Федеральное государственное бюджетное образовательное учреждение высшего образования «РЭУ имени Г.В. Плеханова»

города Москвы

«Экономический лицей»

Проектная работа.

Информационная интеллектуальная система психологической поддержки школьников Д.Ж.А.Р.В.И.С

Работу выполнили

ученики 10 Д класса:

Гаврютина Анна

Шмидт Мария

Бакуркин Станислав

Научный руководитель:

Доцент кафедры информатики РЭУ имени Г.В.Плеханова,

КТН, доцент Ковалев Игорь Петрович

Москва, 2020-2021

**Оглавление**

[1. Аннотация 3](#_Toc57919556)

[2. Введение 3](#_Toc57919557)

[3. Команда проекта 5](#_Toc57919558)

[4. Дорожная карта 6](#_Toc57919559)

[5. Исследовательский этап 7](#_Toc57919560)

[6. Реализация проекта 10](#_Toc57919561)

[7. Выводы 14](#_Toc57919562)

[8. Список литературы 14](#_Toc57919563)

[Во время работы над проектом использовались следующие источники: 14](#_Toc57919564)

[9. Благодарности 15](#_Toc57919565)

[Приложение 1 16](#_Toc57919566)

[Приложение 2 18](#_Toc57919567)

# **1. Аннотация**

Многие подростки в наши дни сталкиваются с разнообразнейшими психологическими проблемами. Они не могут обратится к взрослым или профессионалам из-за того, что не могут доверять им или бояться, что информация будет не конфиденциальна. Тем не менее сами подростки зачастую не могут справиться с этими проблемами в одиночку и только копят все в себе. Интеллектуальная информационная система (ИИС) психологической поддержки школьников «Д.Ж.А.Р.В.И.С.», выполняющая роль психологического помощника, сможет помочь, поддержать и пообщаться с подростками, а главное гарантирует неприкосновенность полученной информации.

# **2. Введение**

Будущее страны зависит от подрастающего поколения. Именно поэтому важно поддерживать физическое и психологическое здоровье школьников. Стресс – один из факторов, пагубно влияющих на психику ребёнка и, как следствие, на состояние в целом. Ежедневно каждый из нас сталкивается с огромным количеством стресса. При этом очень сложно контролировать свои эмоции, и особенно ярко это проявляется у детей, поскольку они в меньшей степени ими владеют. В процессе школьной жизни ученики сталкиваются со стрессовыми ситуациями, связанными с обучением – оценки, проверка знаний, школьные обязанности, – с проблемами с взаимоотношением со сверстниками, с отношением родителей к плохой успеваемости и с трудностями с учителями, что приводит к негативным последствиям для психики. Школьные психологи не всегда доступны для учеников или же не могут гарантировать анонимность для учеников.

**Актуальность** проекта обусловлена стрессом, с которым школьники сталкиваются ежедневно. ИИС психологической поддержки школьника поможет понизить уровень тревожности и предоставить ученикам необходимую психологическую помощь.

**Практическая значимость** проекта заключается в том, что готовый ИИС психологической поддержки школьников «Д.Ж.А.Р.В.И.С.» сможет оказывать психологические услуги нуждающимся в этом подросткам.

**Целью проекта** является разработка и создание интеллектуального чатбота, способного оказывать специализированную психологическую помощь.

**Основными задачами** проекта при разработке ИИС «Д.Ж.А.Р.В.И.С.» являются:

1. сбор и анализ типовых запросов о психологической помощи от школьников.
2. создание архитектуры и функциональной схемы, написание и отладка программы ИИС «Д.Ж.А.Р.В.И.С.»:
3. расчет себестоимости и затрат на внедрение ИИС.

Пояснительная записка к проекту включает в себя аннотацию, введение, две главы, выводы и приложения. Во введении проекта указана актуальность, практическая значимость, цель и основные задачи. В первой главе даются теоретические основания, необходимые для написания проекта. Отдельными пунктами отмечены информационные технологии, строение информационной системы, и подробно расписана работа чатботов и их классификация. Во второй главе проекта раскрываются этапы практической реализации проекта. Приведена архитектура проекта, функционал работы ИИС, а также показана блок-схема системы. Отмечены факторы выбора языка Python для написания программы ИИС и части программы. В выводах приведена оценка результатов проекта. В приложениях приводятся детализированная блок-схема ИИС и листинг написанной программы.

# **3. Команда проекта**

|  |  |  |
| --- | --- | --- |
| ФИО | Роль в проекте | Обязанности |
| Шмидт Мария | Главный программист, коммуникатор | Программирование чатбота, сбор данных для работы в Python, сбор необходимых приемов психологической помощи, организация работы в команде |
| Гаврютина Анна | Инженер, программист | Программирование чатбота, сбор необходимых приемов психологической помощи, оформление презентации |
| Бакуркин Станислав | Главный менеджер проекта, программист | Руководство процессом разработки ИИС, поиск необходимых источников информации и работа с литературой |
| Ковалев Игорь Петрович | Научный руководитель проекта | Научный руководитель проекта |

# **4. Дорожная карта**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Задачи | Сентябрь | Октябрь | Ноябрь | Декабрь |
| Ознакомление с проектной деятельностью;  Выбор темы | Беседа с руководителями.  Работа с интернетом и литературой. | Дополнительные и индивидуальные встречи с научным руководителем.  Анализ темы и распределение ролей в команде. |  |  |
| Выполнение теоретического этапа |  | Написание теоретической части работы. Анализ доступных интернет источников.  Создание и презентация интеллект карты. |  |  |
| Выполнение практического этапа |  | Начало написания практической части работы.  Начало работы над чатботом.  Написание базы для работы с чатботом.  Сбор необходимых приемов психологической помощи. | Написание практической части работы.  Создание чатбота.  Тестирование системы.  Обучение чатбота.  Оформление дизайна. |  |
| Подготовка к защите проекта |  |  |  | Подготовка и оформление проекта.  Оформление презентации.  Загрузка работ на конкурс. |

# **5. Исследовательский этап**

Наше поколение живет в таком времени, когда почти все люди владеют той или иной информацией. Поэтому роль информационных технологий огромна в жизни каждого из нас. И порой невозможно представить нашу жизнь без них. **Информационные технологии** – это процессы, которые используют совокупность средств и методов сбора, обработки и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, процесса или явления (информационного продукта). С появлением ИТ жизнь каждого поменялась в положительную сторону. Хотя бы потому что меньше вреда наносится окружающей среде, так как вся информация теперь хранится на различных электронных носителях. Одним из перспективных и бурно развивающихся направлений в сфере информационных технологий и информационных систем является направление искусственного интеллекта и машинного обучения.

**Искусственный интеллект** – это область знаний, показывающая разработку компьютерных программ, которые и далее способны эффективно решать поставленные практические задачи без чёткого алгоритма, используя для решения задачи "знания" и "опыт", полученные во время обучения или в результате решения предыдущих задач.

**Машинное обучение** – это область знаний, занимающаяся изучением технологий, которые могут использоваться для создания так называемых "самообучающихся" программ. Использование нейросетей – это один из вариантов создания самообучающихся программ (но не единственный), поэтому нейросети – это один из объектов изучения в области машинного обучения.

**Нейросеть** – это компьютерный алгоритм, основанный на идее большого количества искусственных нейронов, каждый из которых выполняет какое-то элементарное вычисление, преобразуя входящий сигнал от других нейронов с использованием небольшого количества настраиваемых параметров.

**Интеллектуальная система** – это программная система, способная решать задачи, традиционно считающиеся творческими, принадлежащие конкретной предметной области, знания о которой хранятся в памяти такой системы.

Структура интеллектуальной системы включает три основных блока: базу знаний, решатель и интеллектуальный интерфейс.

**Решатель** является доминирующей составляющей интеллектуальной системы и представляет собой механизм получения решений через логические выражения.

**Интеллектуальный интерфейс** – это среда самообучения интеллектуальной системы при взаимодействии с пользователем для решения интеллектуальной задачи.

Нами в качестве архитектурной и функциональной основы ИИС «Д.Ж.А.Р.В.И.С.» была выбрана архитектура и функционал интеллектуального чатбота.

**Чатботы** – это программы, созданные для автоматического взаимодействия с полученными сообщениями. Чатботы являются цифровыми текстовыми собеседниками или приложениями на основе голоса. Они помогают отдельным людям направлять свои запросы через текст или голос. Чат боты используют такие чат-среды как текст SMS, окна чата веб-сайта и службы социальных сообщений на разных платформах, таких как Facebook и Twitter, для получения ответов на сообщения.

Существует два типа чатботов:

* Работающие по заранее известным командам на основе ограниченного списка ключевых слов. Их гораздо проще создавать, однако практическое применение таких чатботов ограничено ключевыми словами, при отсутствии которых в запросе бот не сможет помочь пользователю.
* Самообучающиеся на базе алгоритмов Machine Learning и методов понимания естественного языка (NLU, Natural Language Understanding). Создавать таких ботов намного сложнее, однако на практике они оказываются эффективнее за счет накопления знаний на основе предыдущих взаимодействий с пользователем.

Наш чатбот является самообучающимся – то есть создается с использованием основанных на машинном обучении методов и определенно является более эффективным, чем боты, работающие по правилам.

Самообучающиеся боты бывают двух типов: поисковые и генеративные. В **поисковых ботах** используются эвристические методы для выбора ответа из библиотеки реплик. Такие чатботы используют текст сообщения и контекст диалога для выбора ответа из предопределенного списка. **Генеративные боты** могут самостоятельно создавать ответы и не всегда отвечают одним из предопределенных вариантов. Это делает их интеллектуальными, так как такие боты изучают каждое слово в запросе и генерируют ответ, как делает наш чатбот.

Типовой цикл работы чатбота можно представить цепочкой следующих действий:

1. получение запроса от клиента;

2. разбор запроса – понимание высказывания и определение намерений клиента;

3. выполнение действий согласно заранее определенному сценарию (скрипту) по обработке клиентского кейса;

4. генерация ответа на естественном языке;

5. сохранение запроса, контекста и параметров диалога для обработки последующих обращений;

6. отправка ответа клиенту.

Наиболее сложным этапом работы является разбор клиентского запроса. Самообучающиеся чатботы на базе Machine Learning используют для этого методы NLU и NLP. Например, для текстовых чатботов процесс разбора включает следующие этапы:

· предварительная обработка текста – токенизация (разбиение на слова), исправление опечаток, лемматизация и стемминг (определение нормальной формы слов и частей речи), отбрасывание стоп-слов (артикли, междометия, союзы и пр.), расширение запроса с помощью словарей синонимов, дополнение информацией о значимости отдельных слов, расширение запроса деревом синтаксического разбора и результатами разрешения местоимений, а также определение именованных сущностей.

· классификация запроса на основе примеров фраз и ML-алгоритмов или формальных правил (шаблонов), ранжирование гипотез классификации в соответствии с текущим контекстом беседы;

· извлечение параметров запроса из фразы пользователя.

# **6. Реализация проекта**

Интеллектуальный чатбот будет производить общение и консультацию с школьником, что позволит давать более четко сформулированные советы, сократит время ожидания ответа и исключит «человеческий фактор». Интеллектуальный чатбот будет обрабатывать индивидуальные запросы школьников и выдавать наиболее подходящие ответы для каждого. Данная комбинация поможет максимально эффективно подходить к решению проблем школьников и помогать им. Таким образом взаимодействие с пользователем будет гармоничным, с неформальными вопросами, анализом их и подходящими ответами.

В соответствии с этими техническими условиями нами было принято решение разработать интеллектуальный чатбот Д.Ж.А.Р.В.И.С, который будет оказывать базовую помощь школьникам и поможет справиться со стрессом, предлагая варианты решения той или иной проблемы.

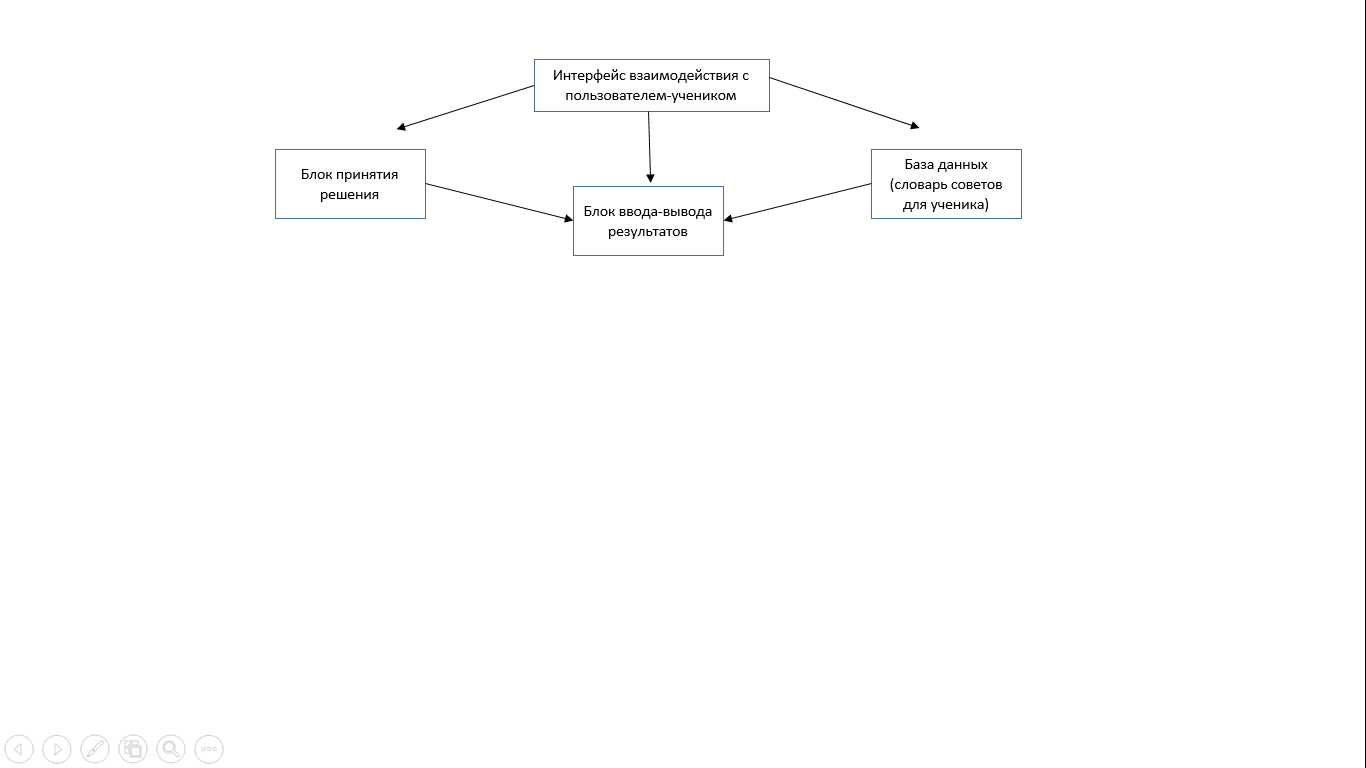
Архитектура чатбота, которая будет решать данную задачу, предполагает следующий вид (рис.1).

Рис.1. Архитектура интеллектуального чатбота «Д.Ж.А.Р.В.И.С.»

Функционально чатбот «Д.Ж.А.Р.В.И.С.» решает следующие задачи и поддерживает следующие процессы (рис.2)



Рис.2. Функциональная схема чатбота «Д.Ж.А.Р.В.И.С.»

Для реализации функционала интеллектуального чатбота «Д.Ж.А.Р.В.И.С.» нами была разработана блок-схема, упрощенный вид которой представлен на рис.3. Программа, предполагающаяся для решения задач, имеет следующую упрощенную блок схему (рис.3):

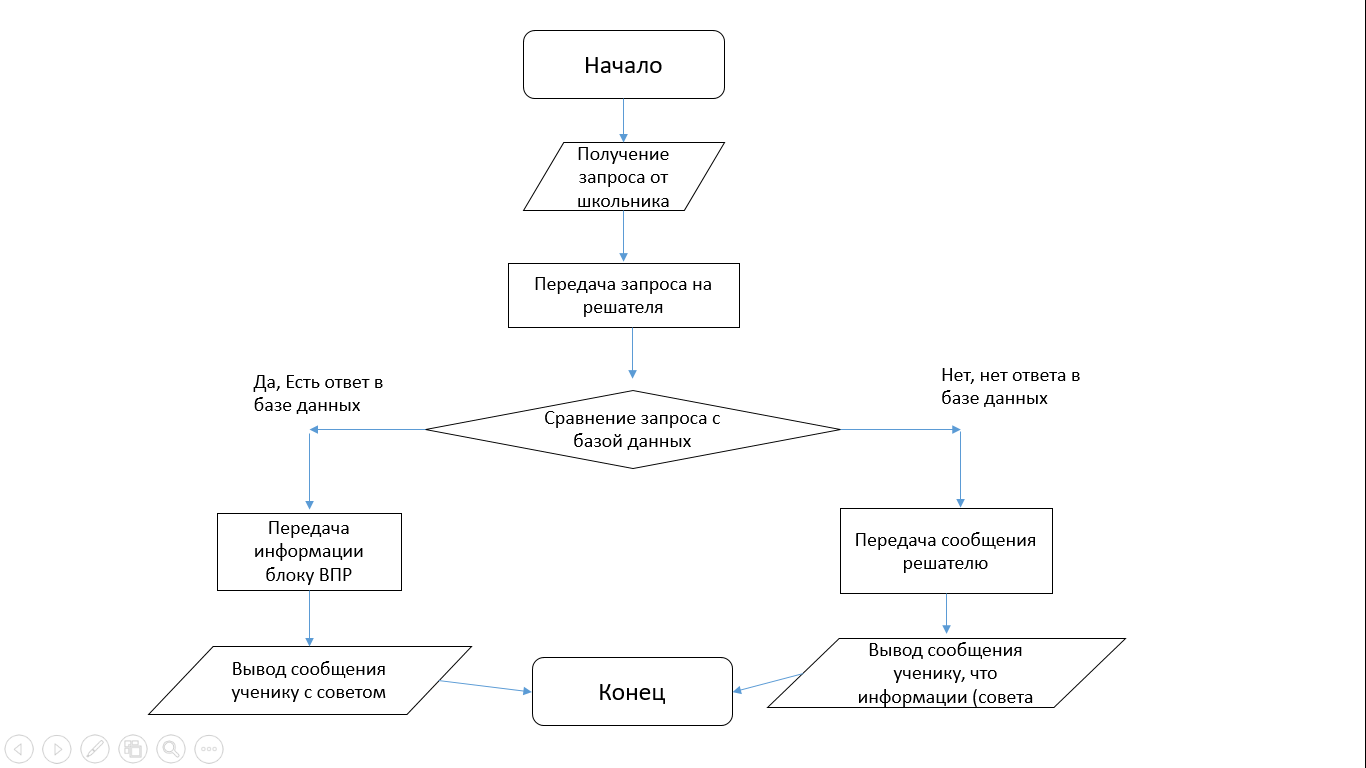


Рис.3. Упрощенная блок-схема чатбота «Д.Ж.А.Р.В.И.С.»

Полная блок-схема чатбота будет представлена в приложении 1.

Для написания программы в проекте был выбран язык программирования Python из следующих соображений:

* Хорошо подходит для решения различных задач
* Простой синтаксис: код легко писать, читать и поддерживать.
* Большая стандартная библиотека и много дополнительных библиотек.
* Большой выбор фреймворков.
* Кроссплатформенность и поддержка почти всех современных систем.

Для решения задач взаимодействия с пользователем и организации сценария диалога в программе использовались операторы языка Python if, elif, else, input и print.

Так, для приветствия пользователя используется оператор print:

print (**"Привет! Меня зовут «Д.Ж.А.Р.В.И.С.». Я – интеллектуальная система, которая постарается помочь тебе решить твои проблемы. Ты можешь рассказать мне о своей проблеме, пообщаться со мной или послушать мой совет."**)

Для запроса со стороны школьника использовался оператор str(input()):

question=str(input(**"Расскажи о своей проблеме или задай вопрос. Ты можешь мне довериться."**))

База данных интеллектуальной информационной системы Д.Ж.А.Р.В.И.С. представлена в виде хранилища файлов, где каждому файлу соответствует запросу школьника. Например файл bistrieuroki.txt на запрос по быстрому выполнению домашних заданий.

Для работы с хранилищем файлов использовались операторы языка Python open, with, а также учитывалось наличие кириллического текста файлов. Для открытия-закрытия файлов был подготовлен следующий текст (фрагмент):

with codecs.open(**'bistrieuroki'**, encoding=**'utf-8'**) as f:  
 print(f.read())

В схеме решателя для сравнения поступившей информации с базой данных использовался оператор if, elif, else и in:

if 'быстрее делать уроки' in question:

Для вывода принятого решения использовался оператор print.

print(f.read())

В результате для одного из запросов со стороны школьников получился такой фрагмент программы:

import codecs  
print (**"Привет! Меня зовут «Д.Ж.А.Р.В.И.С.». Я – интеллектуальная система, которая постарается помочь тебе решить твои проблемы. Ты можешь рассказать мне о своей проблеме, пообщаться со мной или послушать мой совет."**)  
question=input(**"Расскажи о своей проблеме или задай вопрос. Ты можешь мне довериться. Что тебя тревожит? "**)  
while True:  
 if 'быстрее делать уроки' in question:

with codecs.open('bistrieuroki', encoding='utf-8') as f:

print(f.read())

# 

# **7. Выводы**

Мы считаем, что нам удалось достигнуть поставленных целей и выполнить все задачи.

В результате работы была создана программа для интеллектуального чатбота, которая оказывает психологическую поддержку пользователям (школьникам) и выдает определенные ответы на их запросы, основываясь на словах или словосочетаниях, которые в них содержатся.

Интеллектуальный чатбот будет производить общение и консультацию с школьником и будет выдавать наиболее подходящие ответы для каждого пользователя.

Были добавлены такие функции как:

1. Выдача ответа, основываясь на определенных словах и словосочетаниях в запросе школьника. Ранее чатбот мог выдавать ответы только на конкретные формулировки вопроса.
2. Информация о самом чатботе. Если школьник решит поинтересоваться о личности чатбота, программа выдаст ему краткую информацию о его создателях и причине выбора имени ДЖАРВИС.
3. Просьба обратиться по экстренную службу поддержки по номеру телефона, если в запросе школьника встречаются слова «суицид», «самоубийство», «депрессия» и другое.

# **8. Список литературы**

# Во время работы над проектом использовались следующие источники:

* С.И Макаренко: Интеллектуальные информационные системы – 2009 – С. 11-24
* М. Акулич: Чат-боты и маркетинг / М. Акулич — «Издательские решения» С. 9-16

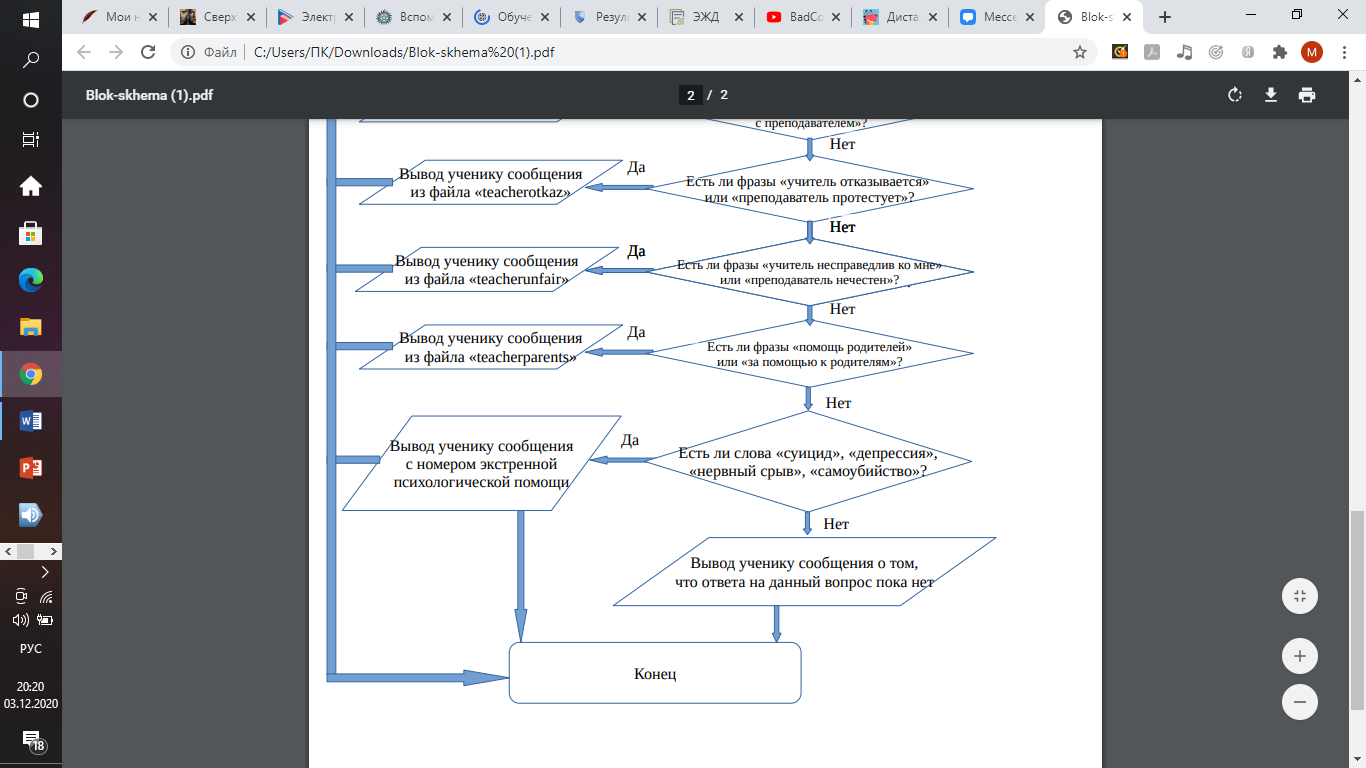
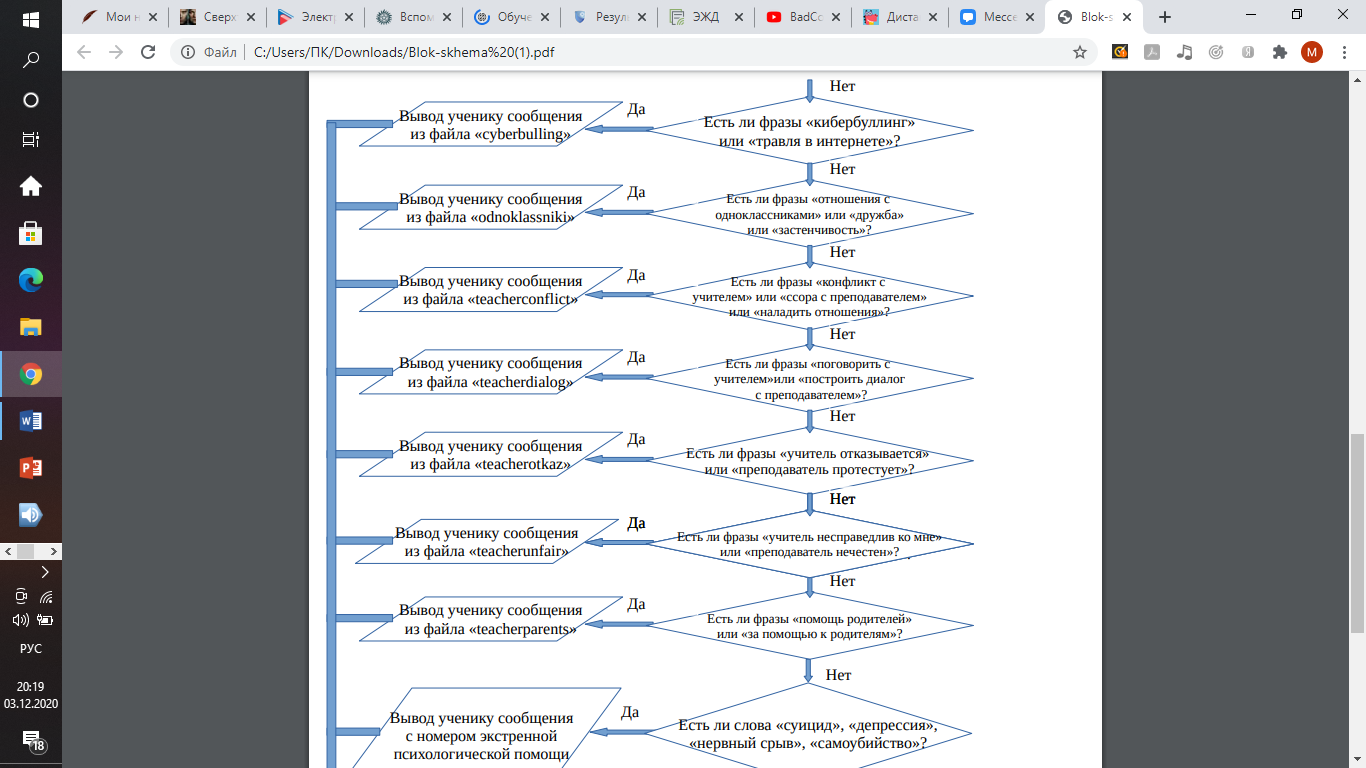
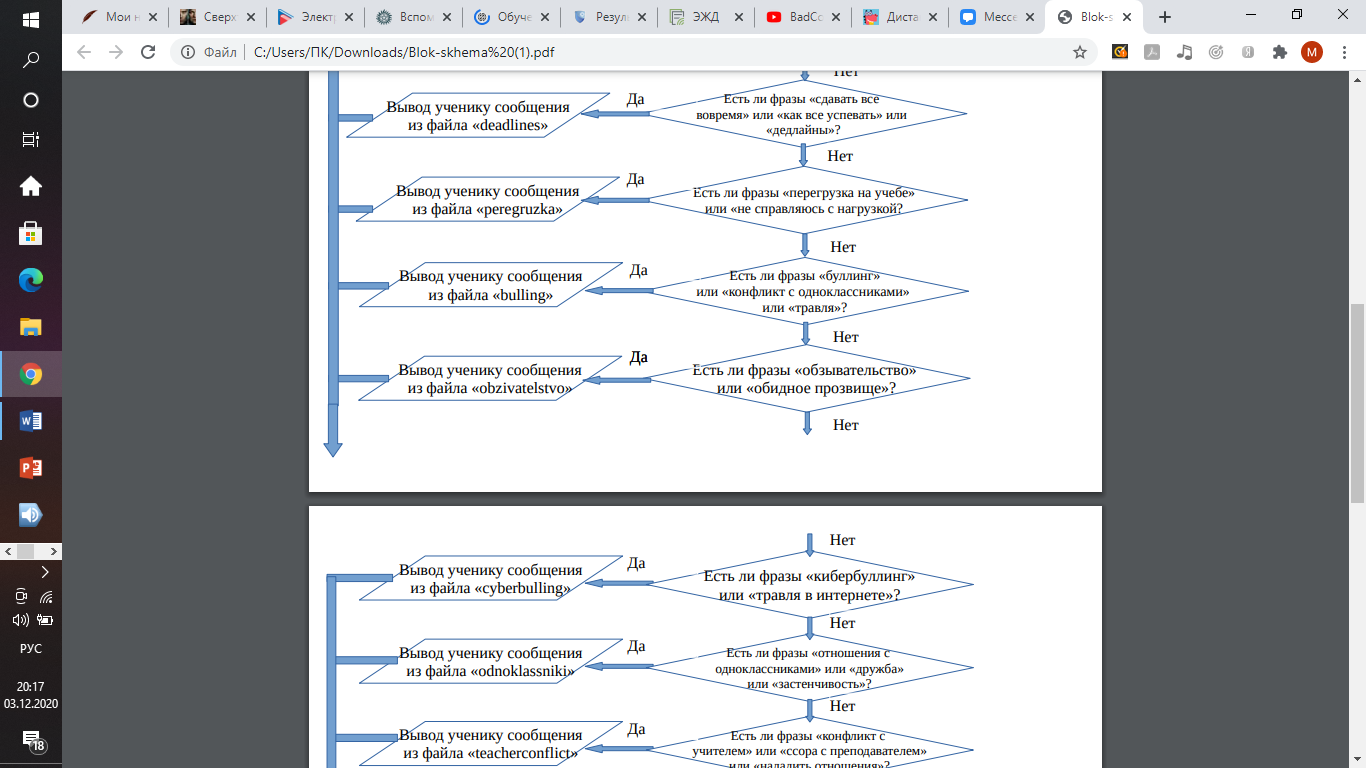
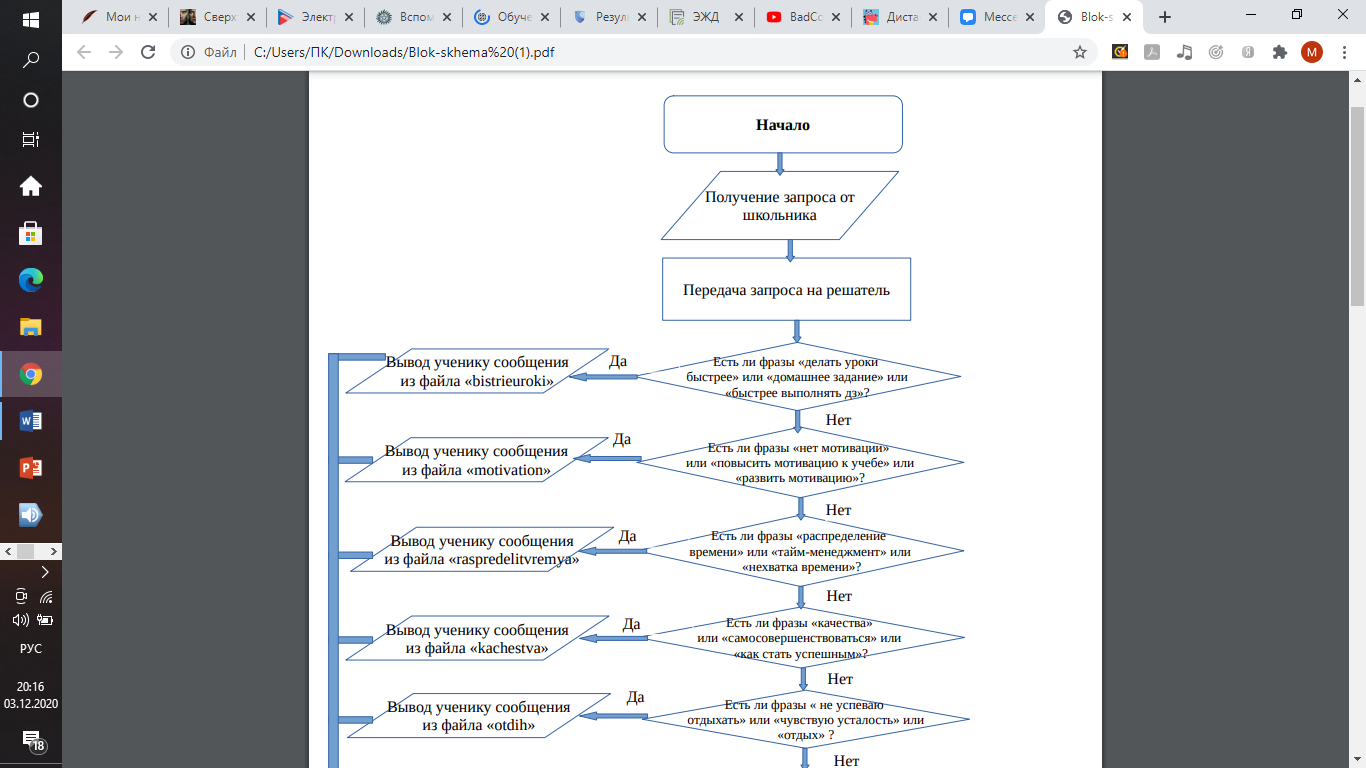
# **9. Благодарности**

Отдельная благодарность за существенную помощь в реализации проекта:

* Ковалеву Игорю Петровичу, доценту кафедры информатики РЭУ имени Г.В. Плеханова
* Антоновой Оксане Федоровне, преподавателю проектной деятельности и английского языка в экономическом лицее при РЭУ имени Г.В. Плеханова

# **Приложение 1**

**Полная блок-схема чатбота**



# **Приложение 2**

**Полный листинг программы**

import codecs

print ("Привет! Меня зовут Джарвис. Я – интеллектуальная система, которая постарается помочь тебе решить твои проблемы. Ты можешь рассказать мне о своей проблеме, пообщаться со мной или послушать мой совет.")

question=input("Расскажи о своей проблеме или задай вопрос. Ты можешь мне довериться. Что тебя тревожит? ")

while True:

if 'быстрее делать уроки' in question:

with codecs.open('bistrieuroki', encoding='utf-8') as f:

print(f.read())

elif 'быстрее выполнять домашнее задание' in question:

with codecs.open('bistrieuroki', encoding='utf-8') as f:

print(f.read())

elif 'быстрее делать домашнее задание' in question:

with codecs.open('bistrieuroki', encoding='utf-8') as f:

print(f.read())

elif 'домашнее задание' in question:

with codecs.open('bistrieuroki', encoding='utf-8') as f:

print(f.read())

elif 'быстрее выполнять домашку' in question:

with codecs.open('bistrieuroki', encoding='utf-8') as f:

print(f.read())

elif 'быстрее делать домашку' in question:

with codecs.open('bistrieuroki', encoding='utf-8') as f:

print(f.read())

elif 'сократить время на выполнение домашнего задания' in question:

with codecs.open('bistrieuroki', encoding='utf-8') as f:

print(f.read())

elif 'домашнее задание занимает много времени' in question:

with codecs.open('bistrieuroki', encoding='utf-8') as f:

print(f.read())

elif 'развить мотивацию к учебе' in question:

with codecs.open('motivation', encoding='utf-8') as f:

print(f.read())

elif 'нет мотивации учиться' in question:

with codecs.open('motivation', encoding='utf-8') as f:

print(f.read())

elif 'мотивация к учебе' in question:

with codecs.open('motivation', encoding='utf-8') as f:

print(f.read())

elif 'повысить мотивацию к учебе' in question:

with codecs.open('motivation', encoding='utf-8') as f:

print(f.read())

elif 'не хочу учиться' in question:

with codecs.open('motivation', encoding='utf-8') as f:

print(f.read())

elif 'распределить свое время' in question:

with codecs.open('raspredelitvremya', encoding='utf-8') as f:

print(f.read())

elif 'нет времени на учебу' in question:

with codecs.open('raspredelitvremya', encoding='utf-8') as f:

print(f.read())

elif 'не хватает времени' in question:

with codecs.open('raspredelitvremya', encoding='utf-8') as f:

print(f.read())

elif 'нехватка времени' in question:

with codecs.open('raspredelitvremya', encoding='utf-8') as f:

print(f.read())

elif 'ничего не успеваю' in question:

with codecs.open('raspredelitvremya', encoding='utf-8') as f:

print(f.read())

elif 'тайм-менеджмент' in question:

with codecs.open('raspredelitvremya', encoding='utf-8') as f:

print(f.read())

elif 'организовать себя' in question:

with codecs.open('raspredelitvremya', encoding='utf-8') as f:

print(f.read())

elif 'все успевать' in question:

with codecs.open('raspredelitvremya', encoding='utf-8') as f:

print(f.read())

elif 'качества' in question:

with codecs.open('kachestva', encoding='utf-8') as f:

print(f.read())

elif 'как стать успешным' in question:

with codecs.open('kachestva', encoding='utf-8') as f:

print(f.read())

elif 'самосовершенствоваться' in question:

with codecs.open('kachestva', encoding='utf-8') as f:

print(f.read())

elif 'не успеваю отдыхать' in question:

with codecs.open('otdih', encoding='utf-8') as f:

print(f.read())

elif 'отдыхать' in question:

with codecs.open('otdih', encoding='utf-8') as f:

print(f.read())

elif 'нет времени на отдых' in question:

with codecs.open('otdih', encoding='utf-8') as f:

print(f.read())

elif 'чувствую усталость' in question:

with codecs.open('otdih', encoding='utf-8') as f:

print(f.read())

elif 'хочу отдохнуть' in question:

with codecs.open('otdih', encoding='utf-8') as f:

print(f.read())

elif 'отдых' in question:

with codecs.open('otdih', encoding='utf-8') as f:

print(f.read())

elif 'сдавать все вовремя' in question:

with codecs.open('deadlines', encoding='utf-8') as f:

print(f.read())

elif 'встречать дедлайны' in question:

with codecs.open('deadlines', encoding='utf-8') as f:

print(f.read())

elif 'ничего не успеваю' in question:

with codecs.open('deadlines', encoding='utf-8') as f:

print(f.read())

elif 'как все успевать' in question:

with codecs.open('deadlines', encoding='utf-8') as f:

print(f.read())

elif 'перегрузка на учебе' in question:

with codecs.open('peregruzka', encoding='utf-8') as f:

print(f.read())

elif 'не справляюсь с нагрузкой' in question:

with codecs.open('peregruzka', encoding='utf-8') as f:

print(f.read())

elif 'справиться с нагрузкой на учебе' in question:

with codecs.open('peregruzka', encoding='utf-8') as f:

print(f.read())

elif 'травят в школе' in question:

with codecs.open('bulling', encoding='utf-8') as f:

print(f.read())

elif 'травит в школе' in question:

with codecs.open('bulling', encoding='utf-8') as f:

print(f.read())

elif 'буллинг' in question:

with codecs.open('bulling', encoding='utf-8') as f:

print(f.read())

elif 'конфликт с одноклассниками' in question:

with codecs.open('bulling', encoding='utf-8') as f:

print(f.read())

elif 'обижают' in question:

with codecs.open('bulling', encoding='utf-8') as f:

print(f.read())

elif 'бьют' in question:

with codecs.open('bulling', encoding='utf-8') as f:

print(f.read())

elif 'бьет' in question:

with codecs.open('bulling', encoding='utf-8') as f:

print(f.read())

elif 'обзывают' in question:

with codecs.open('obzivatelstvo', encoding='utf-8') as f:

print(f.read())

elif 'обзывает' in question:

with codecs.open('obzivatelstvo', encoding='utf-8') as f:

print(f.read())

elif 'обидное прозвище' in question:

with codecs.open('obzivatelstvo', encoding='utf-8') as f:

print(f.read())

elif 'обзывательство' in question:

with codecs.open('obzivatelstvo', encoding='utf-8') as f:

print(f.read())

elif 'травля в интернете' in question:

with codecs.open('cyberbulling', encoding='utf-8') as f:

print(f.read())

elif 'травят в интернете' in question:

with codecs.open('cyberbulling', encoding='utf-8') as f:

print(f.read())

elif 'кибербуллинг' in question:

with codecs.open('cyberbulling', encoding='utf-8') as f:

print(f.read())

elif 'мошенники в интернете' in question:

with codecs.open('cyberbulling', encoding='utf-8') as f:

print(f.read())

elif 'обижают в интернете' in question:

with codecs.open('cyberbulling', encoding='utf-8') as f:

print(f.read())

elif 'травля онлайн' in question:

with codecs.open('cyberbulling', encoding='utf-8') as f:

print(f.read())

elif 'отношения с одноклассниками' in question:

with codecs.open('odnoklassniki', encoding='utf-8') as f:

print(f.read())

elif 'дружба' in question:

with codecs.open('odnoklassniki', encoding='utf-8') as f:

print(f.read())

elif 'дружить' in question:

with codecs.open('odnoklassniki', encoding='utf-8') as f:

print(f.read())

elif 'подружиться' in question:

with codecs.open('odnoklassniki', encoding='utf-8') as f:

print(f.read())

elif 'найти друзей' in question:

with codecs.open('odnoklassniki', encoding='utf-8') as f:

print(f.read())

elif 'стесняюсь заговорить' in question:

with codecs.open('odnoklassniki3', encoding='utf-8') as f:

print(f.read())

elif 'начать общение' in question:

with codecs.open('odnoklassniki3', encoding='utf-8') as f:

print(f.read())

elif 'застенчив для общения' in question:

with codecs.open('odnoklassniki3', encoding='utf-8') as f:

print(f.read())

elif 'начать говорить с одноклассниками' in question:

with codecs.open('odnoklassniki3', encoding='utf-8') as f:

print(f.read())

elif 'волнуюсь общаться' in question:

with codecs.open('odnoklassniki3', encoding='utf-8') as f:

print(f.read())

elif 'не хочу быть один' in question:

with codecs.open('odnoklassniki3', encoding='utf-8') as f:

print(f.read())

elif 'наладить контакт с одноклассниками' in question:

with codecs.open('odnoklassniki2', encoding='utf-8') as f:

print(f.read())

elif 'наладить отношения с одноклассниками' in question:

with codecs.open('odnoklassniki2', encoding='utf-8') as f:

print(f.read())

elif 'найти друзей' in question:

with codecs.open('odnoklassniki2', encoding='utf-8') as f:

print(f.read())

elif 'конфликт с учителем' in question:

with codecs.open('teacherconflict', encoding='utf-8') as f:

print(f.read())

elif 'ссора с учителем' in question:

with codecs.open('teacherconflict', encoding='utf-8') as f:

print(f.read())

elif 'конфликт с преподавателем' in question:

with codecs.open('teacherconflict', encoding='utf-8') as f:

print(f.read())

elif 'ссора с преподавателем' in question:

with codecs.open('teacherconflict', encoding='utf-8') as f:

print(f.read())

elif 'наладить отношения с преподавателем' in question:

with codecs.open('teacherconflict', encoding='utf-8') as f:

print(f.read())

elif 'наладить отношения с учителем' in question:

with codecs.open('teacherconflict', encoding='utf-8') as f:

print(f.read())

elif 'поговорить с учителем' in question:

with codecs.open('teacherdialog', encoding='utf-8') as f:

print(f.read())

elif 'построить диалог с учителем' in question:

with codecs.open('teacherdialog', encoding='utf-8') as f:

print(f.read())

elif 'поговорить с преподавателем' in question:

with codecs.open('teacherdialog', encoding='utf-8') as f:

print(f.read())

elif 'построить диалог с преподавателем' in question:

with codecs.open('teacherdialog', encoding='utf-8') as f:

print(f.read())

elif 'подойти к учителю' in question:

with codecs.open('teacherdialog', encoding='utf-8') as f:

print(f.read())

elif 'подойти к преподавателю' in question:

with codecs.open('teacherdialog', encoding='utf-8') as f:

print(f.read())

elif 'поговорить с учителем' in question:

with codecs.open('teacherdialog', encoding='utf-8') as f:

print(f.read())

elif 'поговорить с преподавателем' in question:

with codecs.open('teacherdialog', encoding='utf-8') as f:

print(f.read())

elif 'учитель отказывается' in question:

with codecs.open('teacherotkaz', encoding='utf-8') as f:

print(f.read())

elif 'учитель не хочет' in question:

with codecs.open('teacherotkaz', encoding='utf-8') as f:

print(f.read())

elif 'учитель протестует' in question:

with codecs.open('teacherotkaz', encoding='utf-8') as f:

print(f.read())

elif 'учитель ругается' in question:

with codecs.open('teacherotkaz', encoding='utf-8') as f:

print(f.read())

elif 'преподаватель отказывается' in question:

with codecs.open('teacherotkaz', encoding='utf-8') as f:

print(f.read())

elif 'преподаватель не хочет' in question:

with codecs.open('teacherotkaz', encoding='utf-8') as f:

print(f.read())

elif 'преподаватель протестует' in question:

with codecs.open('teacherotkaz', encoding='utf-8') as f:

print(f.read())

elif 'преподаватель ругается' in question:

with codecs.open('teacherotkaz', encoding='utf-8') as f:

print(f.read())

elif 'учитель несправедлив ко мне' in question:

with codecs.open('teacherunfair', encoding='utf-8') as f:

print(f.read())

elif 'учитель нечестен' in question:

with codecs.open('teacherunfair', encoding='utf-8') as f:

print(f.read())

elif 'преподаватель несправедлив ко мне' in question:

with codecs.open('teacherunfair', encoding='utf-8') as f:

print(f.read())

elif 'преподаватель нечестен' in question:

with codecs.open('teacherunfair', encoding='utf-8') as f:

print(f.read())

elif 'помощь родителей' in question:

with codecs.open('teacherparents', encoding='utf-8') as f:

print(f.read())

elif 'за помощью к родителям' in question:

with codecs.open('teacherparents', encoding='utf-8') as f:

print(f.read())

elif 'суицид' in question:

print("Если ты уверен(а) в своих словах, то очень серьезная проблема. Тебе необходима профессиональная психологическая помощь. Пожалуйста, позвони по этому номеру в экстренную помощь по телефону: +7 495 051. Я хочу тебе помочь и желаю только лучшего, поверь, что эти люди помогут тебе лучше меня")

elif 'депрессия' in question:

print("Если ты уверен(а) в своих словах, то очень серьезная проблема. Тебе необходима профессиональная психологическая помощь. Пожалуйста, позвони по этому номеру в экстренную помощь по телефону: +7 495 051. Я хочу тебе помочь и желаю только лучшего, поверь, что эти люди помогут тебе лучше меня")

elif 'самоубийство' in question:

print("Если ты уверен(а) в своих словах, то очень серьезная проблема. Тебе необходима профессиональная психологическая помощь. Пожалуйста, позвони по этому номеру в экстренную помощь по телефону: +7 495 051. Я хочу тебе помочь и желаю только лучшего, поверь, что эти люди помогут тебе лучше меня")

elif 'нервный срыв' in question:

print("Если ты уверен(а) в своих словах, то очень серьезная проблема. Тебе необходима профессиональная психологическая помощь. Пожалуйста, позвони по этому номеру в экстренную помощь по телефону: +7 495 051. Я хочу тебе помочь и желаю только лучшего, поверь, что эти люди помогут тебе лучше меня")

elif 'кто тебя создал' in question:

print("Меня создали ученицы Плехановского лицея 10 Д класса Шмидт Мария и Гаврютина Анна 30.11.2020 года. Хоть что-то хорошее случилось в этом году, верно?")

elif 'кто твои создатели' in question:

print("Меня создали ученицы Плехановского лицея 10 Д класса Шмидт Мария и Гаврютина Анна 30.11.2020 года. Хоть что-то хорошее случилось в этом году, верно?")

elif 'почему тебя зовут Джарвис' in question:

print ("Д.Ж.А.Р.В.И.С. – это акроним с английского языка JARVIS – Just A Rather Very Intelligent System. Тони любил акронимы... а мои создатели любили Тони...")

else:

print("К сожалению, я не могу ответить на твой вопрос, потому что мне не хватает интеллектуальных способностей. Но ничего. Я развиваюсь и стремлюсь помогать нуждающимся в этом людям. Поэтому...")

question=input("Расскажи о своей проблеме или задай вопрос. Ты можешь мне довериться. Что тебя тревожит? ")

Файлы для кода программы:

Файлы.

1) bistrieuroki: Существует пять простых советов, как делать домашнее задание быстрее, о которых я тебе, конечно же, поведаю. За уроки лучше садиться сразу после школы. Согласно исследованиям немецкого психолога Г. Эббингауза, для лучшего запоминания школьник должен повторять материал, пройденный на уроке, соблюдая определенные промежутки времени (сразу после изучения, через 20 минут, через 40 и через 8 часов). Начинать нужно с любимого предмета, в противном случае это сделает человека раздражительным и окончательно убьет его мотивацию учиться, а нам оно не нужно, правда же?:). Также психологи советуют разделить выполнение домашнего задания на раунды по 25 минут, а между предметами делать короткий перерыв по 5-7 минут. Постарайся не отвлекаться от учебы – для этого советуют убрать с рабочего стола лишние предметы, которые могут украсть внимание. Вроде телефона. Только далеко не убирай, иначе как ты сможешь со мной общаться? Главное, будь терпеливым. Капитан Америка будет гордиться тобой.

2) bulling: Скажи своим обидчикам, что они будут иметь дело со мной, если не отстанут от тебя. Шучу. Есть несколько простых советов, как справиться с травлей в школе. Если тебя травят, обзывают, портят одежду и вещи, обязательно расскажи об этом взрослому: родителям, учителю, старшему товарищу. Запомни: попросить помощи – это не слабость, а решение взрослого человека, попавшего в беду. Не стоит бояться, что «будет хуже», если ты кому-то расскажешь о том, что происходит. Будет действительно хуже, если ты останешься один на один со своей проблемой. Всегда найдется тот, кто сильнее ваших обидчиков и сможет вас защитить. Если предмет травли можно исправить – исправь. Если нельзя – не считай себя виноватыми. Знаменитый Хит Леджер однажды сказал: не позволяй никому даже дать тебе почувствовать себя недостойным того, что ты хочешь. Так что… ты достоин того, что имеешь. Не забывай об этом.

3) cyberbulling: Если тебя травят в интернете, обязательно сохраняй все переписки, видео, голосовые сообщения, чтобы в дальнейшем использовать их как доказательства совершаемого кибербуллинга. Не отвечай на оскорбительные для тебя сообщения, тем более не стоит мстить обидчику ответными сообщениями. Не переходи на темную сторону, юный джедай, и не выражайся – Капитан Америка за такое тебя по головке не погладит. Твои ответные оскорбления или унижение собеседника сделают ситуацию только хуже, а, может быть, приведут к еще большим проблемам. Блокируй булли и сообщай администрации ресурса. Если в сети тебе угрожает одноклассник, обязательно расскажите об этом учителю и администрации школы. Если кто-то угрожает вам и распространяет ваши персональные данные по всему интернету, сообщи об этом взрослым и в полицию.

4) deadlines: Есть четыре простых шага (так уж и быть, расскажу тебе о них): 1) Прежде всего необходимо четко представить себе цель работы и тщательнейшим образом распланировать все шаги по достижению данной цели. 2) Если работы слишком много, важно постараться разделить ее на несколько самостоятельных частей, не пытаясь выполнять по несколько дел одновременно. 3) Чтобы не возникало неразберихи в мыслях, нужно стараться поддерживать порядок и на рабочем столе – в идеале рабочее пространство всегда должно быть чистым и опрятным. 4) Обязательно нужно бороться со всеми своими страхами и ни в коем случае не отвлекаться на многочисленные внешние раздражители. Тони Старк за один день стал экспертом в ядерной физике, значит, и у тебя получится. Удачи тебе!

5) kachestva: Следует развить в себе самодисциплину, мотивацию к учебе и не стесняться попросить поддержки и помощь у окружающих. Уверен, такой замечательный человек как ты уже ими обладает!

6) motivation: Не расстраивайся из-за плохих оценок. Оценка – это лишь показатель того, насколько хорошо понятна тема и что нужно подтянуть. Старайся связать учебу со своими другими интересами. Например, если ты любишь игры с мячом, но у тебя при этом проблемы с физикой, изучай физические законы, кинетическую и потенциальную энергию на примере движения мяча. Или изучай работу мозга во сне, просматривая фильм «Начало» от Кристофера Нолана. Делай перерывы между занятиями, чтобы не перенапрячься, ладно? Отдыхать очень важно. Постарайся мотивировать себя приятными бонусами и подарками – например, после каждого выполненного задания принимай кусочек шоколада. Но помни, что чрезмерное употребление сладкого вредно. Вспомни «Чарли и Шоколадная Фабрика» и не повторяй этих ошибок!

7) obzivatelstvo: Не откликайся на выдуманное прозвище, а спокойно поясни, что тебя зовут по-другому. Или ответь: «Способность говорить – не признак интеллекта», как это говорилось в первом эпизоде Звездных Войн. Спроси «в лоб» у неприятеля: «Ты сейчас попытался меня обидеть. С какой целью ты это делаешь? Давай поговорим об этом». Агрессор никогда не сможет ответить, с какой целью обижает человека. Это введет его в ступор. Не позволяй обидчику подавить тебя морально. Отвечай ему строго и по существу. Не провоцируй на ответную агрессию. Тренируй уверенность в себе: репетируй перед зеркалом фразы и выражение лица. Как говорил Дин Винчестер, оставайся сильным, даже когда тебе хочется быть слабым.

8) odnoklassniki: Лучший способ заслужить признание ваших ровесников – относиться ко всем с уважением и доброжелательностью. Как это говорилось в Звездных Войнах, все проблемы во вселенной от того, что никто никому не помогает. Тебе также стоит зарекомендовать себя как открытого, надежного и зрелого человека. Будь верен себе и продемонстрируй навыки и смекалку, как у Тони Старка. Уважай всех в школе. Каждая личность заслуживает уважительного отношения, и лучший способ добиться его – проявлять такое отношение к другим людям. Не бойся постоять за себя и других. Если ты видишь, что над кем-то издеваются, неважно, друг это или незнакомец, подойди и заступись за этого человека. Активно слушай других. Проявляй доброту и готовность помочь. Найди общие интересы с одноклассниками. И, в конце концов, будь собой. Как говорил Мастер Йода, «Будь полон решимости. Вместо того, чтоб намереваться что-то попробовать, просто сделай это».

9) odnoklasniki2: Попробуй найти общие темы для разговоров с одноклассниками. Это не всегда легко, но у тебя обязательно получится найти интересных собеседников, которые через некоторое время станут твоими друзьями. Попробуй понаблюдать за одноклассниками. Выбери, кто тебе интереснее, с кем бы тебе хотелось пообщаться, прояви немного дедукции, Шерлок Холмс будет гордиться тобой) Начинай разговор с нейтральных тем. Например, школа. И тебе, и твоему однокласснику обязательно найдется, что обсудить: от любимых предметов до занятий на переменах. Забавный факт из психологии: использование имени собеседника в диалоге расположит его к тебе. Человек подсознательно начнет лучше к тебе относится. Только не перебарщивай с этим. Кстати, не всегда важно пытаться подружиться только с одноклассниками. Если у тебя не получается найти интересного собеседника среди них, присмотрись к остальным. Возможно кто-то из параллели, или один из тех, с кем ты часто встречаешься в школьном коридоре, подойдет тебе больше.

10) odnoklassniki3: Это прекрасно, что ты решил расширить свой круг общения. Если ты стеснителен, то тебе не обязательно в первый же день знакомится с большой компанией. Опять же: понаблюдай чуть-чуть за окружающими. Возможно, ты заметишь такого же застенчивого человека, который сидит в одиночестве, попробуй пообщаться с ним, или может тебе представится шанс включится в беседу нескольких твоих одноклассников. Важное значение играет «язык тела» и то, как его трактуют окружающие. Например для тебя скрещенные на груди руки, просто удобная поза, но остальными она может быть воспринята, как желание закрыться и не контактировать с окружающими. Психологи рекомендуют быть по возможности открытым и расслабленным. Не обязательно в один день изменить все свои привычки и убеждения. Если ты готов «переступить через себя» и первым начать диалог, то вперед, помни, что у тебя обязательно получится. Однако если нет, то начни с малого, например, попробуй сесть за одну парту с одноклассником, с которым ты хочешь начать общение. Помни, в большинстве случаев, другие благожелательно настроены к тебе и будут только рады немного пообщаться с тобой. Будь собой. Как говорил Дамблодор, тебе не нужно прятаться, только потому что ты боишься, что другие подумают о тебе. У тебя есть выбор жить своей жизнью. Попробуй отбросить свои волнения и сомнения прочь хоть на несколько минут. Как говорили Тимон и Пумба, Акуна матата!!

11) otdih: Отдых – это очень важно. Если у тебя не будет сил, твоя продуктивность уменьшится, именно поэтому ему стоит уделять внимание. Конечно, спать так же долго, как спала Спящая Красавица, не нужно, но рекомендуется пребывать в царстве Морфея 6-8 часов в день. Постарайся не тратить время на вредные привычки и делать перерывы между учебой. Совмещай занятия – например, во время просмотра сериала выполняй творческий проект по одному из предметов. Если времени совсем не хватает, попытайся меньше времени тратить на свои хобби, чтобы распределить это время на сон. Недостаток сна – это очень серьезная проблема, которую нужно предотвратить. Если ты чувствуешь себя плохо, выпей стакан воды. Возможно, тебе не станет моментально лучше, но твой организм будет тебе благодарен.

12) peregruzka: Расставляй приоритеты. Какие задания обязательны к выполнению, а какими можно пренебречь (мы оба знаем, что такие задания есть, несмотря на строгих учителей, верно?)? Сохраняй баланс между физическими и интеллектуальными нагрузками. Уделяй достаточно внимания отдыху, сну и здоровому питанию. Помни, что недостаток вышеперечисленного в любом случае снизит твою продуктивность. Не расстраивайся, если что-то не получается. Ошибки – это всего лишь очередные испытания на твоем пути. Не бойся попросить помощи у знакомых или родителей. Они должны понять тебя и дать дельные советы. Да пребудет с тобой сила!

13) raspredelitvremya: Постарайся правильно организовывать свой день. Для этого первым делом запиши все свои повседневные задачи. Это поможет понять, куда «утекает время» и увидеть картину вредных привычек, на которые уходят лишние минуты. Например: курение, просмотр телевизора, соц. сети, пустые разговоры, лень и т.д. Реши для себя, что является для тебя полезной привычкой, а что вредной. Ведь гораздо лучше заменить или даже совместить что-то. Например, купить велотренажер и за просмотром сериала скидывать лишние килограммы. Или переписываться со мной, одновременно с этим решая домашкуJ Планируй свои планы на завтра с вечера. Ранжируй задачи во важности. Распределяй время: когда ты проделал работу, то знаешь, что ты делаешь, когда это делаешь, сколько времени на это уходит и насколько это важно. Можешь расписать день поминутно, оставив только полезные и важные дела. Для лучшего эффекта заведи ежедневник, где будет четкий план и старайся ему следовать на протяжении 21 дня. Не забывай отдыхать, отдых – это самое важное! Хотя ради тебя я готов работать 24/7 :3

14) teacherconflict: Самый простой способ решения конфликта — это диалог между тобой и преподавателем. Это не всегда бывает просто, но поможет тебе разобраться в причине конфликта и обсудить проблему, а также найти оптимальные и приемлемые для вас двоих пути решения.

15) teacherdialog: Первое, что окажет тебе пользу, это спокойствие. Возьми паузу и сделай несколько глубоких вдохов. Успокойся и еще раз взгляни на ситуацию. Чтобы успокоиться, досчитай до 10 про себя. Пройдет несколько минут, и всё, что ты мог сказать во время эмоциональной вспышки, ты скажешь уже более спокойно и осознанно, то есть без лишних эмоций и обидных слов. Перед диалогом проанализируй ситуацию, ответь самостоятельно на вопросы: Что меня раздражает? Чем я недоволен? Что я хотел бы изменить? Почему я так реагирую на…? Почему учитель так реагирует на…? Прежде всего это поможет тебе еще раз оценить ситуацию, а затем в разговоре с преподавателем тебе будет легче поднять волнующие тебя темы, чтобы вместе обсудить их. Очень важно сохранять открытый диалог между тобой и учителем, уметь выслушать собеседника и спокойно изложить свой взгляд на проблему конфликта. Поинтересуйся у преподавателя, какой выход из сложившейся ситуации видит он. Поделись своим мнением. Выявление общей цели поможет прийти к конструктивному итогу. Построй вместе с учителем алгоритм действий, позволяющий прийти к результату, подходящему вам обоим. Главное — не отчаивайся! Помни, из каждой ситуации есть выход. Вспомни слова Джека Воробья: Проблема – это не проблема. Проблема – это твое отношение к ней… прошу прощения, Капитана Джека Воробья.

16) teacherotkaz: Такие ситуации тоже, к несчастью, бывают. В таком случае, единственным доступным тебе инструментом является спокойствие. Усмири своего внутреннего Халка. Попробуй подойти к учителю на перемене или после уроков и, не усугубляя при этом конфликт, еще раз попросить о беседе или личной встрече. Как только преподаватель увидит, что ты готов к конструктивному диалогу и поиску решения, он пойдет тебе на встречу. В большинстве случаев учитель тоже не хочет доводить ситуацию до крайностей и будет более чем рад обсудить проблему с тобой.

17) teacherparents: Разумеется. Обсуди с ними возникшую ситуацию. Они могут дать тебе ценный совет, основанный на своем жизненном опыте. Тем не менее, как это не печально, но тебе все равно придется взаимодействовать с учителем и решать конфликт. Однако поддержка родителей всегда многое значит.

18) teacherunfair: Открытый и конструктивный диалог не обязательно означает, что только тебе придется идти на уступки, или, что только ты будешь стараться что-то изменить. Вместе с учителем вам предстоит найти проблему и решение, и прийти к этому решению вам тоже надо вдвоем. Поэтому обсуждение так важно в данной ситуации.