкостей;
- неправильной эксплуатации тепловых сетей, печей;
- из-за безответственности людей;
- от грозы, иногда может самовозгореться.

Необходимый уровень безопасности пользователей ПЭВМ и обеспечение наиболее быстрого устранения ситуаций, угрожающих здоровью работника, достигнут путем организации комплекса мер по предупреждению и преждевременному устранению подобного рода ситуаций, в частности путем наличия средств защиты пожаротушения.

По взрывоопасности помещение, в котором происходит разработка дипломного проекта относится к категории "В", так как в нем имеются твердые горючие и трудно горючие вещества и материалы (бумага, стол, шкаф), способные гореть при взаимодействии с воздухом. В таблице 5.1 представлены подкатегории пожароопасных помещений.

Таблица 5.1 - Подкатегории пожароопасных помещений

Категория помещения	Удельная пожарная нагрузка $g$ на участке, МДж*м <sup>-2</sup>
В1	Более 2200
В2	1401 — 2200
В3	181 - 1400
В4	1 - 180

Определение пожароопасной категории помещения осуществляется путем сравнения максимального значения удельной временной пожарной

нагрузки на любом из участков с величиной удельной пожарной нагрузки [10].

Пожарная нагрузка ( $Q$ , МДж) помещений может включать в себя различные сочетания горючих, трудно горючих жидкостей, твердых горючих и трудно горючих веществ и материалов в пределах пожароопасного участка.

Она определяется по формуле

$$Q = \sum G_i * Q_i, \quad (5.1)$$

где  $Q$  – пожарная нагрузка, МДж;

$G_i$  – масса  $i$ -го материала пожарной нагрузки, кг;

$Q_i$  – низшая теплота сгорания  $i$ -го материала, МДж/кг.

Удельная пожарная нагрузка определяется по формуле (5.2):

$$g = Q/S, \quad (5.2)$$

где  $g$  - удельная пожарная нагрузка, МДж/м<sup>2</sup>

$Q$  - пожарная нагрузка, МДж

$S$  - площадь размещения пожарной нагрузки, м<sup>2</sup> (но не менее 10 м<sup>2</sup>).

Для помещения площадью  $S = 18 \text{ м}^2$

Пожарная нагрузка  $Q = 1900 \text{ МДж}$ .

$$g = 1900/18 = 105,5 \text{ МДж/м}^2.$$

Таким образом, удельная пожарная нагрузка по формуле 5.2 - 105,5 МДж/м<sup>2</sup>.

Следовательно, помещение относится к категории В4, согласно нормам пожарной безопасности. В целях защиты от возгорания используются огнетушители, исправность которых периодически проверяется.

### **5.2.2 Электробезопасность. Расчет защитного зануления ПЭВМ**

Опасное и вредное воздействия на людей электрического тока, электрической дуги и электромагнитных полей проявляются в виде электротравм и профессиональных заболеваний [11].

Степень опасного и вредного воздействия на человека электрического тока, электрической дуги и электромагнитных полей зависит от:

- рода и величины напряжения и тока;
- частоты электрического тока;
- пути тока через тело человека;
- продолжительности воздействия электрического тока или электромагнитного поля на организм человека;
- условий внешней среды.

Электробезопасность должна обеспечиваться:

- конструкцией электроустановок;
- техническими способами и средствами защиты;
- организационными и техническими мероприятиями.

Электроустановки и их части должны быть выполнены таким образом, чтобы работающие не подвергались опасным и вредным воздействиям электрического тока и электромагнитных полей, и соответствовать требованиям электробезопасности.

Технические способы и средства защиты, обеспечивающие электробезопасность, должны устанавливаться с учетом:

- а) номинального напряжения, рода и частоты тока электроустановки;
- б) способа электроснабжения (от стационарной сети, от автономного источника питания электроэнергией);
- в) режима нейтрали (средней точки) источника питания электроэнергией (изолированная, заземленная нейтраль);
- г) вида исполнения (стационарные, передвижные, переносные);
- д) условий внешней среды;
- е) возможности снятия напряжения с токоведущих частей, на которых или вблизи которых должна производиться работа;
- ж) характера возможного прикосновения человека к элементам цепи тока:

- однофазное (однополюсное) прикосновение;
- двухфазное (двухполюсное) прикосновение;
- прикосновение к металлическим нетоковедущим частям, оказавшимся под напряжением.;

з) возможности приближения к токоведущим частям, находящимся под напряжением, на расстояние меньше допустимого или попадания в зону растекания тока;

и) видов работ: монтаж, наладка, испытание, эксплуатация электроустановок, осуществляемых в зоне расположения электроустановок, в том числе в зоне воздушных линий электропередачи.

Для обеспечения защиты от поражения электрическим током при прикосновении к металлическим нетоковедущим частям, которые могут оказаться под напряжением в результате повреждения изоляции, применяют следующие способы:

- защитное заземление;
- зануление;
- выравнивание потенциала;
- систему защитных проводов;
- защитное отключение;
- изоляцию нетоковедущих частей;
- электрическое разделение сети;
- малое напряжение;
- контроль изоляции;
- компенсацию токов замыкания на землю;

- средства индивидуальной защиты.

Технические способы и средства применяют отдельно или в сочетании друг с другом так, чтобы обеспечивалась оптимальная защита.

Проведем расчет защитного зануления ПЭВМ.

Работа по созданию диплома должна производиться на персональном компьютере в течение четырех месяцев. Для обеспечения безопасной работы в проектно отделе необходимо установить автомат, отключающий ПЭВМ от сети. На рисунке 5.4 представлено Фазовое напряжение сети 220 В.

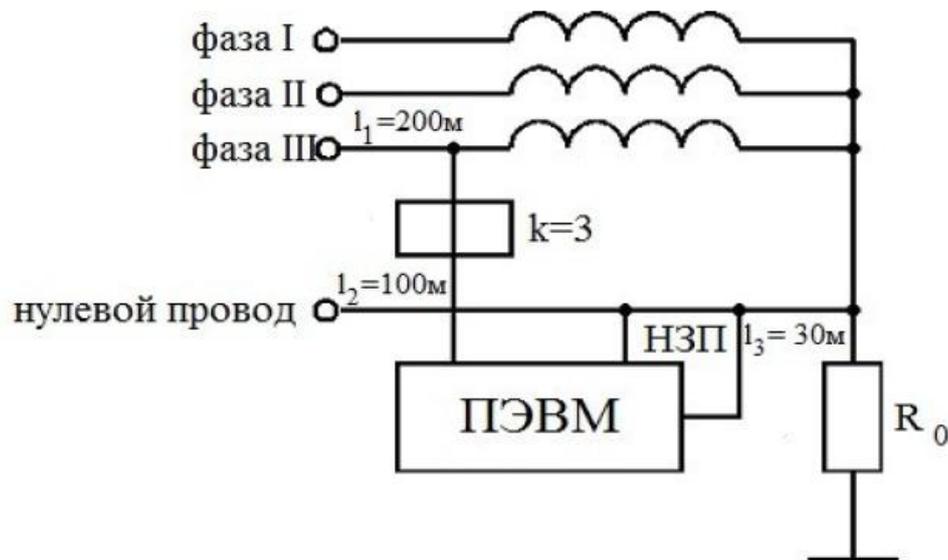


Рисунок 5.4 - Фазовое напряжение сети 220 В.

Исходные данные:

Напряжение на фазе:  $U_{\phi}=220В$ ;

Длины проводов:  $l_1=200 м$ ,

$l_2=100 м$ ,

$l_3=30 м$ ;

$r_{НЗП}$  - сопротивление нулевого защитного проводника. Определяется по формуле (5.3)

$$r_{НЗП} = \frac{\rho \cdot l}{S} [\text{Ом}], \quad (5.3)$$

где  $\rho$  - удельное сопротивление проводника, в нашем случае для алюминия  $[Al] = 0,028 \text{ Ом} \times \text{мм}^2/\text{м}$ ;

$l$  - длина проводника, м;

$S$  - сечение проводника,  $\text{мм}^2$  (в нашем случае  $S = 2 \text{ мм}^2$  для  $l_3$  и  $2,5 \text{ мм}^2$  для  $l_1$  и  $l_2$ );

$r_m$  - сопротивление катушек трансформатора  $[\text{Ом}]$  (в нашем случае, по паспорту,  $r_m=0,412 \text{ Ом}$ );

$K$ - кратность тока короткого замыкания к номинальному току выключателя ( $K = 3$  для помещений с нормальной средой).  
Ток короткого замыкания, определяется по формуле (5.4):

$$I_{K3} = \frac{U\phi}{\frac{r_m}{3} + R_{\text{общ}}}, \quad (5.4)$$

Для этого на нужно рассчитать  $R_{\text{общ}}$  по формуле(5.6):

$$R_{\text{общ}} = r_1 + r_2 + r_3, \quad (5.6)$$

По формуле (5.7) рассчитаем  $r_1, r_2, r_3$ :

$$r = \rho_1 * l_1 / S_1, \quad (5.7)$$

$$r_1 = \rho_1 * l_1 / S_1 = \frac{0,028 \times 200}{2,5} = 2,24 \text{ Ом},$$

$$r_2 = \rho_2 * l_2 / S_2 = \frac{0,028 \times 100}{2,5} = 1,12 \text{ Ом},$$

$$r_3 = \rho_3 * l_3 / S_3 = \frac{0,028 \times 30}{2} = 0,42 \text{ Ом}.$$

По формуле (5.6) находим  $R_{\text{общ}}$ :

$$R_{\text{общ}} = 2,24 + 1,12 + 0,42 = 3,78 \text{ Ом}.$$

Рассчитаем ток короткого замыкания, который определяется по формуле (5.4):

$$I_{K3} = \frac{U\phi}{\frac{r_m}{3} + R_{\text{общ}}} = \frac{220}{\frac{0,412}{3} + 3,78} = 56,2 \text{ А}.$$

Защита работника проектного отдела от напряжения прикосновения достигается путем отключения питания при помощи автоматического выключателя. Для его срабатывания необходимо, чтобы выполнялось следующее условие:

$$K \leq I_{K3} / I_H,$$

По найденной величине  $I_{K3}$  найдем номинальный ток выключателя  $I_H$ , для определения с каким номинальным током необходимо включить в цепь питания ПЭВМ автомат:

$$I_H = I_{K3} / K = 56,2 / 3 = 18,7 \text{ А}$$

Далее произведем проверку условия:  $K \leq I_{K3}/I_H \approx (3 \leq 56,2/18,7)$  – условие выполняется.

Вывод: для зануления необходимо использовать автоматический выключатель с током нагрузки 20 А, что обеспечит безопасность работ в проектноом отделе.

## Заключение

В ходе выполнения дипломного проекта был разработан голосовой помощник распознающий казахскую речь. Было проанализировано понятие голосового интерфейса, история возникновения голосовых помощников и проведено исследование существующих систем распознавания речи казахского языка. В результате был разработан голосовой помощник, который выполняет следующие команды:

- осуществляет поиск в интернете по запросу;
- открывает внешние приложения;
- выводит на экран список доступных команд;
- переводит слова;
- преобразовывает речь в текст;
- выводит и очищает историю запросов.

Также в ходе выполнения дипломного проекта были учтены не маловажные аспекты охраны труда, это правила техники безопасности и гигиены при работе на ПК, а также электробезопасность персонала. Одна из основных задач охраны труда заключается в обеспечении безопасности труда человека, т.е. создание таких условий труда. Было рассчитано защитное зануление ПЭВМ и пожароопасность. Результаты расчетов показали, что, для зануления необходимо использовать автоматический выключатель с током нагрузки 20 А, что обеспечит безопасность работ в проектной отделе.

На базе расчетов затрат на разработку системы доказана экономическая целесообразность и техническая возможность разработки и реализации системы. На основании оценки и анализа системы, системы разработана оптимальным образом исходя из критериев минимальных затрат, качества и простоты реализации.

Таким образом в данном дипломном проекте был разработан совершенно новый голосовой помощник, который распознает казахскую речь и выполняет множество команд на казахском языке.

## Список литературы

- 1 Операционная система Windows-Дизайнеры. URL: <https://sites.google.com/site/opersystwindows/home/dizajnerj>
- 2 Федорова Н.Н., Терехов С.А. – Параллельная реализация алгоритмов обучения нейронных сетей прямого распространения с использованием стандарта MPI. [sta@nine.ch70.chel.su](mailto:sta@nine.ch70.chel.su), 1998.
- 3 Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.:Символ-Плюс, 2011. – 992 с.
- 4 Рейтинг языков программирования 2020// <https://techrocks.ru/>: обучающий сайт. URL: <https://techrocks.ru/2020/02/08/programming-languages-rank-2020/>
- 5 Прохоренок Н.А. Создание оконных приложений на Python, 2011г.
- 6 Джоши П. Искусственный интеллект на Python, 2019г.
- 7 Методические указания к выполнению экономической части дипломных работ. Боканова Г.Ш.. – Алматы: АУЭС, 2020.
- 8 Методическое пособие для экономического обоснования дипломных работ по совершенствованию технологических процессов механической обработки детали. Н.В. Зубкова –ТГУ, 2006.
- 9 ГОСТ 12.0.003-2015. Система стандартов по безопасности труда Опасные и вредные производственные факторы. Классификация.
- 10 Методические указания по выполнению контрольной работы для студентов специальностей ФПЭ заочной формы обучения/ Т. В. Тупицына.– Хабаровск : Изд-во Тихоокеан. гос. ун-та, 2012-43 с.
- 11 Рабочее место. URL: <http://dic.academic.ru/dic.nsf/ruwiki/675479>

## Приложение А

(листинг программы)

```
# импорт необходимых модулей
import pyttsx3
import os
import random
import webbrowser
import time
import speech_recognition as sr
import pandas as pd
from tkinter import *
from fuzzywuzzy import fuzz
from colorama import *
from PIL import Image, ImageTk

# раздел глобальных переменных
text = "
r = sr.Recognizer()
engine = pyttsx3.init()
adress = "
j = 0
task_number = 0
ndel = ['айсулу', 'айсұлу', 'салем', 'жарайды', 'көмектесе аласыңба', 'өтінемін']
commands = ['сәлемдесу', 'сыртқы командаларды ашу',
            'компьютерды өшіру', 'іздеу', 'қоштасу', 'файлды аш',
            'командалар', 'аудармашы', 'компьютерді өшір', 'сау бол',
            'командалар', 'vk қос', 'ғаламторды аш', 'ютуб', 'музыканы өшір',
            'vkны аш', 'музыка қос', 'файлды тазала', 'файлды көрсет',
            'файлды аш', 'музыка', 'аудар', 'кино көру', 'сурет салу', 'атың кім',
            'сәлемдесу', 'компьютерды өшіру', 'ғаламтордан
            іздеу', 'қоштасу', 'калькулятор', 'командаларды көрсету', 'аудармашы']
def speak(what):
print (what)
# раздел описания функций команд
def pri_com(): # выводит на экран историю запросов
z = {}
mas = []
mas2 = []
mas3 = []
mas4 = []
```

*Продолжение приложения А*

```
file = open('commands.txt', 'r', encoding='UTF-8')
k = file.readlines()
for i in range(len(k)):
line = str(k[i].replace('\n', " ").strip())
mas.append(line)
file.close()
for i in range(len(mas)):
x = mas[i]
if x in z:
z[x] += 1
if not (x in z):
b = {x: 1}
z.update(b)
if not (x in mas2):
mas2.append(x)
for i in mas2:
mas3.append(z[i])
for i in range(1, len(mas3) + 1):
mas4.append(str(i) + ') ')
list = pd.DataFrame({
'command': mas2,
'count': mas3
}, index=mas4)
list.index.name = '№'
print(list)

def clear_analis(): # очистка файла с историей запросов
global engine
file = open('commands.txt', 'w', encoding='UTF-8')
file.close()
engine.say('Файл тазаланды')

def add_file(x):
file = open('commands.txt', 'a', encoding='UTF-8')
if x != "":
file.write(x + '\n')
file.close()
def comparison(x): # осуществляет поиск самой подходящей под запрос
функции
global commands, j, add_file
ans = "
for i in range(len(commands)):
```

## Продолжение приложения А

```
k = fuzz.ratio(x, commands[i])
if (k > 50) & (k > j):
    ans = commands[i]
    j = k
    if (ans != 'пока') & (ans != 'сәлем'):
        add_file(ans)
    return (str(ans))

def web_search(): # осуществляет поиск в интернете по запросу (adress)
    global adress
    webbrowser.open('https://yandex.ru/yandsearch?clid=2028026&text={ }&lr=11373'.
    format(adress))

def check_searching(): # проверяет нужно-ли искать в интернете
    global text, wifi_name, add_file
    global adress
    global web_search
    if 'іздеу' in text:
        add_file('іздеу')
        adress = text.replace('іздеу', '').strip()
        text = text.replace(adress, '').strip()
        web_search()
        text = ""
    elif 'іздеу' in text:
        add_file('іздеу')
        adress = text.replace('іздеу', '').strip()
        text = text.replace(adress, '').strip()
        web_search()
        text = ""
        adress = ""

def clear_task(): # удаляет ключевые слова
    global text, ndel
    for z in ndel:
        text = text.replace(z, '').strip()
        text = text.replace(' ', '').strip()

def hello(): # функция приветствия
    global engine
    z = ["Сызге калай комектесе аламын?", 'Кандай комек керек?', 'Салем Сызды
тындап турмын']
    x = random.choice(z)
```

## Продолжение приложения А

```
engine.say(x)
```

```
def name(): # диалог
    global engine
    z = ["Менын атым Айсулу"]
    x = random.choice(z)
    engine.say(x)
```

```
def quit(): # функция выхода из программы
    global engine
    x = ['тагы кездесемиз деп ойлаймын', 'комектескеныме куаныштымын', 'сау
    болыңыз']
    engine.say(random.choice(x))
    engine.runAndWait()
    engine.stop()
    os.system('cls')
    exit(0)
```

```
def show_cmds(): # выводит на экран список доступных команд
    my_com =
        ['1) сәлемдесу',
        '2) сыртқы командаларды ашу',
        '3) компьютерды өшіру',
        '4) ғаламтордан іздеу',
        '5) қоштасу',
        '6) файлды ашу',
        '7) командаларды көрсету',
        '8) сөз және сөйлемдерді аудару']
```

```
for i in my_com:
    print(i)
    time.sleep(2)
```

```
def brows(): # открывает браузер
    webbrowser.open('https://google.ru')
```

```
def ovk(): # открывает вк
    webbrowser.open('https://vk.com/feed')
```

```
def youtube(): # открывает ютюб
    webbrowser.open('https://www.youtube.com')
```

## Продолжение приложения А

```
def calc(): # открывает калькулятор
webbrowser.open('calc')

def paint(): # открывает ютуб
webbrowser.open('C:\ProgramData\Microsoft\Windows\StartMenu\Programs\Accessories\paint')
def shut(): # выключает компьютер
global quit
os.system('shutdown /s /f /t 10')
quit()

def musik(): # включает музыку
webbrowser.open('C:\ProgramData\Microsoft\Windows\StartMenu\Programs\Accessories')

def check_translate():
global text, tr
tr = 0
variants = ['аудар', 'аудару', 'аудармасы', 'аударшы']
for i in variants:
if (i in text) & (tr == 0):
word = text
word = word.replace('аудар', '').strip()
word = word.replace('аударып берші', '').strip()
word = word.replace('қалай аударылады', '').strip()
word = word.replace('аудармасы', '').strip()
word = word.replace('сөзін', '').strip()
word = word.replace('сөзді', '').strip()
word = word.replace('сөйлемін', '').strip()

webbrowser.open('https://translate.google.ru/#view=home&op=translate&sl=kk&tl=ru&text={}'.format(word))
tr = 1
text = ""
def cinema():
global text, tr
tr = 0
variants = ['кино', 'көру', 'қазақша']
for i in variants:
if (i in text) & (tr == 0):
word = text
```

## Продолжение приложения А

```
word = word.replace('кино', '').strip()
word = word.replace('көргім келеді', '').strip()
word = word.replace('қазақша', '').strip()

webbrowser.open('https://www.youtube.com/results?search_query=қазақша+кино
+көру={}'.format(word))
tr = 1
text = ""

cmds = {
'сәлем': hello, 'калькулятор': calc,
'компьютерді өшір': shut,
'сау бол': quit, 'командалар': show_cmds,
'vk қос': ovk, 'ғаламторды аш': brows,
'ютуб': youtube, 'музыканы өшір': musik, 'vkны аш': ovk,
'музыка қос': musik, 'файлды тазала': clear_analis,
'файлды көрсет': pri_com, 'файлды аш': pri_com, 'музыка': musik,
'аудар': check_translate, 'кино көру': cinema,
'сәлемдесу': hello, 'компьютерды өшіру': shut, 'ғаламтордан іздеу': brows,
'қоштасу': hello, 'командаларды көрсету': pri_com,
'аудармашы': check_translate, 'сурет салу': paint, 'атың кім':
name, 'іздеу': check_searching
}

# распознавание
def talk():
global text, clear_task
text = ""
with sr.Microphone() as source:
print('Сізді тыңдап тұрмын: ')
r.adjust_for_ambient_noise(source)
audio = r.listen(source, phrase_time_limit=3)
try:
text = (r.recognize_google(audio, language="kk-KZ")).lower()
except(sr.UnknownValueError):
pass
except(TypeError):
pass
os.system('cls')
lb['text'] = text
clear_task()
```

## Продолжение приложения А

```
# выполнение команд
def cmd_exe():
    global cmds, engine, comparison, check_searching, task_number, text, lb
    check_translate()
    text = comparison(text)
    print(text)
    check_searching()
    if (text in cmds):
        if (text != 'салем') & (text != 'сау бол') & (text != 'командалар'):
            k = ['Быр минут', 'Казыр орындаймын', 'орындап жатырмын']
            engine.say(random.choice(k))
            cmds[text]()
        elif text == "":
            pass
        else:
            print('Команда табылган жоқ!')
            task_number += 1
            if (task_number % 10 == 0):
                engine.say('Тагы тапсырманыз барма?')
                engine.runAndWait()
                engine.stop()

# исправляет цвет
print(Fore.YELLOW + ", end=")
os.system('cls')

# основной бесконечный цикл

def main():
    global text, talk, cmd_exe, j
    try:
        talk()
    if text != "":
        cmd_exe()
    j = 0
    except(UnboundLocalError):
        pass
    except(TypeError):
        pass

# раздел создания интерфейса
root = Tk()
```

### *Продолжение приложения А*

```
root.geometry('400x400')
canvas = Canvas(root, width=110, height=265)
canvas.pack()
pilImage = Image.open("kaz.jpg")
image = ImageTk.PhotoImage(pilImage)
imagesprite = canvas.create_image(50, 40, image=image)
root.configure(bg='pink')
root.title('Айсұлу')
root.resizable(False, False)

lb = Label(root, text=text)
lb.configure(bg='white')
lb.place(x=25, y=220, height=50, width=300)

but1 = Button(root, text='Тыңдау', command=main)
but1.configure(bd=1, font=('Castellar', 25), bg='green')
but1.place(x=200, anchor="se", y=350, height=50, width=150)

but2 = Button(root, text='Шығу', command=quit)
but2.configure(bd=1, font=('Castellar', 25), bg='gray')
but2.place(x=200, anchor="sw", y=350, height=50, width=150)
root.mainloop()
while True:
    main()
```