

Факультет «Информационные технологии»

Кафедра «Прикладная информатика и мультимедийные технологии»

**КУРСОВАЯ РАБОТА**

«Проектный практикум в предметной области»

***( по дисциплине «…», по модулю «….» или научно-исследовательская работа)***

на тему:

«Разработка чат-бота Телеграмм для самодиагностики уровня профессионального стресса»

Направление подготовки 09.03.03 Прикладная информатика

(код и наименование)

Направленность программы   Прикладная информатика в психологии

(наименование)

Группа 20 ИТ-ПИ(б/о)ПИП-1

Студент \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ В.О. Савинков

(подпись) (дата)

Объем заимствованного текста не превышает 20 %.

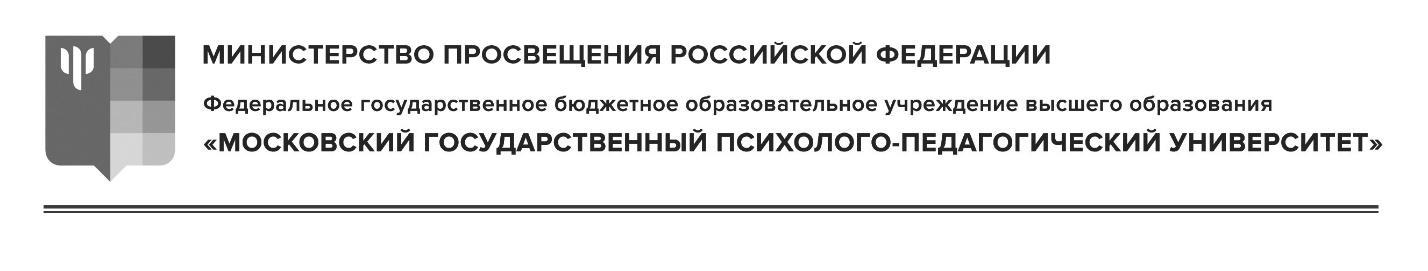
Допустить к защите \_\_\_\_\_\_\_\_\_ С.С. Ермаков руководителя

(подпись)

Руководитель курсовой работы \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ С.С. Ермаков

(подпись) (дата)

Москва, 2023



ОГЛАВЛЕНИЕ

[Введение 3](#_1fob9te)

[Глава 1. Профессиональное выгорание 5](#_2et92p0)

[1.1 Стресс 5](#_tyjcwt)

[1.2 Что такое профессиональное выгорание 8](#_1t3h5sf)

[1.3 Как узнать свой уровень профессионального выгорания 10](#_2s8eyo1)

[Глава 2. Программно-технический комплекс 13](#_17dp8vu)

[2.1 Написание программы 13](#_3rdcrjn)

[2.2 Настройка сервера 15](#_lnxbz9)

[2.3 Интерфейс программы 18](#_289o46ko5wg2)

[Глава 3. Проведение опросов 20](#_1ksv4uv)

[3.1 Проведение опроса с целью выявления корректности опросника К.Маслач 20](#_44sinio)

[3.2 Проведение опроса с целью выявления корректности работы программы 22](#_z337ya)

[Заключение 24](#_1y810tw)

[Список литературы 26](#_2xcytpi)

[Приложение 27](#_3whwml4)

# 

# Введение

Идею этой работы я придумал когда работал администратором и каждый день замечал как не самая сложная, но довольно однотипная работа высасывает из человека все соки. Мне стало интересно, а могу ли я использую свои знания помочь своим коллегам справляться с дистрессом и профессиональным выгоранием.

По мнению многих современных исследователей, в современном мире люди более чем раньше подвержены стрессам и выгоранию из-за огромного количества факторов, вот только некоторые из них:

1. Нехватка времени,
2. Часто приходиться делать не то что хочеться,
3. Человек не чувствуете уважения к себе,
4. Постоянные конфликты на работе или в семье,
5. Ощущение неудовлетворенностью жизнью.

Цель работы состоит в попытке формирования инструмента для раннего выявления у людей состояния профессионального выгорания и представления консолидированной информации для последующего анализа профильных специалистов и поиска/подбора ими вариантов преодоления состояния выгорания у людей. Для этого я решил выполнить следующий ряд шагов:

1. Telegram бота с тестом, который выявляет уровень профессионального выгорания,
2. Выгрузить бота на сервер, для того что доступ к нему был в любое время,
3. Добавить к боту SQL таблицы, для проведения в дальнейшем крупномасштабных опросов.
4. Написание мобильного приложения под Android и добавления данного бота в Vk, для охвата более крупной области польователей.

Стрессовые ситуации возникают как дома, так и на работе. С точки зрения управления, наибольший интерес представляют организационные факторы, которые вызывают стресс на рабочих местах. Знание этих факторов и уделение им особого внимания поможет предотвратить многие стрессовые ситуации и повысить эффективность управленческого труда, а также достичь целей организации с минимальными психологическими и физиологическими потерями персонала. Ведь именно стресс является причиной многих заболеваний, а значит, наносит ощутимый вред здоровью человека, тогда как здоровье – одно из условий достижения успеха в любой деятельности.

# 

# Глава 1. Профессиональное выгорание

## Стресс

Перед тем как дать определение профессиональному выгоранию и разобраться что оно из себя представляет, нужно ответить на следующий вопрос, что такое стресс? Данный вопрос я предлагаю разбить на следующие подвопросы:

1. Является ли он необходимым для человека или без него можно спокойно существовать?
2. Какие существуют виды стресса?
3. Какие существуют стадии стресса?

Одним из первых кто начал серьезно разбираться в данной области был Австро-Канадский патолог и [эндокринолог](https://ru.wikipedia.org/wiki/%D0%AD%D0%BD%D0%B4%D0%BE%D0%BA%D1%80%D0%B8%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F) Ганс Селье, и именно его работы легли в основу этой главы.

Стресс — это любое более или менее выраженное напряжение организма, связанное с его жизнедеятельностью. Речь идет о совокупности реакций организма, вызываемых воздействием различных интенсивных стимулов окружающей нас среды, трудными жизненными ситуациями. По своей начальной сути возникающие реакции организма имеют адаптационный характер. И в этом качестве стресс - неотъемлемое проявление жизни https://psyfactor-org.turbopages.org/psyfactor.org/s/lib/selye-stress-distress-2.htm.

По типу воздействия на человека стрессы можно подразделить на следующие виды:

1. Системные - отражающие напряжение преимущественно биологических систем. Они вызываются отравлением, воспалением тканей, ушибами и т. п.
2. Психические - возникающие при любых видах воздействий, вовлекающих в реакцию эмоциональную сферу.

Г. Селье выделяет три основные стадии развития стресса https://studfile.net/preview/6055368/page:3/:

1. Первая стадия - аларм-стадия, или стадия тревоги.
2. Вторая стадия - стадия резистентности, или сопротивления.
3. Третья стадия - стадия истощения.

Разберем каждую из стадий подробнее.

На первой стадии происходит мобилизация адаптационных ресурсов организма, человек находится в состоянии напряженности и настороженности.

Селье отмечает, что ни один организм не может постоянно находиться в состоянии тревоги. Если стресс-агент слишком сильный, организм погибает еще на стадии тревоги, в течение первых часов или дней. Если он выживает, то за первоначальной реакцией обязательно наступает стадия «резистентности»

Если стрессогенный фактор слишком силен или продолжает свое действие, то наступает стадия резистентности, которая характеризуется практически полным исчезновением признаков тревоги; уровень сопротивляемости организма значительно выше обычного. На этой стадии осуществляется сбалансированное расходование адаптационных ресурсов. Если стрессогенный фактор является чрезвычайно сильным или же действует длительно, развивается стадия истощения.

На стадии истощения энергия исчерпана, физиологическая и психологическая защиты оказываются сломленными. Вновь появляются признаки тревоги. В отличие от первой стадии, когда стрессовое состояние организма ведет к раскрытию адаптационных резервов и ресурсов, состояние третьей стадии больше похоже на “призыв о помощи”.

Селье выделяет два вида стресса — эустресс и дистресс. Дистресс всегда неприятен, он связан с вредоносным стрессом. Эустресс сочетается с желательным эффектом — активизируются психические процессы, эмоции носят стенический характер http://psyberia.ru/book/distress.pdf.

В отличие от дистресса, эустресса не следует избегать. Он является и приспособлением к изменениям среды, и мобилизацией сил для защиты организма. Полная свобода от стресса означает смерть. Селье считает возможным использовать стресс и наслаждаться им, если лучше понять его механизмы и выработать соответствующую философию жизни. Тогда стресс сопровождается приятными переживаниями радости, достижения, самовыражения.

В стрессе, вызванном необходимостью сосуществовать друг с другом, Ганс Селье видит главную причину дистресса. Приязнь и благодарность, ненависть и жажда мести отвечают за наличие или отсутствие дистресса в человеческих отношениях.

Один и тот же стрессор может вызвать неодинаковые последствия у разных людей. Селье связывает это с “факторами обусловливания”, которые избирательно усиливают или тормозят то или иное проявление стресса. “Обусловливание” может быть внутренним (генетическое предрасположение, возраст, пол) и внешним (прием внутрь гормонов, лекарственных препаратов, диета). Играет роль и реактивность организма, изменяющаяся в зависимости от внутренних и внешних условий. Одна и та же ситуация может вызвать тревогу у одного человека, фрустрацию у другого, конфликт у третьего. Кроме того, один и тот же человек способен проявлять разные состояния — от безразличия до психического срыва.

Особое внимание Селье уделяет проблеме мотивации в связи с проявлениями общего адаптационного синдрома. Отсутствие мотивации он считает самой большой душевной трагедией, которая разрушает все жизненные основы личности. Он подчеркивает значение мотивации - преимущественно в форме потребности свершений, которые всем принесут удовольствие и никому не сделают зла.

Подводя итог всему вышесказанному, хочется выделить следующие моменты:

1. Стресс является неотъемлемой частью жизни.
2. Стресс не всегда является вредоносным, часто он помогает адаптироваться к окружающей нас среде.
3. Отсутствие мотивации является одной из главных причин адаптационного синдрома, который довольно тесно связан с профессиональным выгоранием.

## 

## Что такое профессиональное выгорание

У людей, занятых в различных сферах коммуникативной деятельности, наблюдалось значительное снижение интереса к работе, выраженная склонность к конфликтам, деструктивному поведению, злоупотреблению алкоголем и другим вредным привычкам. Затем у них развились невротические расстройства или соматические заболевания. Это связано с влиянием профессиональной деятельности и адаптацией личности к окружающей https://onkto.ru/blog/psychology/vliyanie-vygoraniya-na-organizm-cheloveka.

Социальный психолог Маслач внесла новые характеристики “burnout”, под которым она понимала “Выгорание” как синдром и включало три компонента: эмоциональное истощение, деперсонализация (обезличивание человека), а также редукцию личных достижений - умаление собственного достоинства https://moluch.ru/archive/450/.

Психоэмоциональное истощение испытывают специалисты, которые в рамках своей деятельности постоянно общаются с другими людьми (области определения профессии, такие как “человек-человек”). Это сопровождается состоянием тревоги и разочарования из-за несоответствия определенному идеальному образу жизни. Такое поведение характеризуется появлением низкой самооценки, негативным отношением к профессиональной деятельности, потерей эмпатии и сопереживания в отношениях с окружающими https://research-journal.org/archive/7-121-2022-july/study-of-emotional-burnout-syndrome-in-representatives-of-various-professional-groups.

Эмоциональное выгорание является реакцией на непрерывный эмоциональный стресс и выражается в следующих характеристиках:

1. Снижение эмоциональной активности и умственных способностей; – холодность в общении с близкими людьми;
2. Негативное и циничное отношение ко всем участникам профессиональной деятельности: студентам, пациентам, клиентам, коллегам;
3. Растущее безразличие к профессиональным обязанностям;
4. Снижение деловой активности, снижение личной мотивации;
5. Снижение эмоционального фона;
6. Возникновение чувства профессиональной некомпетентности, неудовлетворенности результатами своей работы;
7. Деструктивная оценка результатов профессиональной деятельности и снижение личной ответственности за них;
8. Беспричинная агрессия, выраженное недовольство собой и окружающими вас людьми;
9. Ухудшение соматического состояния, головные боли, нарушения сна и т.д. снижение качества жизни в целом.

Традиционно факторы, вызывающие эмоциональное выгорание, группируются в два больших блока:

1. Внешние (организационные) факторы — к этим факторам относятся специфика профессиональной деятельности и окружающая среда. Это условия и содержание работы, различные социальные факторы и стимулы (материальные и нематериальные), специфика рынка труда, уровень ответственности и психологическая обстановка в коллективе, а также уровень справедливости;
2. Внутренние (личностные) факторы — это индивидуальные характеристики самих профессионалов. К таким факторам в первую очередь относятся личностный уровень психологической устойчивости (индекс стрессоустойчивости), уровень эмоциональной компетентности.

## 

## Как узнать свой уровень профессионального выгорания

Для выявления уровня профессионального выгорания я решил воспользоваться опросником уже упомянутой вскользь выше социальным психологом Карлой Маслач https://solnyshkochg.ru/docs/knyazeva/masluch.pdf.

Опросник состоит из 22 пунктов, по которым возможно вычисление значений 3-х шкал: “Эмоциональное истощение”, “Деперсонализация”, “Редукция профессиональных достижений”.

1. “Эмоциональное истощение” проявляется в переживаниях сниженного эмоционального тонуса, повышенной психической истощаемости и аффективной лабильности, утраты интереса и позитивных чувств к окружающим, ощущении “пресыщенности” работой, неудовлетворенностью жизнью в целом. В контексте синдрома перегорания “деперсонализация” предполагает формирование особых, деструктивных взаимоотношений с окружающими людьми.
2. “Деперсонализация” проявляется в эмоциональном отстранении и безразличии, формальном выполнении профессиональных обязанностей без личностной включенности и сопереживания, а в отдельных случаях – в негативизме и циничном отношении. На поведенческом уровне “деперсонализация” проявляется в высокомерном поведении, использовании профессионального сленга, юмора, ярлыков.
3. “Редукция профессиональных достижений” отражает степень удовлетворенности работника собой как личностью и как профессионалом. Неудовлетворительное значение этого показателя отражает тенденцию к негативной оценке своей компетентности и продуктивности и, как следствие, - снижение профессиональной мотивации, нарастание негативизма в отношении служебных обязанностей, тенденцию к снятию с себя ответственности, к изоляции от окружающих, отстраненность и неучастие, избегание работы сначала психологически, а затем физически.

Ответы испытуемого оцениваются следующим образом:

1. 0 баллов - “Никогда”,
2. 1 балл - “Очень редко”,
3. 2 балла - “Редко”,
4. 3 балла - “Иногда”,
5. 4 балла - “Часто”,
6. 5 баллов - “Очень часто”,
7. 6 баллов - “Каждый день”;

К шкале Эмоциональное истощение относятся следующие вопросы (1, 2, 3, 8, 13, 14, 16, 20 и зеркальный бал 6 вопроса).

К шкале Деперсонализация относятся следующие вопросы (5, 10, 11, 15, 22)

К шкале Редукция профессионализма относятся следующие вопросы (4, 7, 9, 12, 17, 18, 19, 21).

Оценка уровня выгорания для:

1. Эмоционального истощения
   1. Низкий уровень 0 -15,
   2. Средний уровень 16 -24,
   3. Высокий уровень 25 и больше;
2. Деперсонализации
   1. Низкий уровень 0 -5,
   2. Средний уровень 6 -10,
   3. Высокий уровень 11 и больше;
3. Редукции профессионализма
   1. Низкий уровень 37 и больше,
   2. Средний уровень 31 -36,
   3. Высокий уровень 30 и меньше.

Данный опросник предназначен для определения степени выраженности и распространенности синдрома выгорания (перегорания) среди профессионального контингента. Методика может быть использована в практическом здравоохранении, образовании, бизнесе и управлении для своевременной и всесторонней диагностики синдрома перегорания с последующим определением мишеней и уровней профилактического воздействия, как в направлении оптимизации терапевтической среды, так и в отношении профилактики накопления эмоционального напряжения. Полученные данные могут быть рекомендованы при подготовке руководящих кадров с информированием и тренингом построения программ профилактической санации производственной обстановки в коллективе. Методика может применяться при проведении профессионального отбора

# *Глава 2.* [*Программно-технический комплекс*](#_17dp8vu)

## Написание программы

Языком программирования было решено выбрать python, так как уже имелся опыт создание небольших telegram ботов на данном языке. Перед преступлением к написанию самой программы, было решено для начало определить из каких компонентов будет состоять программа. В итоге она состоит из 5 компонентов:

1. Answer - в данной папке находятся файлы, в которых этапы проводимого пользователем теста.
2. Bot commands - в данной папке находятся файлы, в которых прописаны все доступные команда и поведение программы при их вводе.
3. Keyboard - в данной папке находятся файлы, в которых прописаны состояния клавиатуры, её внешний вид в разных условиях.
4. Tests - в данной папке находятся файлы, в которых прописаны непосредственного сами тесты доступные для прохождения пользователю.
5. Main - в данном разделе находится текстовые файлы, в которые записываются результаты прохождения теста, файл, отвечающий за запуск программы и файл, отвечающий за хранение токена для telegram бота.

Далее нужно было определиться какую библиотеку выбрать ядром будущей программы:

1. Aiogram - относительно новая библиотека, которая быстро набирает популярность. Она является асинхронной, это означает что код не останавливает работу во время ожидания ответа от пользователя, кроме того, она обладает довольно обширной локализованной документацией на русском языке.
2. Python-telegram-bot - в отличие от aiogram, python-telegram-bot долгое время поддерживал только синхронным подходом к работе, то есть при ожидании ответа от пользователя выполнение кода останавливается, и только относительно недавно разрешил использование и асинхронного подхода, в связи с чем нередки ошибки из-за путаницы. Из плюсов можно выделить более подробную документацию по сравнению с aiogram.
3. TeleBot - данная библиотека используется для создания простых проектов. В моем случае она не подходит, так как в дальнейшем к ней будет проблематично присоединить весь интересующий меня функционал.
4. Telethon - используется для написания ботов “управляющих” аккаунтом. Например, одна из сфер применения этого - так называемые спам боты. В моем случае эта библиотека мне категорически не подходит.
5. No code - используется при создание очень простых программ, без написания непосредственно кода “если пользователь сказал X, то программа ответила Y”. В моём случаи это не подходит.

В конечном счеты было принято решения написание программы, в основе которой будет лежать библиотека Aiogram.

## 

## Настройка сервера

Для постоянного доступа к боту мною был приобретен самый простой виртуальный сервер на сайте <https://cp.omro.host/services> со следубшими характеристаками:

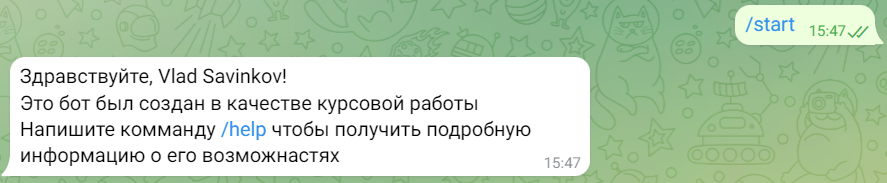
1. Micro SSD,
2. VMmanager - внешняя панель управления сервером,
3. Пользователь: root.

Далее, для запуска бота на сервере я выполнил следующие шаги:

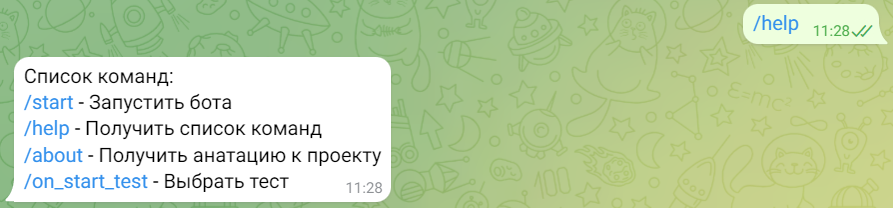
1. Создал виртуальную машину с операционной системой linux и образом iso ubuntu-22.04.2-desktop-amd64.
2. Запустил терминал:
   1. Далее подключился к серверу через ssh протокол:
      1. ssh root@ip address;
   2. Создал специальную директорию в домашнем каталоге, для этого использую следующие команды:
      1. cd ../,
      2. cd home,
      3. mkdir telegram\_bot;
   3. Теперь нужно создать виртуальное окружение, а для этого нужно установить все необходимые библиотеки:
      1. apt instal sudo,
      2. sudo apt update,
      3. sudo apt install python3-venv;
   4. Теперь перемещаюсь в директорию проекта и создаю виртуальное окружение для проекта:
      1. cd telegram\_bot,
      2. python3 -m venv venvBot,
      3. source venvBot /bin/activate,
      4. pip install requests,
      5. pip install wheel,
      6. pip install aiogram,
      7. pip install pytlegramboyapi;
3. Теперь нужно переместить файлы с компьютера на сервер:
   1. Запускаю Firefox и захожу в свою учётную запись github и скачиваю свою программу, которую я заранее туда загрузил,
   2. Устанавливаю FileZilla и подключаюсь к серверу:
      1. Ввожу ip адрес своего сервера, пароль, пользователя (в моем случаи это root) и выбираю 22 порт;
   3. В FileZilla захожу в папку telegram\_bot и перемещаю туда все файлы своего проекта;
4. Теперь можно возвращаться в терминал и запускать сервер:
   1. source venvBot /bin/activate,
   2. python3 main.py;
5. Готово, программа работает, но есть одно но, в этом случае при закрытии терминала перестанет работать и сервер. Для того чтобы сервер продолжал работал даже при выключенной машине, запущу виртуальную сессию через программу screen, для этого выполнил следующий ряд шагов:
   1. Заново запустил терминал,
   2. ssh root@ip address,
   3. sudo apt install screen,
   4. cd ../,
   5. cd home,
   6. cd telegram\_bot,
   7. screen;
6. Теперь можно смело запускать сервер:
   1. source venvBot /bin/activate,
   2. python3 main.py.
7. Готово, программа опять работает, но теперь можно смело выключать компьютер, на котором и установлена виртуальная машина, не переживая о том, что программа перестанет работать.

## Интерфейс программы

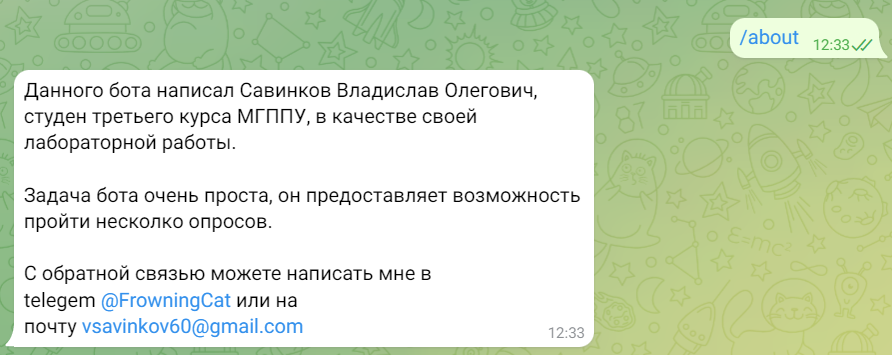
При первом запуске бота пользователю необходимо ввести команду “start”, после чего он получить вводную информацию по данному проекту, а для более подробной информации он может написать “help”



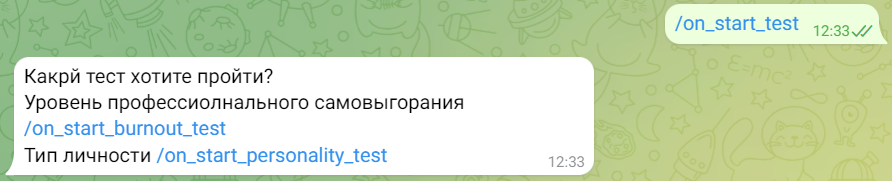
При введение этой команды пользователь получит небольшое меню, с набором самых основных команд, рассмотрим каждую из них поподробнее.



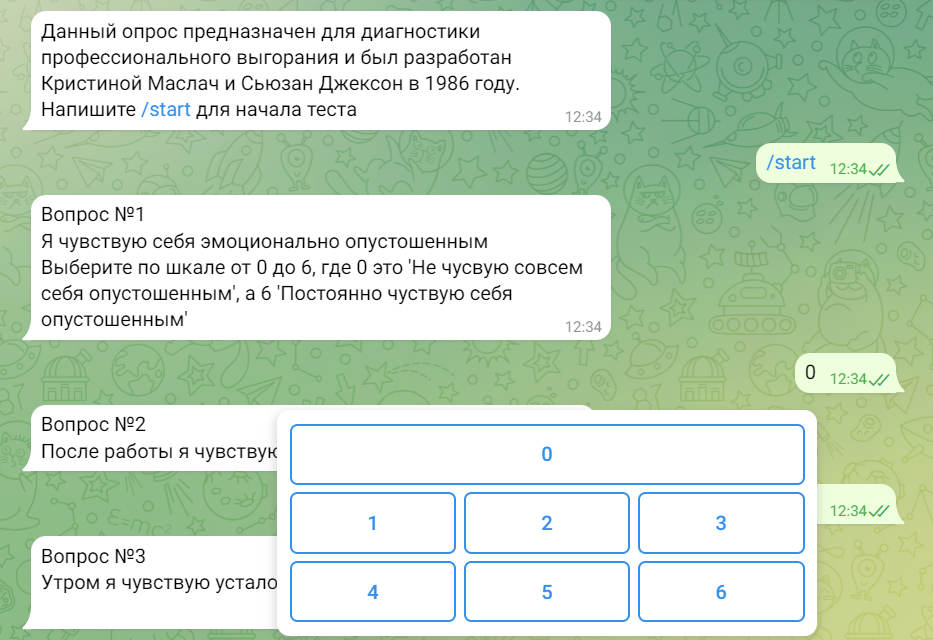
Команда “about” предоставляет пользователю информацию о создателе данной программы.



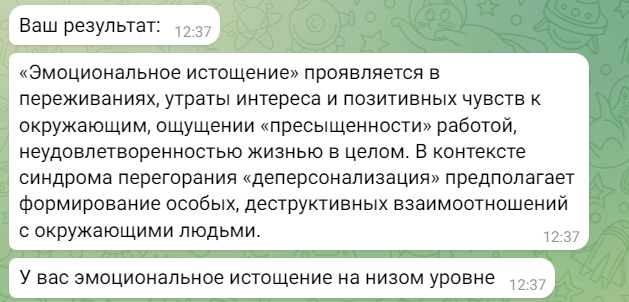
Команда “on\_start\_test” предоставляет пользователю возможность выбрать тест для прохождения.



После выбора теста пользователь получает небольшую вводную информацию после которой начинается сам тест. В данном тесте реализована пользовательская клавиатура для более удобного прохождения теста.



При завершении тестирования пользователь получает вердикт на основе своих ответов.



# Глава 3. Проведение опросов

## Проведение опроса с целью выявления корректности [опросника](#_44sinio) К.Маслач

Во время изучения опросника К.Маслач меня не покидала мысль, а вдруг он уже неактуален, что если результаты, полученные в ходе опроса окажутся неверными? Если мои опасения окажутся правдивыми, то люди решившие пройти мой тест получат некорректные вердикт в сторону их эмоционального истощения, и в лучшем случае окажутся просто недовольными, а в худшем могут не узнать о необходимости начать делать изменения в своей жизни на фоне высокого истощения.

Поэтому, я решил сходить в школу, в которой учился с 9 по 11 класс, и провести небольшое исследование среди учителей с целью ответить на следующий вопрос “А работоспособен ли данный опросника”. Этот вопрос я разбил на следующие вопросы:

1. На сколько актуален/полезен каждого из вопросов?
2. На сколько велика необходимость добавление новых вопросов?
3. Как хорошо опрашиваемые понимают сами вопросы?
4. Как хорошо опрашиваемые понимают, как нужно оценивать каждый из вопросов?

Так как опрос довольно большой и провести его перемену довольно сложно, было принято решение прийти в школу во время весенних каникул. Для этой цели я договорился со своей бывшей классной руководительницей, и она оформила мне пропуск на 10 апреля.

За несколько дней до эксперимента я выделил ряд условий, при достижении которых эксперимент будет считаться удачным:

1. Я нашел ответ на каждый из вышеперечисленных вопросов.
2. Узнал мнение опрашиваемых по поводу результатов опроса, на сколько они с ними согласны.
3. Опросил минимум 10 учителей.
4. Минимум 4 педагога должны работать не более 5 лет, это нужно было для сравнения разных групп.

Опрос был проведен в одном из корпусов школе №1514, который находится по адресу улица Новаторов, 28. В опросе приняло участие 15 учителей, 4 из которых работали менее 5 лет, все опрашиваемые были согласны с результатами опроса и каждый поделился своим мнением.

Проанализировав результат опроса, я пришел к следующим заключениям:

1. Вопросы актуальные и не требуют изменений.
2. В добавление новых вопросов нет никакого смысла, так как имеющиеся уже в полной мере дают все необходимые данные для подведения итога.
3. Все вопросы понятны, и не требуют дополнительных комментариев.
4. Опрашиваемые понимают, как нужно оценивать свое состояние по каждому из вопросов.

Из заключений сделал следующий вывод, опросник К.Маслач полностью подходить для моей программы.

## 

## Проведение опроса с целью выявления корректности работы программы

После создания работающей программы я решил провести ещё один опрос, но уже с другой целью, а именно ответить на следующие вопрос “Насколько удобно пользоваться данным телеграмм ботом?” Этот вопрос я разбил на следующие подвопросы:

1. На сколько интуитивно понятен функционал, “интерфейс” бота?
2. Комфортно ли читать вопросы?
3. Удобно ли пользоваться созданной клавиатурой?
4. На сколько комфортен вывод результатов?

Данный опрос было решено проводить уже не в школе, так как к моменту создания программы каникулы закончились, плюс, у учителей появилась дополнительная нагрузка в виде подготовки к ОГЭ и ЕГЭ.

Для того чтобы выбрать наиболее подходящее место для опроса, оно должно отвечать следующим требованиям:

1. Количество опрашиваемых должно быть более 20.
2. Они должны иметь разный уровень владения телефоном, ноутбуком и т.п.
3. Будет здорово, если среди опрашиваемых будет разный уровень владения русским языком.

Проанализировав требования, я пришел к выводу, что самое подходящее место для проведения опроса, доступных мне, это кадровое агентство Яндекс Еды по адресу Москва, Огородный проезд, 12. Спустившись с начальницей этого отделения, мы договорились на следующих условиях:

1. Я могу приходит с 9–11 часов, когда поток людей минимальный.
2. Опрос я можно проводить только в РФ зале.
3. Опрос можно проводить среди кандидатов (людей передних устраиваться курьером) только если они полностью подготовлены к дальнейшему оформлению (устройству на работу) и это их не задаривает.
4. Опрос можно проводить среди администраторов, но если они согласны и это не тормозит очередь.
5. В конце эксперимента я сообщу результаты опроса, без фамилий управляющей, опрашиваемые знают про этот пункт.

По прошествию двух дней и завершению опроса, мною были сформулированы следующие выводы:

1. Функционал, “интерфейс” бота понятен и дальнейшей настройки не требует.
2. Читать вопросы комфортно.
3. Расположение кнопок не очень удобное, цифры 5 и 6 не помещаются на экране.
4. Конечный результат перенасыщен информацией.

Далее я изменил дизайн клавиатуры, и она приобрела нынешний вид, раньше одна цифра занимала одну строку. Конечный вывод результатов тоже был претерпел изменения, но всё ещё требует доработки.

# 

# Заключение

Подводя итог проделанной работы хочется отметить следующие моменты которых удалось достичь:

1. По итогам работы удалось сформировать инструмент оценки уровня профессионального выгорания, не требующий специальных ИТ-навыков и удобный для использования широким кругом специалистов. По предварительной оценке, полученный инструмент можно развивать и дальше, обогащая его следующим:
   1. корректировка способа представления информации (на отдельных этапах работы с инструментом информация представлена достаточно сложно, местами интерфейс перегружен текстом,
   2. реализация в инструменте механизмов верификации вводимых данных (проверки на достоверность/корректность, вводимой пользователем информации),
   3. адаптация клавиатуры для использования на ПК;
2. В результате сбора информации (проведения интервью) и последующего ее анализа были намечены шаги для последующего развития инструмента оценки уровня профессионального выгорания, в том числе такими как:
   1. дополнение программы новыми тестами,
   2. адаптация для использования в инструменте реляционной БД для хранения и последующей агрегации полученных результатов и возможности использования результатов для более широкой выборки интервью. Кроме этого, наличие в архитектуре продукта такой БД позволило бы интегрировать продукт с другими средствами и инструментами, используемыми специалистами, профессионально занимающимися проблемой кадров на предприятии, анализом и корректировкой психологического климата на предприятии,
   3. расширения функциональности инструмента различными аналитическими модулями и компонентами,
   4. корректировка программного кода – приведение его к принципам объектно-ориентированного программирования (ООП), В результате такой корректировки представляется возможным достичь следующего эффекта: более структурированный и читаемый код для специалистов ИТ (важно для последующего развития и технической поддержки продукта), оптимизация быстродействия и размера приложения на хосте;
3. Проводя интервью, значительная часть опрашиваемых смогла яснее сформулировать для себя причины своего профессионального выгорания и совместно, на стадии проведения интервью, наметить шаги преодоления этого состояния.

# 

# Список литературы

Смысл жизни и стресс без дистресса. Теория Ганса Селье:

<https://psyfactor-org.turbopages.org/psyfactor.org/s/lib/selye-stress-distress-2.htm>

Три стадии стресса:

<https://dnkom.ru/o-kompanii/stati/diagnostika-v-detalyakh/stadii-stressa/>

Теория стресса г.Селье:

<https://studfile.net/preview/6055368/page:3/>

Стресс без дистресса:

<http://psyberia.ru/book/distress.pdf>

Эмоциональное выгорание и как с ним жить:

<https://moluch.ru/archive/450/>

Масштабное исследование выгорания:

<https://rb.ru/opinion/burnout-theory/>

5 стадий выгорания и как с ними справиться:

<https://rb.ru/story/burnout-stages/>

Исследование синдрома эмоционального выгорания у представителей различных профессиональных групп:

<https://research-journal.org/archive/7-121-2022-july/study-of-emotional-burnout-syndrome-in-representatives-of-various-professional-groups>

Влияние выгорания на организм человека:

<https://onkto.ru/blog/psychology/vliyanie-vygoraniya-na-organizm-cheloveka>

Опросник выгорания К.Маслач:

<https://solnyshkochg.ru/docs/knyazeva/masluch.pdf>

# 

# Приложение

Структура программы

<https://github.com/FrowningCat/Telegram-bot-Professional-burnout.git>

1. Папка answers - служит для хранения файлов, в которых прописано количество вопросов в каждом из вопросов.
2. Папка bot\_commands - служит для хранения файлов, в которых прописаны все команды распознаваемые программой.
3. Папка keyboard - служит для хранения различных видов клавиатур.
4. Папка tests - хранит написанные тесты.
5. Файла main.py - служит для за запуска программы.
6. Файла loader.py - служит для хранения токена моего telegram бота.
7. Файла test\_burnout.txt - служит для хранения ответов пользователей по тесту “Уровень профессионального выгорания”.
8. Файла test\_personality.txt - служит для хранения ответов пользователей по тесту “Тип личности”.

Папка answers содержит следующие файлы

1. burnout\_answer.py
2. personality\_answer.py
3. \_\_init\_\_.py

Папка bot\_commands содержит следующие файлы

1. about.py
2. help.py
3. start.py
4. start\_testing.py
5. type\_bot\_commands.py

Папка keybordсодержит следующие файлы

1. \_\_init\_\_.py
2. digital\_data\_type.py

Папка tests содержит следующие файлы

1. \_\_init\_\_.py
2. burnout.py
3. personality.py

Содержание файла burnout\_answer.py:

from aiogram.dispatcher.filters.state import StatesGroup, State

class CallbackOnStart(StatesGroup):

Q0 = State()

Q1 = State()

Q2 = State()

Q3 = State()

Q4 = State()

Q5 = State()

Q6 = State()

Q7 = State()

Q8 = State()

Q9 = State()

Q10 = State()

Q11 = State()

Q12 = State()

Q13 = State()

Q14 = State()

Q15 = State()

Q16 = State()

Q17 = State()

Q18 = State()

Q19 = State()

Q20 = State()

Q21 = State()

Q22 = State()

Q23 = State()

Содержание файла \_\_init\_\_.py в папке answers :

from .burnout\_answer import CallbackOnStart

Содержание файла personality\_answer.py:

пусто

Содержание файла about.py:

from aiogram import types

from aiogram.dispatcher.filters import Command

from loader import dp

@dp.message\_handler(Command("about"))

async def about(message: types.Message):

text = "Данного бота написал Савинков Владислав Олегович, студен третьего курса МГППУ, в качестве своей лабораторной" \

" работы.\n\nЗадача бота очень проста, он предоставляет возможность пройти несколко опросов. \n\nС" \

" обратной связью можете написать мне в \ntelegem @FrowningCat или на \nпочту vsavinkov60@gmail.com"

await message.answer(text=text)

Содержание файла help.py:

from aiogram import types

from aiogram.dispatcher.filters.builtin import CommandHelp

from loader import dp

@dp.message\_handler(CommandHelp())

async def bot\_help(message: types.Message):

text = ("Список команд: ",

"/start - Запустить бота",

"/help - Получить список команд",

"/about - Получить анатацию к проекту",

"/on\_start\_test - Выбрать тест")

await message.answer("\n".join(text))

Содержание файла start.py:

from aiogram import types

from aiogram.dispatcher.filters.builtin import CommandStart

from loader import dp

@dp.message\_handler(CommandStart())

async def bot\_start(message: types.Message):

await message.answer(f"Здравствуйте, {message.from\_user.full\_name}!\nЭто бот был создан в качестве курсовой "

"работы\nНапишите комманду /help чтобы получить подробную информацию о его возможнастях")

Содержание файла start\_testing.py:

from aiogram.dispatcher.filters import Command

from loader import dp

from aiogram import types

@dp.message\_handler(Command("on\_start\_test"))

async def on\_start\_test(message: types.Message):

await message.answer("Какрй тест хотите пройти? \nУровень профессиолнального самовыгорания "

"/on\_start\_burnout\_test \nТип личности /on\_start\_personality\_test")

Содержание файла type\_bot\_commands.py:

from aiogram import types

async def set\_default\_commands(dp):

await dp.bot.set\_my\_commands(

[

types.BotCommand("start", "Запустить бота"),

types.BotCommand("help", "Список всех команд"),

types.BotCommand("on\_start\_test", "Выбрать опррс для прохождения"),

]

)

Содержание файла \_\_init\_\_.py в папке keybord:

from .digital\_data\_type\_bytton import kb

Содержание файла digital\_data\_type.py:

from aiogram.types import ReplyKeyboardMarkup, KeyboardButton

b0 = KeyboardButton("0")

b1 = KeyboardButton("1")

b2 = KeyboardButton("2")

b3 = KeyboardButton("3")

b4 = KeyboardButton("4")

b5 = KeyboardButton("5")

b6 = KeyboardButton("6")

kb = ReplyKeyboardMarkup(resize\_keyboard=True)

kb.add(b0).row(b1, b2, b3).row(b4, b5, b6)

Содержание файла \_\_init\_\_.py в папке tests:

пусто

Содержание файла burnout.py:

import os.path

from aiogram.dispatcher import FSMContext

from aiogram.dispatcher.filters import Command

from aiogram.types import ReplyKeyboardRemove

from loader import dp

from aiogram import types

from answers import CallbackOnStart

from keyboard import kb

result = []

module = 0

points\_for\_emotional\_exhaustion = []

points\_for\_depersonalization = []

points\_for\_the\_reduction\_of\_professionalism = []

@dp.message\_handler(Command("on\_start\_burnout\_test"))

async def on\_start\_test(message: types.Message, state: FSMContext):

path = "./test\_burnout.txt"

check\_file = os.path.isfile(path)

if check\_file:

result.append(message.from\_user.id)

await message.answer("Данный опрос предназначен для диагностики профессионального выгорания и был разработан"

" Кристиной Маслач и Сьюзан Джексон в 1986 году.\nНапишите /start для наала теста")

await CallbackOnStart.Q0.set()

else:

await message.answer("Упс, один из файов не найден, обратитесь в поддержку")

await state.finish()

@dp.message\_handler(state=CallbackOnStart.Q0)

async def question(message: types.Message):

await message.answer("Кем вы работаете?")

result.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q1)

async def question(message: types.Message):

await message.answer(

text="Вопрос №1\nЯ чувствую себя эмоционально опустошенным\nВыберите по шкале от 0 до 6, где 0 это 'Не чусвую совсем себя опустошенным', а 6 'Постоянно чуствую себя опустошенным'")

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q2)

async def question(message: types.Message):

await message.answer(

text="Вопрос №2\nПосле работы я чувствую себя как 'выжатый лимон'",

reply\_markup=kb)

points\_for\_emotional\_exhaustion.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q3)

async def question(message: types.Message):

await message.answer(

text="Вопрос №3\nУтром я чувствую усталость и не желание идти на работу ",

reply\_markup=kb)

points\_for\_emotional\_exhaustion.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q4)

async def question(message: types.Message):

await message.answer(

text="Вопрос №4\nЯ хорошо понимаю, что чувствуют мои подчиненные и коллеги, и стараюсь учитывать это в интересах дела",

reply\_markup=kb)

points\_for\_emotional\_exhaustion.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q5)

async def question(message: types.Message):

await message.answer(

text="Вопрос №5\nЯ чувствую, что общаюсь с некоторыми подчиненными и коллегами как с предметами (без теплоты и расположения к ним)",

reply\_markup=kb)

points\_for\_the\_reduction\_of\_professionalism.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q6)

async def question(message: types.Message):

await message.answer(

text="Вопрос №6\nПосле работы на некоторое время хочется уединиться от всех и всего",

reply\_markup=kb)

points\_for\_depersonalization.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q7)

async def question(message: types.Message):

await message.answer(

text="Вопрос №7\nЯ умею находить правильное решение в конфликтных ситуациях, возникающих при общении с коллегами",

reply\_markup=kb)

module = (int(message.text) - 6) \* (-1)

points\_for\_emotional\_exhaustion.append(module)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q8)

async def question(message: types.Message):

await message.answer(

text="Вопрос №8\nЯ чувствую угнетенность и апатию",

reply\_markup=kb)

points\_for\_the\_reduction\_of\_professionalism.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q9)

async def question(message: types.Message):

await message.answer(

text="Вопрос №9\nЯ уверен, что моя работа нужна людям",

reply\_markup=kb)

points\_for\_emotional\_exhaustion.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q10)

async def question(message: types.Message):

await message.answer(

text="Вопрос №10\nВ последнее время я стал более «черствым» по отношению к тем, с кем работаю",

reply\_markup=kb)

points\_for\_the\_reduction\_of\_professionalism.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q11)

async def question(message: types.Message):

await message.answer(

text="Вопрос №11\nЯ замечаю, что моя работа ожесточает меня",

reply\_markup=kb)

points\_for\_depersonalization.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q12)

async def question(message: types.Message):

await message.answer(

text="Вопрос №12\nУ меня много планов на будущее, и я верю в их осуществление",

reply\_markup=kb)

points\_for\_emotional\_exhaustion.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q13)

async def question(message: types.Message):

await message.answer(

text="Вопрос №13\nМоя работа все больше меня разочаровывает",

reply\_markup=kb)

points\_for\_depersonalization.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q14)

async def question(message: types.Message):

await message.answer(

text="Вопрос №14\nМне кажется, что я слишком много работаю",

reply\_markup=kb)

points\_for\_emotional\_exhaustion.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q15)

async def question(message: types.Message):

await message.answer(

text="Вопрос №15\nБывает, что мне действительно безразлично то, что происходит c некоторыми моими подчиненными и коллегами",

reply\_markup=kb)

points\_for\_emotional\_exhaustion.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q16)

async def question(message: types.Message):

await message.answer(

text="Вопрос №16\nМне хочется уединиться и отдохнуть от всего и всех",

reply\_markup=kb)

points\_for\_depersonalization.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q17)

async def question(message: types.Message):

await message.answer(

text="Вопрос №17\nЯ легко могу создать атмосферу доброжелательности и сотрудничества в коллективе",

reply\_markup=kb)

points\_for\_emotional\_exhaustion.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q18)

async def question(message: types.Message):

await message.answer(

text="Вопрос №18\nВо время работы я чувствую приятное оживление",

reply\_markup=kb)

points\_for\_the\_reduction\_of\_professionalism.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q19)

async def question(message: types.Message):

await message.answer(

text="Вопрос №19\nБлагодаря своей работе я уже сделал в жизни много действительно ценного",

reply\_markup=kb)

points\_for\_the\_reduction\_of\_professionalism.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q20)

async def question(message: types.Message):

await message.answer(

text="Вопрос №20\nЯ чувствую равнодушие и потерю интереса ко многому, что радовало меня в моей работе",

reply\_markup=kb)

points\_for\_the\_reduction\_of\_professionalism.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q21)

async def question(message: types.Message):

await message.answer(

text="Вопрос №21\nНа работе я спокойно справляюсь с эмоциональными проблемами",

reply\_markup=kb)

points\_for\_emotional\_exhaustion.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q22)

async def question(message: types.Message):

await message.answer(

text="Вопрос №22\nВ последнее время мне кажется, что коллеги и подчиненные все чаще перекладывают на меня груз своих проблем и обязанностей",

reply\_markup=kb)

points\_for\_the\_reduction\_of\_professionalism.append(message.text)

await CallbackOnStart.next()

@dp.message\_handler(state=CallbackOnStart.Q23)

async def end(message: types.Message, state: FSMContext):

points\_for\_depersonalization.append(message.text)

await message.answer(text="Ваш результат:", reply\_markup=ReplyKeyboardRemove())

emotional\_exhaustion = [int(s) for s in points\_for\_emotional\_exhaustion]

emotional\_exhaustion = sum(emotional\_exhaustion)

depersonalization = [int(s) for s in points\_for\_depersonalization]

depersonalization = sum(depersonalization)

reduction\_of\_professionalism = [int(s) for s in points\_for\_the\_reduction\_of\_professionalism]

reduction\_of\_professionalism = sum(reduction\_of\_professionalism)

await message.answer("«Эмоциональное истощение» проявляется в переживаниях, утраты интереса и позитивных "

"чувств к окружающим, ощущении «пресыщенности» работой, неудовлетворенностью жизнью в целом. "

"В контексте синдрома перегорания «деперсонализация» предполагает формирование особых, "

"деструктивных взаимоотношений с окружающими людьми.")

if (emotional\_exhaustion < 16):

await message.answer("У вас эмоциональное истощение на низом уровне")

elif (emotional\_exhaustion > 15) and (emotional\_exhaustion < 25):

await message.answer("У вас эмоциональное истощение на среднем уровне")

else:

await message.answer("У вас эмоциональное истощение на высоком уровне")

await message.answer("«Деперсонализация» проявляется в эмоциональном отстранении и безразличии, формальном "

"выполнении профессиональных обязанностей без личностной включенности и сопереживания, а в "

"отдельных случаях – в негативизме и циничном отношении. На поведенческом уровне "

"«деперсонализация» проявляется в высокомерном поведении, использовании профессионального "

"сленга, юмора, ярлыков.")

if (depersonalization < 6):

await message.answer("Ваша деперсонализация на низком уровне")

elif (depersonalization > 5) and (depersonalization < 11):

await message.answer("Ваша деперсонализация на среднем уровне")

else:

await message.answer("Ваша деперсонализация на высоком уровне")

await message.answer("«Редукция профессиональных достижений» отражает степень удовлетворенности "

"работника собой как личностью и как профессионалом. Неудовлетворительное значение этого "

"показателя отражает тенденцию к негативной оценке своей компетентности и продуктивности и,"

" как следствие, - снижение профессиональной мотивации, нарастание негативизма в отношении "

"служебных обязанностей, тенденцию к снятию с себя ответственности, к изоляции от окружающих, "

"отстраненность и неучастие, избегание работы сначала психологически, а затем физически.")

if (reduction\_of\_professionalism > 36):

await message.answer("Редукция профессиональных достижений у вас на низком уровне")

elif (reduction\_of\_professionalism > 30) and (reduction\_of\_professionalism < 37):

await message.answer("Редукция профессиональных достижений у вас на среднем уровне")

else:

await message.answer("Редукция профессиональных достижений у вас на высоком уровне")

result.append(emotional\_exhaustion)

result.append(depersonalization)

result.append(reduction\_of\_professionalism)

with open("test\_burnout.txt", "w") as file:

file.write(str(result))

file.close()

await state.finish()

Содержание файла personality.py:

from aiogram.dispatcher import FSMContext

from aiogram.dispatcher.filters import Command

from loader import dp

from aiogram import types

@dp.message\_handler(Command("on\_start\_personality\_test"))

async def on\_start\_test(message: types.Message, state: FSMContext):

await message.answer("Упас, данный тест ещё не готов")

await state.finish()

Содержание файла main.py:

from aiogram import Bot, Dispatcher, types

from aiogram.contrib.fsm\_storage.memory import MemoryStorage

token = "6208826151:AAFO-4j83DT9-\_t\_hc4F2Dfx5rsicCrc3oM"

bot = Bot(token=token, parse\_mode=types.ParseMode.HTML)

storage = MemoryStorage()

dp = Dispatcher(bot, storage=storage)

Содержание файла loader.py:

from aiogram import executor

from loader import dp

from bot\_commands import help, start, start\_testing, about

from tests import burnout, personality

from bot\_commands.type\_bot\_commands import set\_default\_commands

async def on\_startup(dispatcher):

# Устанавливаем дефолтные команды

await set\_default\_commands(dispatcher)

if \_\_name\_\_ == "\_\_main\_\_":

executor.start\_polling(dp, on\_startup=on\_startup)