

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук  
Образовательная программа «Прикладная математика и информатика»

**Выпускная квалификационная работа на тему:**  
**Программа, оценивающая устойчивость к прокрастинации**

**Выполнил студент:**

группы БПМИ209, 4 курса  
Наумова Евгения Ильинична

\_\_\_\_\_

(подпись)

\_\_\_\_\_

(дата)

**Принял руководитель ВКР:**

Шершаков Сергей Андреевич  
Доцент  
Кандидат компьютерных наук  
Факультет компьютерных наук НИУ ВШЭ

\_\_\_\_\_

(подпись)

\_\_\_\_\_

(дата)

# Содержание

<b>Аннотация</b>	<b>3</b>
<b>Ключевые слова</b>	<b>4</b>
<b>1 Введение</b>	<b>5</b>
1.1 Актуальность . . . . .	5
1.2 Новизна . . . . .	5
<b>2 Обзор литературы</b>	<b>7</b>
<b>3 Описание программы</b>	<b>10</b>
3.1 Требования . . . . .	10
3.1.1 Функциональные требования . . . . .	10
3.1.2 Нефункциональные требования . . . . .	10
3.2 Описание использованных методов и метрик . . . . .	11
3.2.1 Алгоритм работы программы . . . . .	11
3.2.2 Существующие методы тестирования и примеры задач . . . . .	11
3.3 Отобранные тестирующие задачи и метрики оценивания . . . . .	13
3.4 Разработка макетов и реализация интерфейса . . . . .	14
3.5 Реализация задач . . . . .	16
3.6 Пример использования . . . . .	17
3.7 Проведение тестирования, анализ результатов, вывод формулы оценки устойчи- чивости к прокрастинации . . . . .	24
3.7.1 Анализ полученных результатов тестирования . . . . .	24
3.7.2 Формула оценки с помощью штрафов . . . . .	27
3.7.3 Формула оценки выведенная аналитически . . . . .	28
<b>4 Заключение</b>	<b>31</b>
<b>Список литературы</b>	<b>32</b>
<b>Приложения</b>	<b>33</b>

## Аннотация

Одним из важных направлений в области исследования мозга является раздел, связанный с когнитивными функциями. НИИ развития мозга и высших достижений РУДН занимается большим количеством вопросов, связанных с этим разделом науки, и предлагает широкий спектр задач и проблем для исследований в этой области.

В данной работе представлена разработка, относящаяся к одной из важных тем в области исследования человеческого мозга – устойчивости человека к прокрастинации. Прокрастинация, феномен откладывания задач на потом, имеет большое значение для понимания когнитивных процессов и их влияния на поведение человека. В данной работе проводятся эксперименты различной сложности, направленные на определение точности выполнения задач и временных рамок, чтобы оценить уровень устойчивости к прокрастинации. Результаты применения программы могут послужить основой для дальнейших исследований в этой области.

## Ключевые слова

- Прокрастинация – в психологии склонность к постоянному откладыванию даже важных и срочных дел, приводящая к жизненным проблемам и болезненным психологическим эффектам. Впервые данный термин был введен Р. Ringenbach в 1977 году в книге «Procrastination in Human Life» [3].
- Поведенческие модели – то модели, которые описывают поведение людей в различных ситуациях. Они используются в психологии, социологии, экономике и других областях для объяснения причин и механизмов человеческого поведения.
- Тестирование аспектов интеллекта – это стандартизированный метод оценки интеллектуальных способностей человека.
- Моделирование данных – это анализ и определение всех типов данных, а также взаимосвязей между этими фрагментами данных.

# 1 Введение

Не теряет своей актуальности оценка способности когнитивных функций человеческого мозга. В современной науке большой интерес у ученых вызывает вопрос о том как влияют различные факторы на эффективность мыслительных процессов. Прокрастинация, как и другие негативные факторы, может значительно влиять на продуктивность и качество жизни людей. Поэтому все больше внимания уделяется исследованиям и разработке новых методов в борьбе с этим явлением.

Наши коллеги из НИИ развития мозга и высших достижений РУДН уже занимаются большим количеством вопросов связанных с диагностикой когнитивных функций, и программа, которая будет оценивать устойчивость к прокрастинации может во многом помочь с дальнейшими исследованиями этой тематики.

## 1.1 Актуальность

В современном обществе все чаще возникает проблема, связанная с откладыванием задач на потом. Ученые дали четкое определение такому поведению – прокрастинация.

Разработка программы, оценивающей устойчивость к прокрастинации, может во многом помочь в решении проблем, которые возникают вследствие нее. В первую очередь, это довольно сильно повлияет на повышение производительности в работе или учебе. Появится индивидуальный подход к каждому человеку в вопросе эффективного распределения и четкого контроля своего времени. К примеру, на основе реализованной нами программы появится возможность учитывать индивидуальные особенности каждого человека.

Также результаты реализованной программы отразятся на вопросе медицинского характера. Прокрастинация для организма в первую очередь – стресс, а разработка программы, которая сможет оценить устойчивость человеческого организма к данному явлению, даст возможность более глубоко понять сущность проблемы и разработать новые подходы в борьбе с ней.

## 1.2 Новизна

В настоящее время существует большое количество различных теоретических подходов для оценки устойчивости человека к прокрастинации. В нашей же работе основная идея заключается в сборе уже имеющихся исследований и объединении их в единую программу, что даст нам возможность учесть разнообразие подходов, тем самым позволив программе

быть довольно гибкой к различным типам пользователей.

Более того, реализация всех существующих подходов в целостной программе даст новые возможности в сфере изучения различных когнитивных функций не только в тематике прокрастинации.

## 2 Обзор литературы

Одним из первых этапов в работе над данным проектом является анализ уже имеющихся на данный момент подходов, методик и технологий в данной сфере.

- «Development and Validation of the General Procrastination Scale (GPS)» (Steel, 2010) [5]

В данной статье описывается процесс разработки шкалы прокрастинации GPS, которая предназначена для общей (основанной не только на учебе или работе) оценки предрасположенности человека к прокрастинации. Также уделяется время для тестирования и анализа полученных результатов. Основная идея данного метода GPS выглядит довольно стандартно – это опросник, состоящий из ряда утверждений, с помощью которого участники самостоятельно оценивают свою склонность к прокрастинации. Вот несколько примеров: «Часто я откладываю дела на последний момент», «Мне трудно начать работу, если у меня нет жесткого срока» или «Мне нравится работать под давлением, поскольку это мотивирует меня».

Сама шкала строится на основе составленных автором утверждений и их тестировании на обширном количестве выборок для оценки характеристик. В статье также анализируется точность разработанной шкалы на основе данных, собранных в ходе исследований. Основываясь на методе и результатах полученных в данной статье, мы приходим к пониманию того, какие задачи мы можем в дальнейшем использовать в нашей программе для оценки устойчивости к прокрастинации.

- «Assessment of Procrastination: A Critical Review» (Steel, 2007) [2]

Данная статья представляет собой обзор различных методов оценки прокрастинации. Автор, исследует различные подходы к измерению прокрастинации и выделяет в каждом из них преимущества и недостатки. Автором были выделены два типа: классический и современный подходы. Первый вариант включает в себя дневники активности, вопросники и тесты личностных черт. Второй же в отличие от первого предполагает разработку программы, которая мониторит время активности.

Автор обращает внимание, что разработка программы, которая сможет точно измерять прокрастинацию, является важным фактором, поскольку прокрастинация является значимой для понимания аспектов человеческой деятельности.

В результате изучения данной статьи мы еще раз убеждаемся в необходимости реализации программы которая будет оценивать устойчивость к прокрастинации. Основываясь на том, что в ходе анализа всех существующих методов автор смог обозначить

какие «слабости» есть у людей, страдающих от прокрастинацией, мы можем выделить несколько основных метрик, с помощью которых будем реализовывать программу. Главными аспектами в вопросе прокрастинации стали ограниченные сроки и качество работы. Если от человека не требовалось сделать задачу в короткие сроки или же сделать ее качественно, то выполнение такого задания вызывало у него сильное отвращение и нежелание им заниматься. В обратном случае, испытуемые довольно быстро и качественно справлялись с заданиями.

- «The Procrastination Assessment Scale-Students (PASS)» (Solomon Rothblum, 1984) [4]

Авторы статьи, представляют инструмент, который предназначен для оценки степени прокрастинации у студентов в процессе обучения.

Авторы описывают принцип работы шкалы PASS, который заключается в том что она оценивает частоту возникновения прокрастинации и ее степень влияния на повседневную жизнь человека. Используемые в данной исследовании методы также очень схожи с первой изученной мной статьей. Авторы также проводят опросники, включающие в себя ряд стандартных вопросов о прокрастинации, но уже среди студентов. Основываясь на результатах данной статьи, могут быть выявлены основные группы лиц, с помощью которых можно проводить исследования и тестирование нашей программы.

- «Procrastination: A Positive or Negative Trait?» (Kim Seo Seo Eun Hee, 2015) [1]

В данной статье авторы рассматривают фактор влияния прокрастинации на успеваемость в учебе с двух сторон, негативной и позитивной. К позитивной они относят так называемую “активную прокрастинацию”, это явление когда люди откладывают дела на потом, чтобы создавался эффект давления ограниченных сроков для более эффективной, на их взгляд работы. Но, все же, основные исследования демонстрируют, что прокрастинация имеет скорее негативный эффект на успеваемость в учебе. Тесты которые проводили авторы включали в себя элементы самоконтроля - дневник активности, также проводились эксперименты, предполагающие участие стороннего наблюдателя. Результаты полученные в результате экспериментов с участием наблюдателя, были более устойчивые, это объясняется тем, что участники, как правило, более предвзяты к себе. Основываясь на результатах, полученных в данной статье, можно определить тип задач, с помощью которых можно более точно оценить прокрастинацию.

В результате проведенного нами анализа статей, можно заметить, что в основном все предложенные методы и эксперименты опирались на самоотчет испытуемого (дневники ак-



тивности, вопросы и т.д.), что не даёт точного результата, поскольку участник, как правило, предвзят к себе. Поэтому мы предлагаем проводить эксперименты, которые будут базироваться на ранее изученных методах тестирования, но не требующих самоотчета, а связанных с конкретными тестирующими задачами, которые имеют четкий ответ. Это даст возможность оценить полученный результат с высокой точностью. Также, помимо метрики, предложенной в статье «Assessment of Procrastination: A Critical Review» (Steel, 2007) - ограниченное время, мы будем пользоваться и другими, которые подробно описаны в разделе [Отобранные тестирующие задачи и метрики оценивания](#).

## 3 Описание программы

В настоящее время вопросом прокрастинации занимается около 1000 учёных по всему миру. Это объясняется тем, что данное явление затрагивает многие сферы жизни каждого человека. Для борьбы с прокрастинацией, в лабораториях проводят эксперименты связанные с оценкой устойчивости человеческого организма к данному явлению. Рассматривается большое разнообразие задач, но все они делятся на два типа, это ведение дневника отчетности, где человек сам пытается отследить возникновение прокрастинации или же тестирование с ограничением по времени, срочности и т.д. В нашем случае мы будем рассматривать второй тип задач. Поэтому нашей целью является реализация программы, которая будет оценивать устойчивость к прокрастинации с помощью стандартных задач, направленных на внимание, концентрацию и память.

### 3.1 Требования

#### 3.1.1 Функциональные требования

В качестве основного языка программирования выбор был сделан в пользу Python, код программы будет размещён в GitHub-репозитории. Перечислим еще несколько не менее важных требований:

- 1 Проверка введенных данных на корректность.
- 2 Отслеживание времени, потраченного пользователем на задачу.
- 3 Отчётность. Программа должна предоставлять участнику отчёт о его предрасположенности к прокрастинации.

#### 3.1.2 Нефункциональные требования

Помимо функциональных требований, необходимо чтобы программы обладала удобным интерфейсом, высокой точностью и выбор метрик имел строгое обоснование.

## **3.2 Описание использованных методов и метрик**

### **3.2.1 Алгоритм работы программы**

В первую очередь при запуске программы проводится сбор и предобработка данных для дальнейшего тестирования, в нашем случае это основные данные о пользователе (ФИО, возраст, пол), предобработка включает в себя проверку данных на корректность и дальнейший ввод в поле для тестирования. Тестирование включает в себя два основных этапа – тренировочную часть и тестирующую. Первая часть позволяет пользователям ознакомиться с функционалом программы, научиться использовать её и освоить работу с интерфейсом для дальнейшего тестирования. В конечном итоге работы программы каждый участник будет получать результаты и краткий отчёт, обобщающий их. Помимо этого, мы планируем сохранять полученные результаты каждого из участников в отдельном файле. Это необходимо для наших коллег из лаборатории и для дальнейших исследований в области когнитивных функций.

### **3.2.2 Существующие методы тестирования и примеры задач**

Следующим этапом является отбор и придумывание задач для тестирования. Приступая непосредственно к работе мы осознаем, что прокрастинация – явление, с которым сталкивается каждый из нас ежедневно. Особенно это проявляется в тех моментах, когда требуется приложить дополнительные усилия или сосредоточиться на задаче. По этой причине в лабораториях проводится большое количество экспериментов, которые направлены на оценку того насколько человек устойчив к данному явлению. К примеру, одним из экспериментов могут быть задачи срочности. Смысл таких задач заключается в ограничении по времени или важности их выполнения, участникам предлагается выполнить серию задач, которые отличаются по степени срочности. Исследователь фиксирует время, затраченное на выполнение каждой задачи и уровень стресса при выполнении, далее, основываясь на полученных результатах, он проводит анализ, склонны ли участники откладывать выполнение важных задач в пользу менее значимых, что может свидетельствовать о прокрастинации. В данном исследовании будут использоваться эксперименты подобные данному для оценки точности.

Задачи будут подобраны таким образом, чтобы их решение требовало различных усилий, чтобы выявить корреляцию между ними в ходе экспериментов для дальнейшей оценки устойчивости к прокрастинации.

Эксперимент будет делиться на несколько этапов, для начала мы будем проводить

довольно простые эксперименты без ограничений, но при этом они будут отличаться сложностью выполнения. Это необходимо для того чтобы провести анализ того, как коррелирует объем заданных задач с фактором возникновения прокрастинации и как часто у участника возникнет желание прервать выполнение.

*Первый вариант. Простые задачи.* В данном случае испытуемому будут предложены простые арифметические задачи, прочтение небольшого текста, простые вопросы с ответом да/нет.

*Второй вариант. Усложненные задачи.* Это работа с большим объемом текста, сложные математические задачи, вопросы.

Далее мы будем усложнять данные задачи с помощью вторичного фактора. Для более точного результата мы будем рассматривать корреляцию различных факторов по сложности задач со временем, важностью и возможностью прерваться:

### 1 *Задачи на время.*

Испытуемому будет предложено выполнить задачу в определенном промежутке времени. Будет оцениваться количество и качество выполненной работы, а также приниматься личный отчет о том какое количество раз человеку захочется оторваться от выполнения задания.

### 2 *Задачи срочности и важности.*

Участнику предлагаются задачи с разными уровнями важности. Будет также оцениваться расстановка приоритетов при выполнении задач. К примеру:

Испытуемый получает список простых и сложных арифметических задач на день с указанием их важности. Программа предлагает свое «идеальное» решение о том, в каком порядке выполнять задачи, исходя из их приоритетов. Участнику необходимо либо предложить свою расстановку, либо согласиться с предложенной.

### 3 *Задачи с перерывами.*

В данном случае испытуемому будет представлена задача либо на логику, либо на память. В процессе ее выполнения участнику будет представлен список слов или задач, на некоторое ограниченное время. После окончания указанного времени, необходимо будет сделать перерыв, затем, нужно будет выполнить какую-то задачу связанную с информацией, которая давалась ему еще в самом начале задания. Данный тип задач позволяет оценить устойчивость памяти и её продолжительность при отвлекающих факторах.

Все предложенные варианты тестирования направлены на оценку того, как часто возникает прокрастинация в различных условиях.

### 3.3 Отобранные тестирующие задачи и метрики оценивания

Возвращаясь к нашей программе и отбору задач для тестирования, необходимо учитывать, что программа должна быть доступна для прохождения любому человеку, начиная с 16 лет, поэтому сложность задач подобрана с учётом данного фактора.

Для ознакомления участника со средой было отобрано 2 простых задачи на арифметику и исправление ошибок в тексте.

Для тестирующей части было отобрано 6 задач, разбитых на три блока, как в пункте 3.2.2. Перечислим их:

- 1 Простая задача на арифметику: В окне программы предлагается решить 3 арифметических задачи, сгенерированных случайным образом, выдается 1 минута на решение.
- 2 Система линейных уравнений: Участнику предлагается за 5 минут решить СЛУ с двумя неизвестными, все коэффициенты также генерируются случайным образом (при этом, гарантируется, что решение будет существовать всегда).
- 3 Система линейных уравнений и исправление ошибок в тексте: На данную задачу выдается 7 минут, за это время участнику необходимо также найти решение СЛУ с двумя неизвестными и, вдобавок, необходимо найти все ошибки в представленном тексте и исправить их в предложенном окне.
- 4 Распределение задач по приоритетности: Далее участнику предлагается список задач на день состоящий из 9 пунктов, ему необходимо интуитивно понять расстановку “для себя” и вписать в соответствующий блок ответ в виде последовательности чисел.
- 5 Задача на память с прерыванием: Участнику выдается список, состоящий из 7, случайно сгенерированных, английских слов и 1 минута, чтобы запомнить их, далее делается 3-х минутный перерыв, после которого участнику будет необходимо найти среди 17 представленных слов те 7, которые были представлены в первом окне и нажать на соответствующие им кнопки (порядок не имеет значения).
- 6 Усложненная задача на память: Участнику выдается список из 7 русских слов и 1 минута на запоминание, далее также делается перерыв 3 минуты и, затем, испытуемому

будет необходимо не только найти исходные 7 слов среди представленных, но и расположить их в алфавитном порядке.

Далее необходимо определиться с метриками, с помощью которых мы сможем давать оценку выполнения задач участниками. Основными являются следующие параметры: время, затраченное на выполнение задач, точность выполнения задач, количество задач, решенных в определенное время, вопрос о желании отвлечься (он будет возникать каждый раз после завершения задания). Такой выбор объясняется тем, что любое отклонение по какому-то из этих показателей, означает, что человек имеет склонность к прокрастинации.

### 3.4 Разработка макетов и реализация интерфейса

После подбора задач и отбора метрик, было решено реализовать приложение с графическим интерфейсом. Для реализации интерфейса программы использовалась библиотека PyQt [6] графического фреймворка Qt и приложение Qt Designer для создания макетов, такой выбор объясняется тем, что:

- Qt Designer позволяет визуально создать интерфейс, перетаскивая элементы управления и, тем самым, делает реализацию графического интерфейса интуитивно понятной.
- Привязка к Python. Использование PyQt позволяет легко интегрировать созданные в Qt Designer GUI в среду разработки Python. Если учитывать что мы изначально остановились на реализации данной программы с помощью языка программирования Python, то возможности данной библиотеки облегчают реализацию.
- Большой набор виджетов в Qt Designer. PyQt представляет широкий набор инструментов для реализации интерфейса, что делает его удобным в контексте реализации нашей программы. К примеру, есть возможность реализовать таймер, который будет отображаться на экране у пользователя.

Из возможностей реализованного нами интерфейса:

- Пользователь по кнопке может переходить на следующую страницу.
- В задачах с ограниченным временем отображается оставшееся время.
- Есть возможность редактировать текст или вписать ответ в соответствующих задачах.

Покажем примеры разработанных макетов для задач:

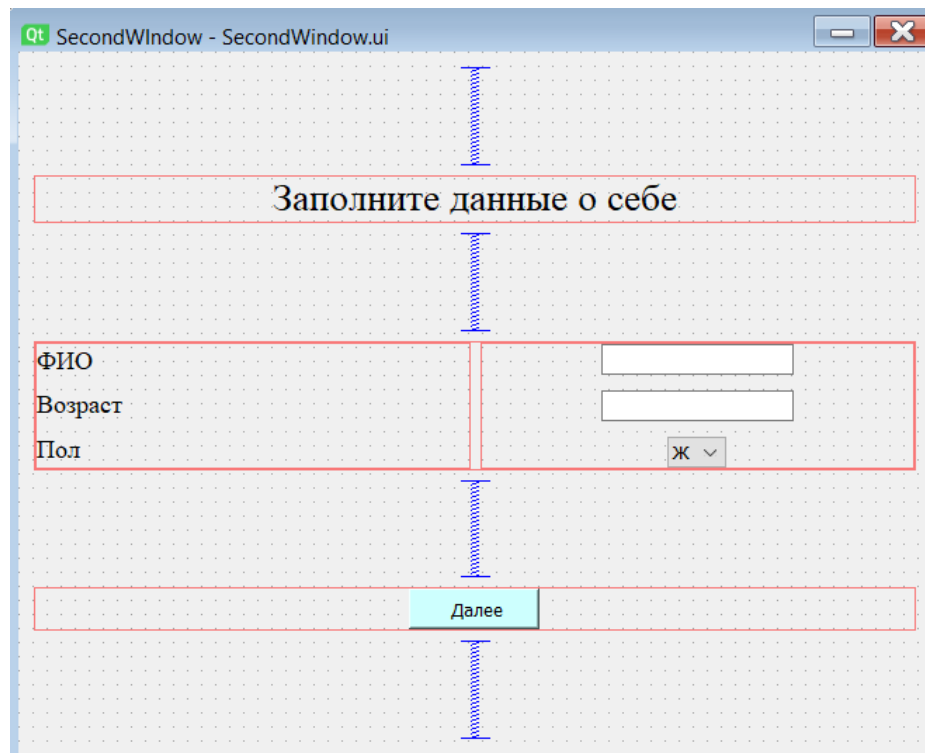


Рис. 3.1: Макет для окна регистрации, разработанного в Qt Designer.

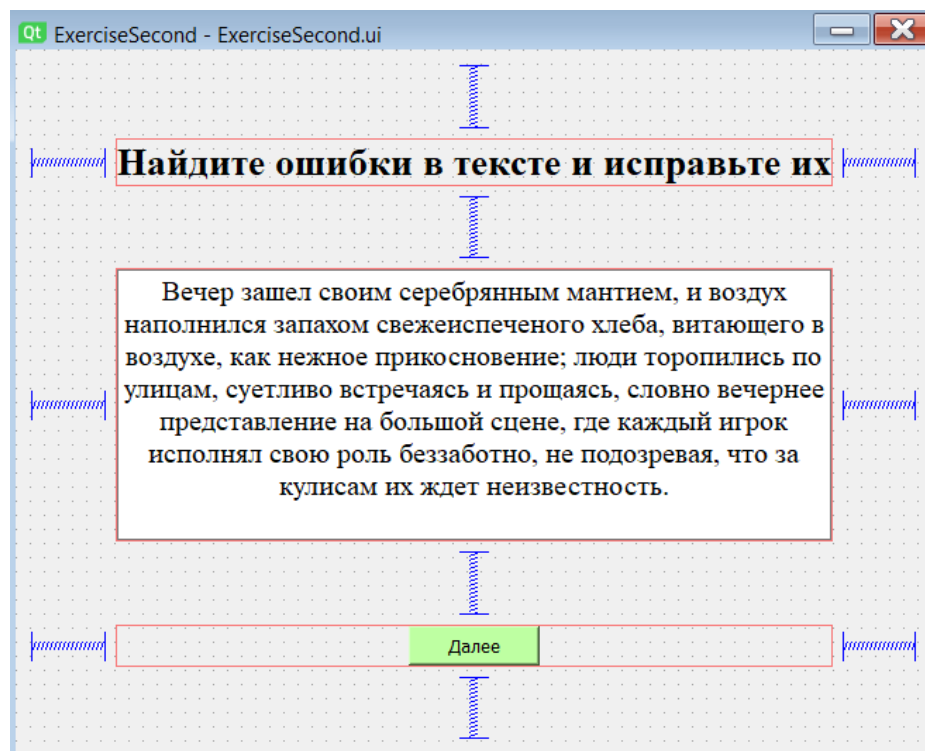


Рис. 3.2: Макет 2 задания тренировочной сессии, разработанного в Qt Designer.

После разработки макетов, они все были преобразованы в \*.py файлы и подключены в основную программу, также была прописана реализация таймера, переключение между окнами и т.д.

### 3.5 Реализация задач

Следующим этапом стала разработка и реализация выбранных нами задач. Мы не будем останавливаться на реализации подробно (реализацию можно посмотреть на github страничке, ссылка которой прикреплена в [Приложения](#)), опишем кратко алгоритм для каждой.

- Задача №1.

Необходимо было прописать реализацию случайной генерации чисел для вычисления арифметических задач. В соответствующей функции с помощью метода `np.random.randint` генерируются случайные числа и передаются на экран, далее, считывается ответ пользователя и сравнивается с фактическим результатом сложения в программе.

- Задача №2.

Генерация системы линейных уравнений с двумя неизвестными. Необходимо было также прописать реализацию генерации случайных чисел, что сделано ровно таким же способом как и в первой задаче, только здесь необходимо гарантировать, что решение будет существовать всегда, поэтому проверялся ранг составленной матрицы (что он не больше 2-х). Далее, после того, как коэффициенты были переданы на экран, они разделялись на две матрицы, матрицу коэффициентов  $A$  и матрицу решений  $b$  соответственно. Далее необходимо было найти решение уравнения вида  $Ax = b$ , что было сделано с помощью метода `np.linalg.solve`. Затем ответы пользователя сравнивались с фактическими.

- Задача №3.

Помимо СЛУ, в данной части также было необходимо исправить текст с ошибками, он не генерировался случайным образом, поэтому мы хранили в переменной соответствующее предложение без ошибок и после ввода пользователем исправленного текста, происходило сравнение с помощью метрики Левенштейна (подсчитывает разницу между двумя строками), которая реализована в отдельной функции `levenshtein_distance`.

- Задача №4.



Задача срочности и важности. Задача направлена на четкое понимание и умение распределять приоритеты. Основываясь на изученной нами литературе, была придумана правильная расстановка каждой из предложенных задач по порядку, поэтому мы просто хранили “идеальную” последовательность и сравнивали ее с той, которую пользователь вводит в соответствующую строку для ответа. Нам важно, чтобы каждая цифра находилась именно на своём месте, поэтому при сравнении двух массивов, если они не были равны по длине или их элементы не совпадали, то увеличивался счётчик ошибок.

- Задача №5.

Задача с генерацией случайных английских слов. Для реализации данной части кода была импортирована библиотека **Faker**, с помощью которой и происходила генерации 7 случайных слов, которые мы сохраняли в список. После определенного перерыва (мы не будем уделять время описанию реализации данной части, поскольку она очевидна), генерировалось ещё 10 случайных слов и, для каждой кнопки, рандомно присваивалось название, соответствующее слову, при нажатии которой, слово добавлялось в новый список. Далее начальный список слов сравнивался с конечным, поскольку последовательность в данной задаче неважна, то сравнение происходило с помощью пересечения `set()` для каждого списка, что давало нам количество совпадающих слов, а значит, чтобы определить количество ошибок, необходимо было из размерности первоначального вычесть пересечение.

- Задача №6.

Отбор соответствующих слов и расстановка по алфавиту. В данной задаче идея схожа с задачей №4, поскольку слова генерировались не случайным образом.

Для каждой задачи также была прописана валидация, когда пользователь мог вводить в строку предназначенную для числового ответа - буквы, когда человек мог ничего не ввести в поле для ввода или же ввести неполный ответ.

### 3.6 Пример использования

Для запуска программы необходимо открыть файл формата .exe. Первым делом перед участником возникает ознакомительно окно с информацией о проекте (рис. 3.3).

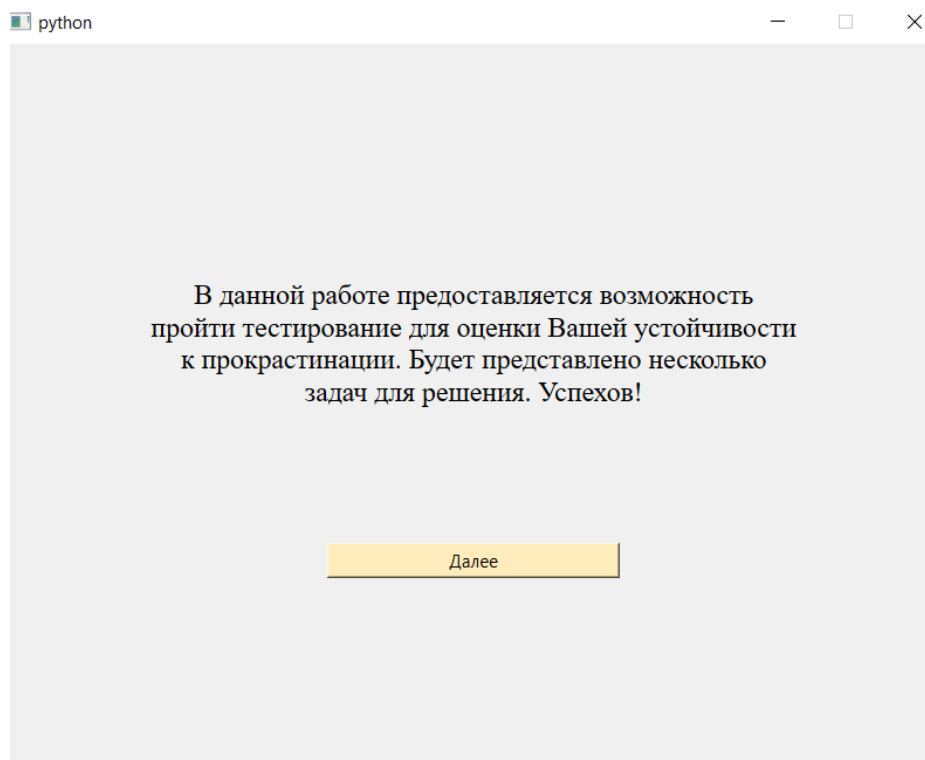


Рис. 3.3: Первое окно при запуске программы

Далее возникает страничка с регистрацией, где участнику необходимо ввести основную информацию о себе (рис. 3.4), в этом же окне, если участник вводит информацию неверно или забывает заполнить поле, возникает соответствующее окно, с информацией о том, в каком поле была совершена ошибка и очищает его (рис. 3.5).

Рис. 3.4: Страничка регистрации с невведенным полем “Возраст”.

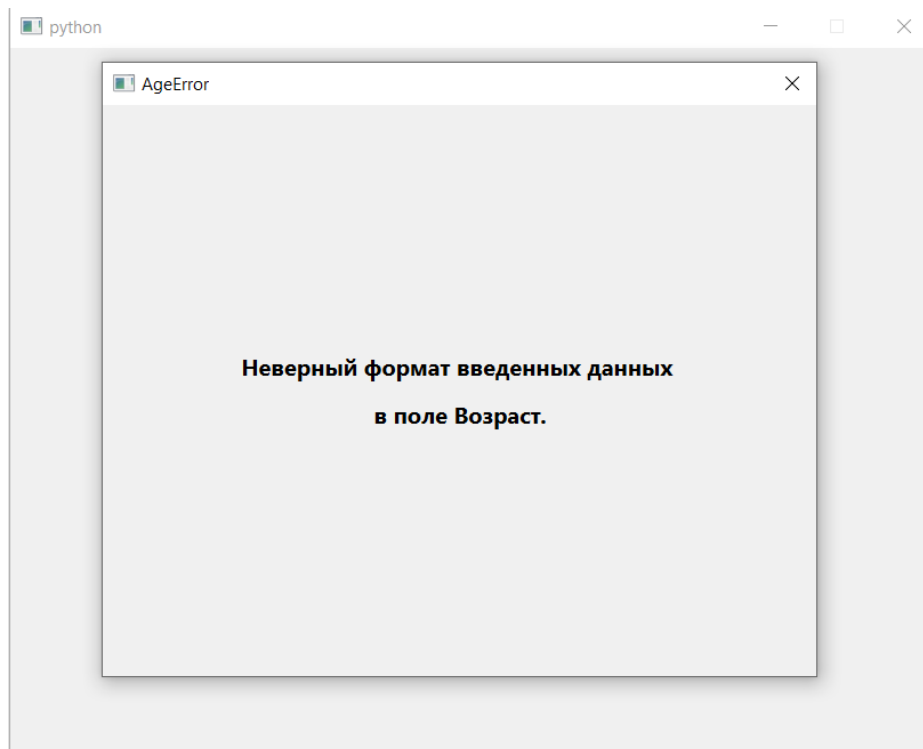


Рис. 3.5: Пример возможной ошибки.

В случае успешной регистрации пользователя переносит на следующее окно с информацией о начале тренировочной сессии. Она состоит из двух задач: 1) 3 Арифметических выражения (рис. 3.6). 2) Исправление текста с ошибками (рис. 3.2).

A screenshot of a Python application window. The main window has a title bar with 'python' and standard window controls. The content area is titled 'Арифметические задачи' (Arithmetic tasks). Below the title, there are three arithmetic problems, each followed by an empty input field for the answer:  
 $7 + 2 =$    
 $18 : 6 =$    
 $6 * 9 =$    
At the bottom center, there is a purple button with the text 'Далее' (Next).

Рис. 3.6: Первое задание тренировочной части.

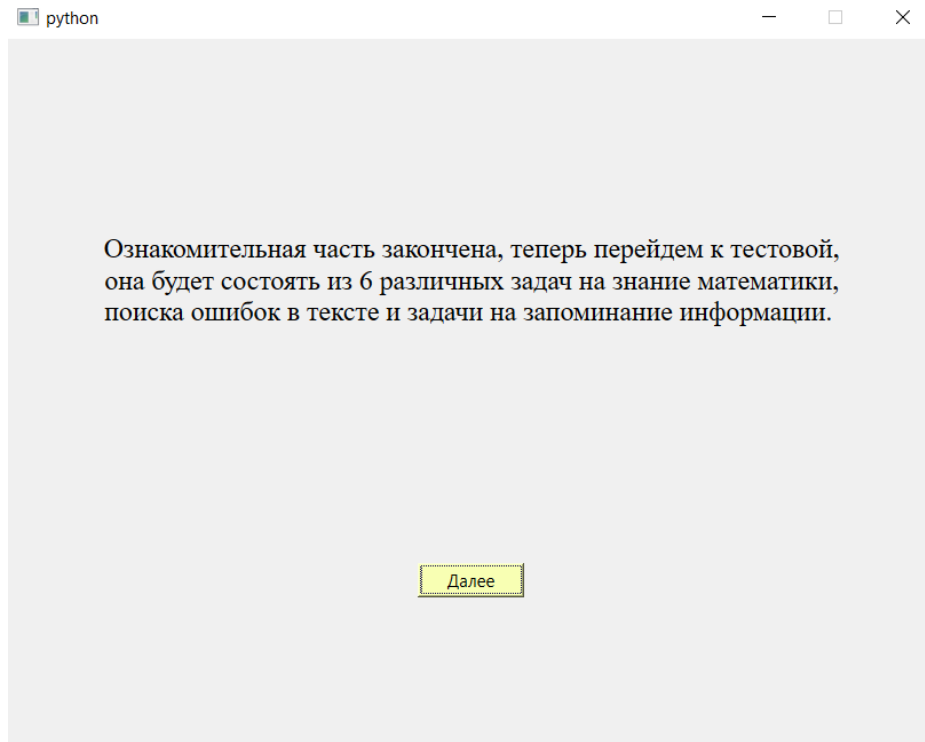


Рис. 3.7: Переход к тестовой части.

После прохождения тренировочной сессии участнику будет доступна тестовая часть (рис. 3.7), которая будет содержать в себе 6 задач, поэтому приведем несколько основных примеров, остальные добавлены в раздел [Приложения](#).

Первый блок тестовой сессии состоит из 3 заданий: Решение 3 арифметических выражений (сгенерированных случайным образом), решение системы линейных уравнений с двумя неизвестными (рис. 3.8), сгенерированных случайно, решение СЛУ и нахождение ошибок в тексте. Каждая из этих задач имеет свое временное ограничение (1, 5, и 7 минут соответственно), которое будет отображаться в окне. Если участник решит задачу раньше времени, он всегда может нажать на кнопку далее и перейти к следующей задаче.

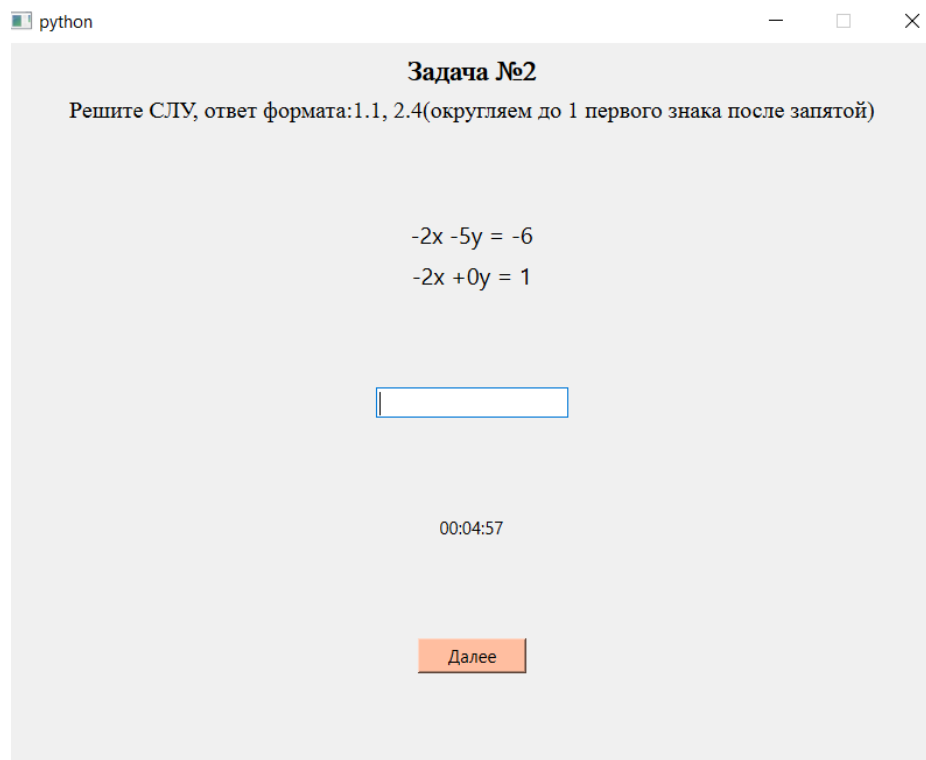


Рис. 3.8: Пример задачи из первого блока.

После каждого задания перед участником будет всплывать окно с вопросом (рис. 3.9) о том, захотелось ли ему сделать перерыв в выполнении. Данный фактор мы будем учитывать как вторичный при расчете итогового балла.

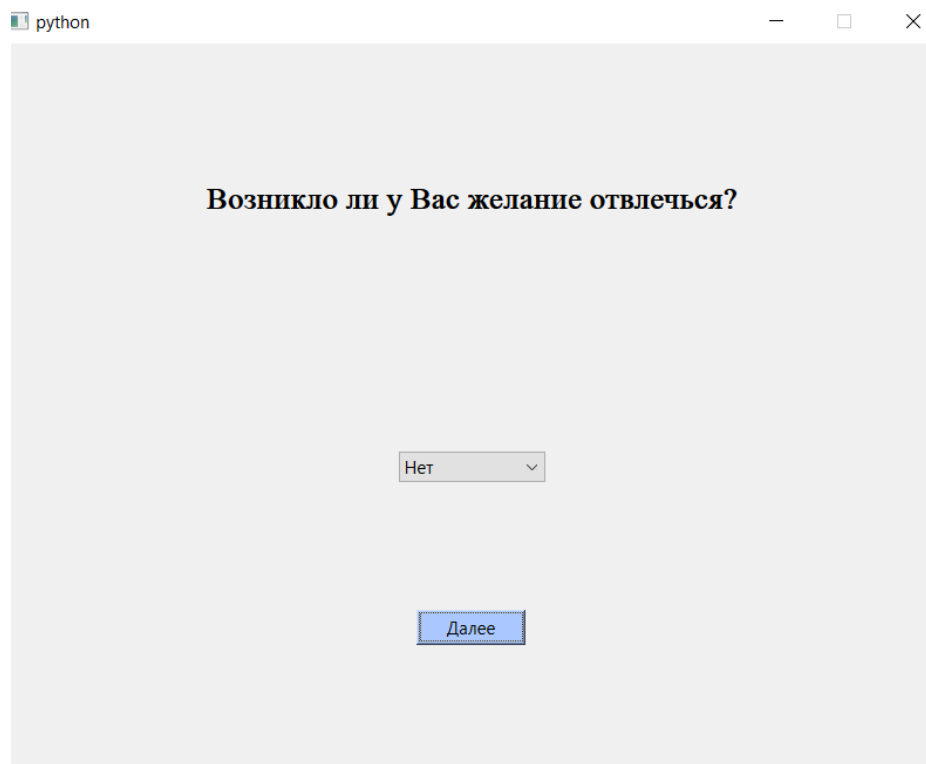
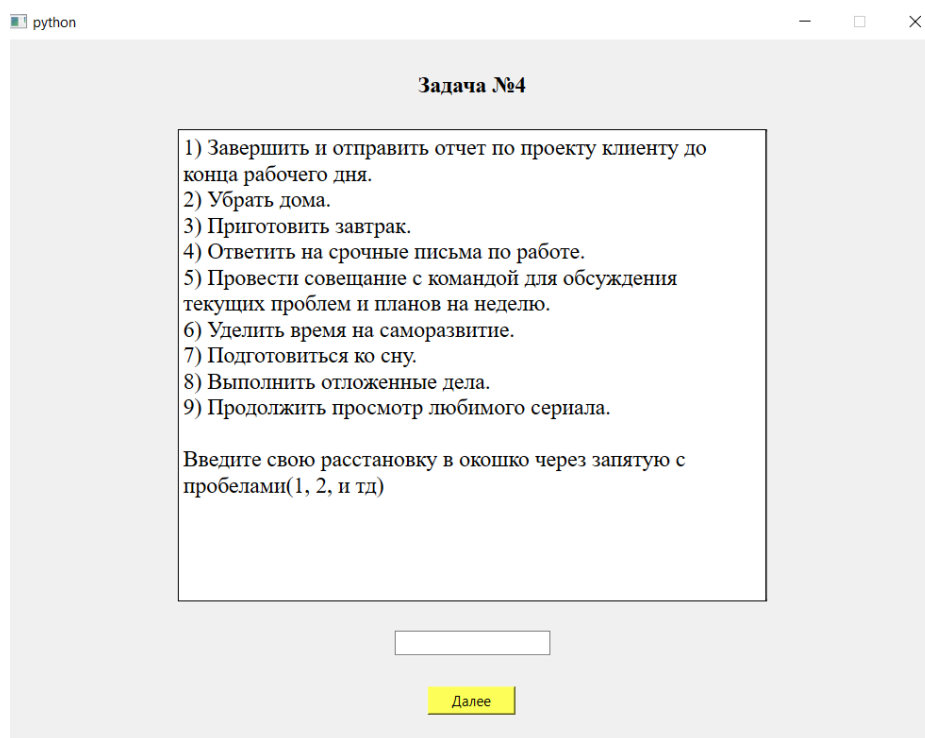


Рис. 3.9: Окно с вопросом.

После выполнения первого блока, пользователь увидит информацию о задаче из второго блока, где после нажатия кнопки “Далее” откроется окно (рис. 3.10) в котором будет сама задача и блок для ответа.



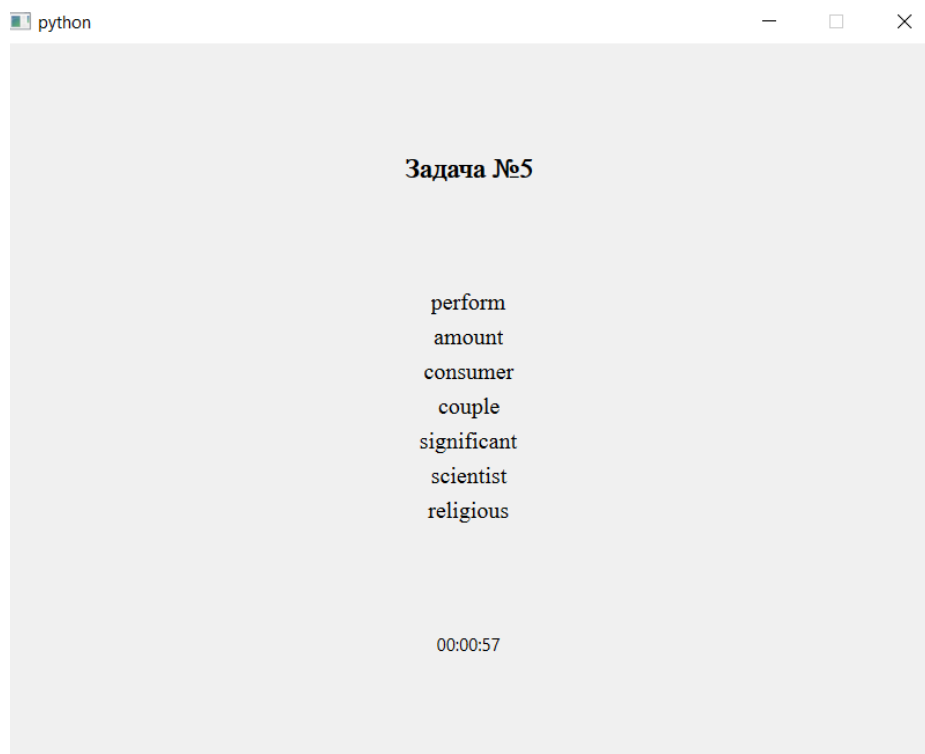
**Задача №4**

- 1) Завершить и отправить отчет по проекту клиенту до конца рабочего дня.
- 2) Убрать дома.
- 3) Приготовить завтрак.
- 4) Ответить на срочные письма по работе.
- 5) Провести совещание с командой для обсуждения текущих проблем и планов на неделю.
- 6) Уделить время на саморазвитие.
- 7) Подготовиться ко сну.
- 8) Выполнить отложенные дела.
- 9) Продолжить просмотр любимого сериала.

Введите свою расстановку в окошко через запятую с пробелами(1, 2, и тд)

Далее

Рис. 3.10: Задача срочности и важности.



**Задача №5**

perform  
amount  
consumer  
couple  
significant  
scientist  
religious

00:00:57

Рис. 3.11: Первая задача из раздела задач с перерывами.

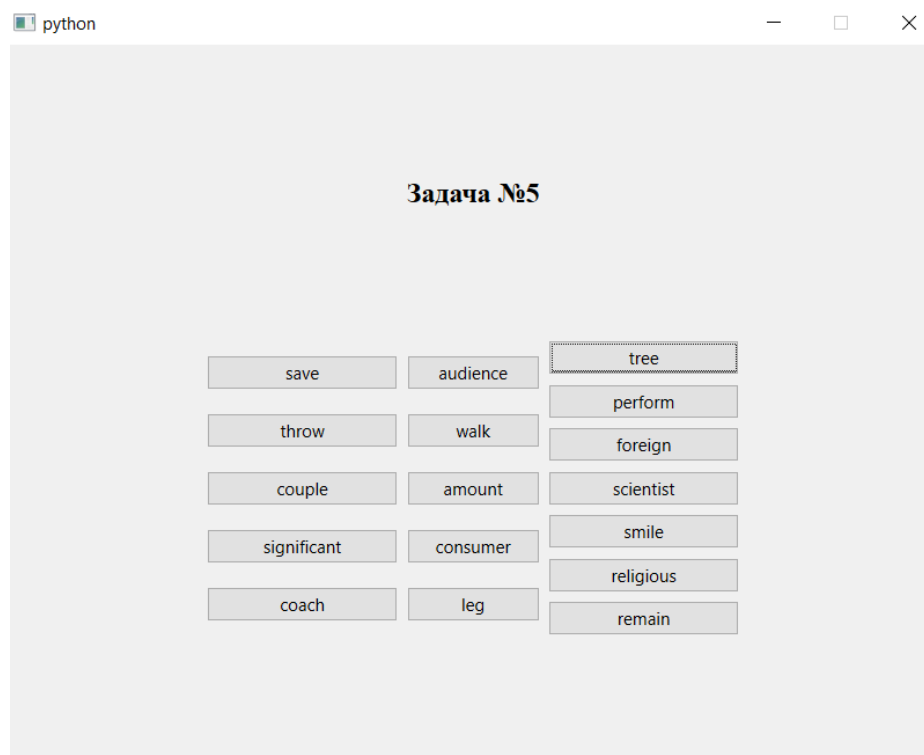


Рис. 3.12: Окно с выбором слов.

После выполнения 4 задачи, пользователю будет доступен последний блок задач для прохождения, откроется соответствующее окно с информацией. После изучения информации пользователь перейдет на страничку с задачей (рис. 3.11), в которой необходимо будет за 1 минуту запомнить 7 случайно сгенерированных английских слов. После истечения заданного времени, пользователя перекинет на страничку с перерывом в 3 минуты, а затем, его автоматически перенесет на страничку (рис. 3.12), где будет необходимо найти и прокликать кнопки, соответствующие словам.

Далее пользователь перейдет на страничку с информацией о последней задаче, ее я прикреплю в разделе [Приложения](#).

После окончания всех задач, пользователь будет переходить на страничку с результатами (рис. 3.13).

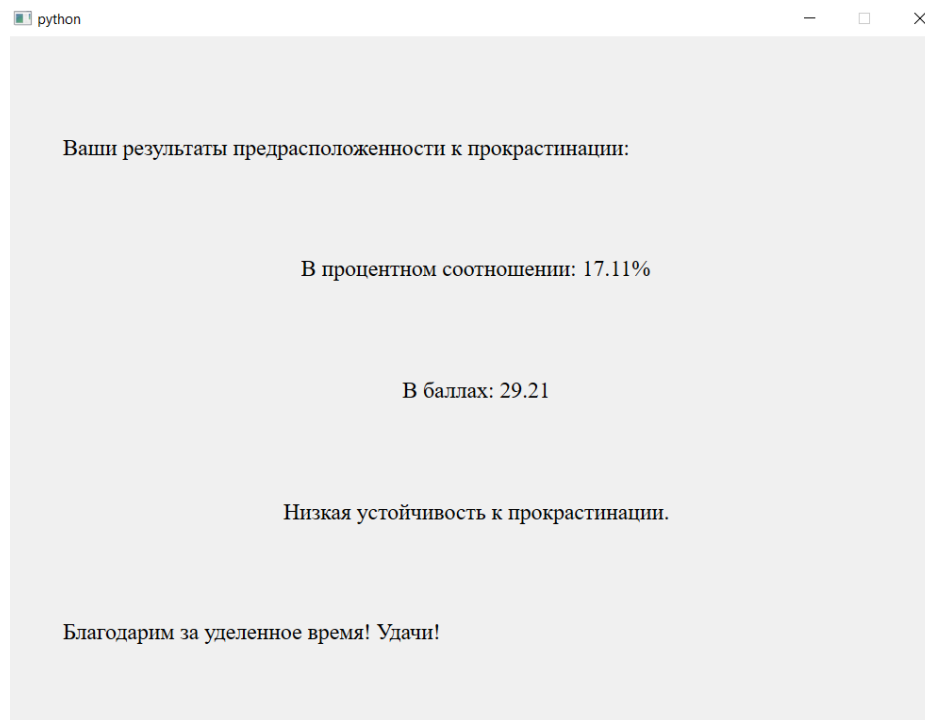


Рис. 3.13: Окно с результатами.

### 3.7 Проведение тестирования, анализ результатов, вывод формулы оценки устойчивости к прокрастинации

#### 3.7.1 Анализ полученных результатов тестирования

Для оценки работы программы собрана небольшая группа участников из 6 человек, которые прошли тестирование. Группая испытуемых была собрана из учащихся в возрасте от 20 до 25 лет. Для того чтобы мы смогли вывести формулу оценивания с помощью разбалловки, был проведен небольшой анализ полученных результатов в среде выполнения google-collab, там считывались данные, значения проверялись на пустоту и далее мы переходили к изучению зависимостей.

Первым, на что мы обратим внимание, это зависимость времени выполнения от количества ошибок(рис. 3.14), красная линия – это линия тренда, она помогает увидеть общую тенденцию: Задачи 3 и 6 оказались более сложными, поскольку количество потраченного времени на них сильно варьируется, как и количество ошибок.



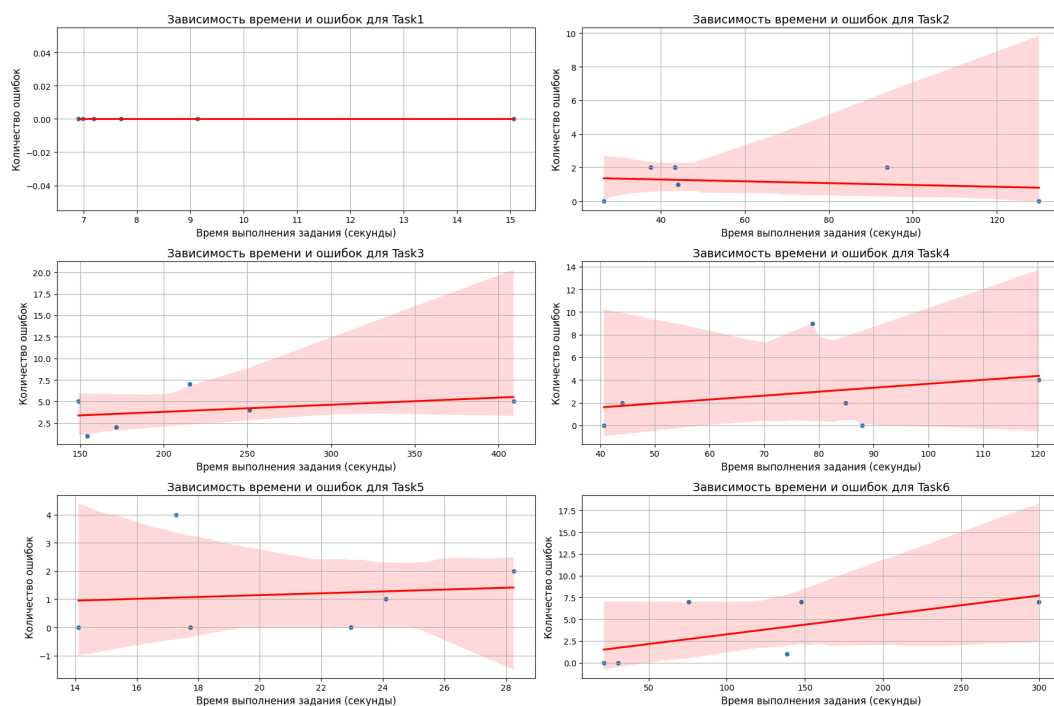


Рис. 3.14: Зависимость времени выполнения задач от количества ошибок.

Также проведен анализ зависимости ошибок от ответа на вопрос, на графике (рис. 3.15) можно заметить, что среднее количество ошибок было значительно выше для тех людей, которые ответили утвердительно на вопрос о желании отвleчься. Получается, что количество ошибок имеет прямую зависимость с желанием человека отвleчься.

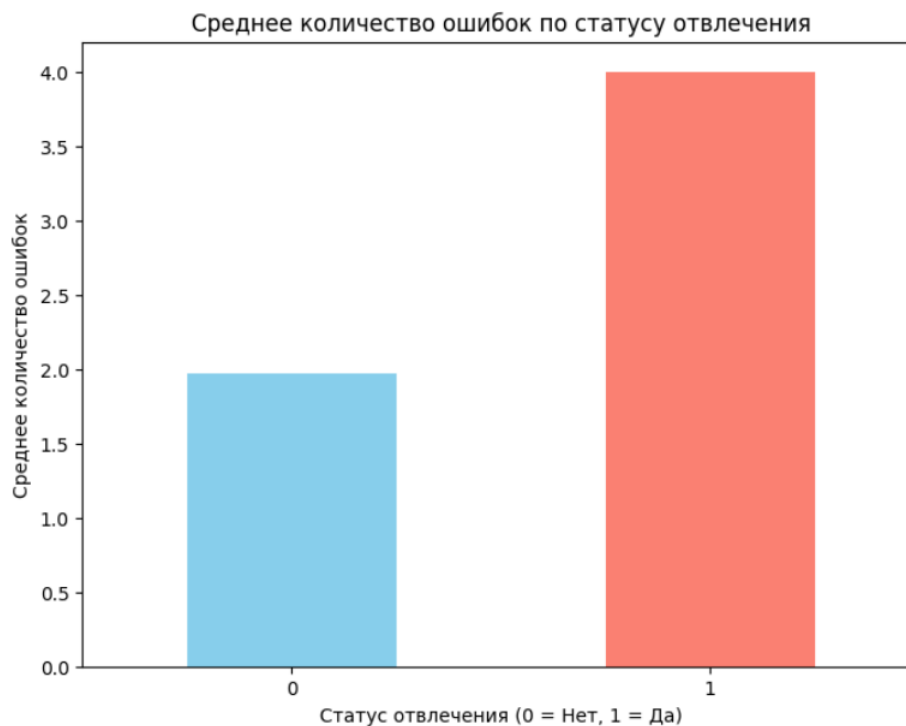


Рис. 3.15: Среднее количество ошибок в зависимости от ответа на вопрос.

Далее рассмотрим зависимость между ответом на вопрос и количеством потраченного на задачу времени. На графике (рис. 3.16) мы можем заметить, что среднее время выполнения задачи значительно выше для тех участников, которые захотели отвлечься, по сравнению с теми, кто не захотел. Значит мы также можем сделать вывод, что отвлечение связано с увеличением потраченного времени.

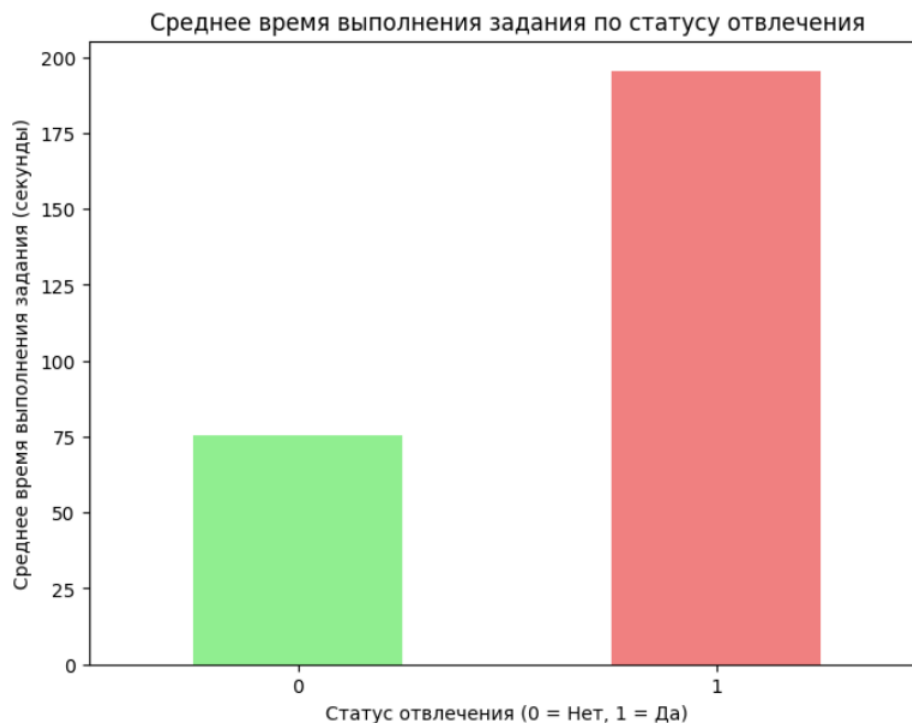


Рис. 3.16: Среднее количество потраченного времени в зависимости от ответа на вопрос.

В результате, на раннем этапе была выведена первичная формула вычисляющая устойчивость человека к прокрастинации, которая основана на идее: чем больше ошибок и времени, тем больше штраф и меньше результирующий балл.

$$\text{score} = \max(0, 100 - (\text{time} \times 0.1 + \text{errors} \times 10 + \text{question} \times 20)),$$

где:

- time — время выполнения задачи.
- errors — количество ошибок.
- question — желание отвлечься (1, если да и 0, если нет).

Теперь на задачу можно посмотреть с вопросом о том, как меняется устойчивость к прокрастинации в зависимости от задачи.

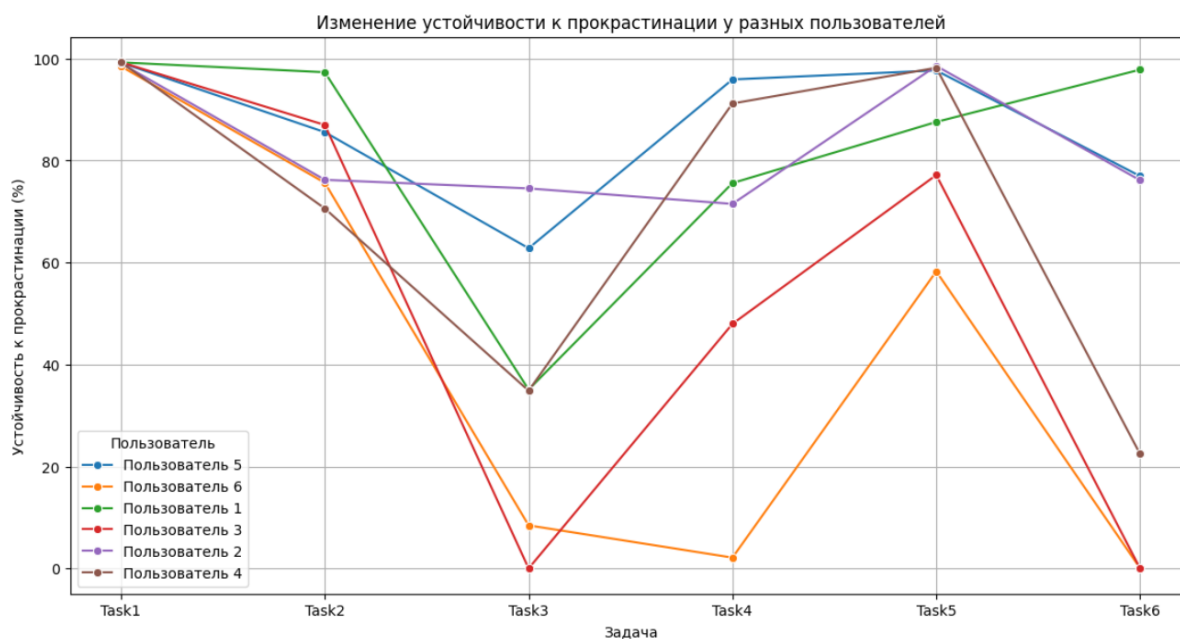


Рис. 3.17: Изменение устойчивости к прокрастинации в зависимости от задачи.

Основные наблюдения:

- Задача 1: Показатели стабильно высокие, значит пользователи показывают хорошую устойчивость к прокрастинации в данной задаче.
- Задача 2: Устойчивость начинает снижаться по мере усложнения задачи.
- Задача 3: Показатели значительно снизились, в сравнении со 2-й задачей, что указывает на ее сложность.
- Задача 4 и 5: В сравнении с задачей 3, показатели значительно повышаются.
- Задача 6: Также как и с задачей 3, показатели сильно снижены из-за ее сложности.

### 3.7.2 Формула оценки с помощью штрафов

Переходя непосредственно к оцениванию устойчивости к прокрастинации на практике, мы не можем четко вывести формулу, поскольку эта задача закреплена за нашими коллегами из лаборатории РУДН. Но можем аналитически, исходя из наших исследований различных зависимостей, предположить как она будет выглядеть. В результате которых получилось вывести две различные формулы.

Первая формула, в которой для каждой задачи мы начинаем с максимального балла, штрафует за превышение определенного временного порога, штрафует за превышение порога ошибок и за желание отвлечься, если оно есть:

Нормализованный итоговый балл:

$$\text{normalized\_total\_score} = \left( \frac{\sum_{i=1}^6 \text{score}_i}{6 \times \text{max\_score}} \right) \times 100,$$

где

$$\begin{aligned} \text{score}_i = & \text{max\_score} \\ & - \left( \frac{\max(0, \text{errors}_i - \text{error\_threshold}_i)}{\text{max\_errors}_i} \times \text{max\_score} \right) \\ & - \left( \frac{\max(0, \text{time}_i - \text{time\_threshold}_i)}{\text{time\_threshold}_i} \times \text{max\_score} \times 0.1 \right) \\ & - (\text{wants\_distraction}_i \times 10). \end{aligned}$$

- Параметры задачи  $i$ :

- Максимальное количество ошибок:  $\text{max\_errors}_i$ ,
- Порог ошибок:  $\text{error\_threshold}_i$ ,
- Порог времени:  $\text{time\_threshold}_i$ ,
- $\text{errors}_i$  — количество ошибок в задаче,
- $\text{time}_i$  — время выполнения задачи,
- $\text{wants\_distraction}_i$  — количество раз, когда пользователь захотел отвлечься.

Но, в таком случае, можно заметить, что все штрафы суммируются, это может привести к тому, что итоговый балл многих участников будет достаточно низким. Это связано с тем, что у нас не хватает ограничений на максимальное количество потраченного времени, поэтому, в следующей формуле, мы постарались учесть данный фактор и нормировали время.

### 3.7.3 Формула оценки выведенная аналитически

В данной задаче мы пытаемся вывести формулу оценивающую устойчивость к прокрастинации в процентах. Мы пытаемся подобрать коэффициенты следующим образом:

Пусть  $T_i$  — время выполнения задачи  $i$ ,  $E_i$  — количество ошибок в задаче  $i$ , и  $\text{distraction}_i$  — параметр отвлечения для задачи  $i$  (1, если было отвлечение, 0 — если не было).

- Параметры задачи  $i$ :

- $\text{max\_errors}_i$  — максимальное количество ошибок.

- $\text{error\_threshold}_i$  — порог ошибок.
- $\text{time\_threshold}_i$  — порог времени.
- $\text{max\_result\_for\_formula}$  — коэффициенты нормализация формулы для каждого из 4-условий для подсчета процентов  $[0.07, 0.12, 0.014, 0.09]$ .
- $\text{mn}$  — коэффициенты нормализации для времени:  $[1, 5, 7, 2, 2, 3]$ .
- $\text{id\_for\_tasks}$  — коэффициенты весов для времени и ошибок соответственно:  $[[0.5, 3], [0.5, 10], [3, 0.5], [3, 3]]$ .

Формула для вычисления результирующего значения для задачи  $i$ :

$$\text{formula}_i = \begin{cases} \frac{1}{\text{id\_for\_tasks}[0][0] \cdot \frac{T_i}{\text{mn}[i]} + \text{id\_for\_tasks}[0][1] \cdot E_i + 1}, & (1) \\ \frac{1}{\text{id\_for\_tasks}[1][0] \cdot \frac{T_i}{\text{mn}[i]} + \text{id\_for\_tasks}[1][1] \cdot E_i}, & (2) \\ \frac{1}{\text{id\_for\_tasks}[2][0] \cdot \frac{T_i}{\text{mn}[i]} + \text{id\_for\_tasks}[2][1] \cdot E_i}, & (3) \\ \frac{1}{\text{id\_for\_tasks}[3][0] \cdot \frac{T_i}{\text{mn}[i]} + \text{id\_for\_tasks}[3][1] \cdot E_i}. & (4) \end{cases}$$

Условия:

- (1) если  $T_i \leq \text{time\_threshold}_i$  и  $E_i \leq \text{error\_threshold}_i$ ,
- (2) если  $T_i \leq \text{time\_threshold}_i$  и  $E_i > \text{error\_threshold}_i$ ,
- (3) если  $T_i > \text{time\_threshold}_i$  и  $E_i \leq \text{error\_threshold}_i$ ,
- (4) если  $T_i > \text{time\_threshold}_i$  и  $E_i > \text{error\_threshold}_i$ .

Нормализация результата  $x_i$ :

$$x_i = \begin{cases} 1, & \text{если } \text{formula}_i > \text{max\_result\_for\_formula}[j] \\ \frac{\text{formula}_i}{\text{max\_result\_for\_formula}[j]}, & \text{иначе.} \end{cases}$$

Корректировка результата с учётом отвлечения:

$$\text{result+} = \begin{cases} x_i \cdot 0.75, & \text{если } \text{distraction}_i = 1 \\ x_i, & \text{иначе.} \end{cases}$$

Итоговый результат для всех задач вычисляется следующим образом:

$$\text{final\_result} = \left( \frac{\text{result}}{6} \right) \cdot 100$$

Полученные формулы были интегрированы в программный код и в результате была выведена градация Хорошая/средняя/низкая устойчивость к прокрастинации следующим образом:

- Если значение переменной **final\_result** находится в диапазоне от 66 до 100 включительно, то показатель устойчивости высокий.
- Если значение переменной **final\_result** находится в диапазоне от 36 до 65 включительно, то показатель устойчивости средний.
- Если значение **final\_result** находится в диапазоне от 0 до 35 включительно, то показатель устойчивости низкий.

## 4 Заключение

В результате проделанной работы мы получили программу, которая оценивает устойчивость человека к прокрастинации и может быть использована в качестве основы для многих работ в сфере когнитивных функций человеческого мозга.

Были проанализированы существующие подходы и метрики, и, получен способ тестирования, основывающийся на 6 основных задачах и отобранных метриках, также была разработана формула оценивания прокрастинации, разработан и внедрен графический интерфейс, непосредственно проведено само тестирование и анализ результатов работы данной программы на небольшой группе людей. В дальнейшем планируется доработка формулы оценивания и тестирующих задач, для более точного результата.

В настоящее время вопрос прокрастинации затрагивает большое количество людей и является одной из важных проблем в науке о когнитивных функциях. Поскольку прокрастинация прямо пропорционально влияет на фактор трудоспособности в той или иной сфере. Отсюда следует, что оценка способности устойчивости к данному явлению, внесет большой вклад в ускорение процесса понимания прокрастинации и во многом поможет в вопросах борьбы с этим явлением. А программа, которая будет включать в себя эту оценку, даст учёным возможность сэкономить большое количество времени и финансовых затрат для изучения данного вопроса.

## Список литературы

- [1] Seo Kim и Eun Hee Seo. «Procrastination: A Positive or Negative Trait?» B: *Personality and Individual Differences* 76 (2015), с. 163—168.
- [2] P.Steel. “Assessment of Procrastination: A Critical Review”. In: *Psychological Bulletin* 133.1 (2007), pp. 65–94.
- [3] P. Ringenbach. *Procrastination in Human Life*. Vol. 1. 1. Institute for Rational Living, 1977, p. 94.
- [4] Laura J Solomon and Esther D Rothblum. “The Procrastination Assessment Scale-Students (PASS)”. In: *Psychological Reports* 54.1 (1984), p. 122.
- [5] Piers Steel. “Development and Validation of the General Procrastination Scale (GPS)”. In: *Journal of Counseling Psychology* 53.4 (2010), p. 483.
- [6] Mark Summerfield. *Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming*. Vol. 1. 1. Prentice Hall, 2007, p. 648.



## Приложения

В данном разделе добавлена ссылка на github-репозиторий с реализованной программой.

Ссылка на репозиторий: <https://github.com/Naumcheck/procrastination-program>.

Пример окна с перерывом для соответствующих задач:

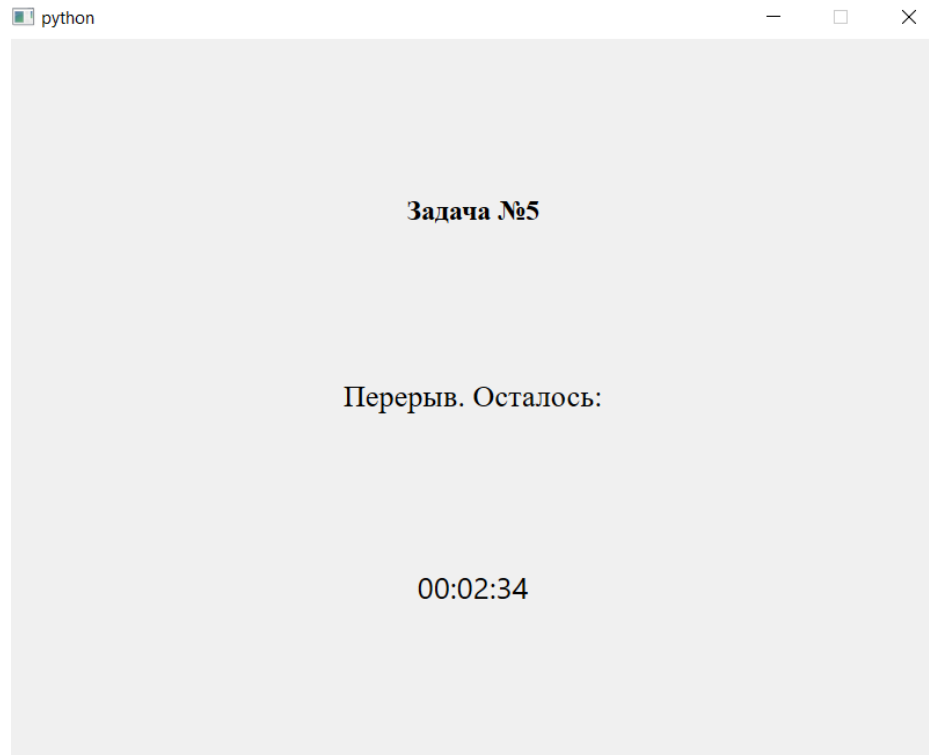


Рис. 4.1: Окно перерыва.

## Задача №6:

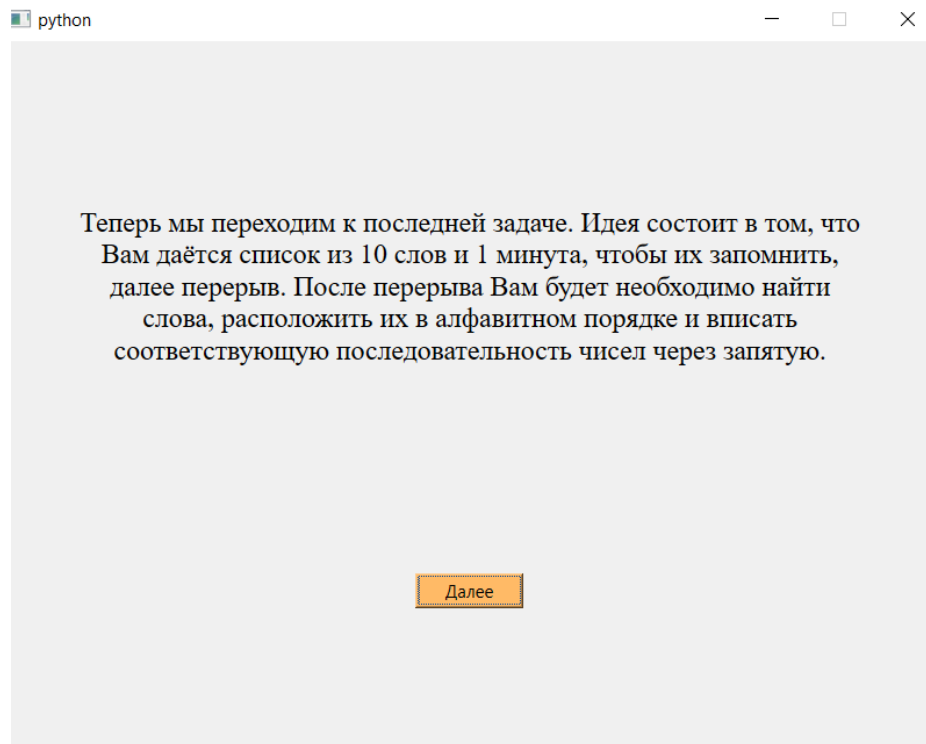


Рис. 4.2: Окно с информацией о задаче.

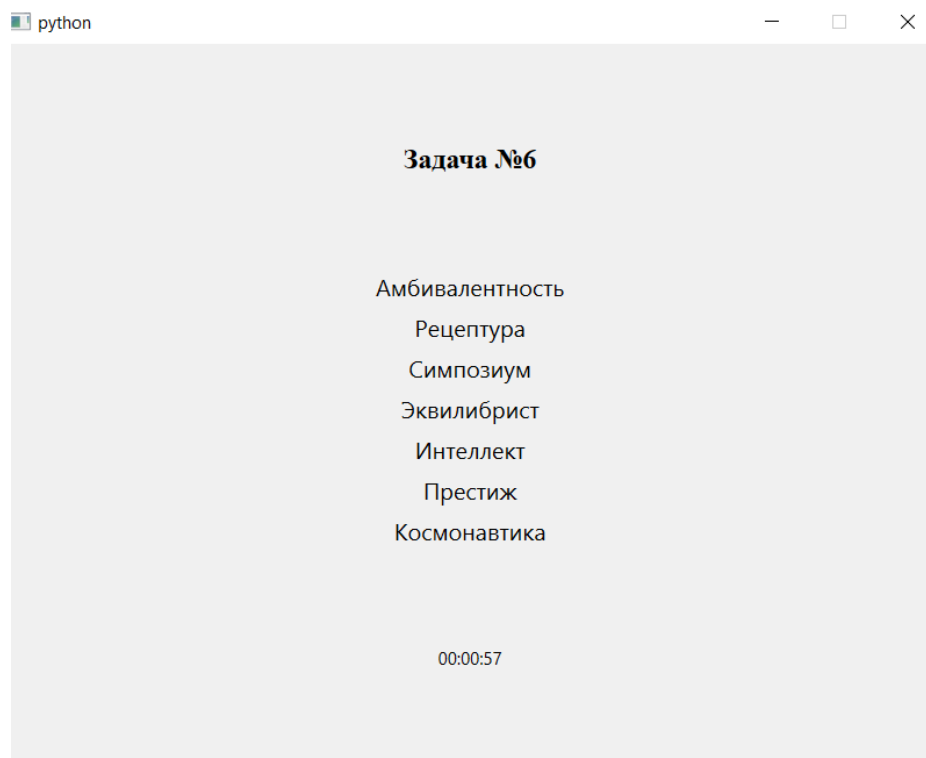


Рис. 4.3: Окно со словами.

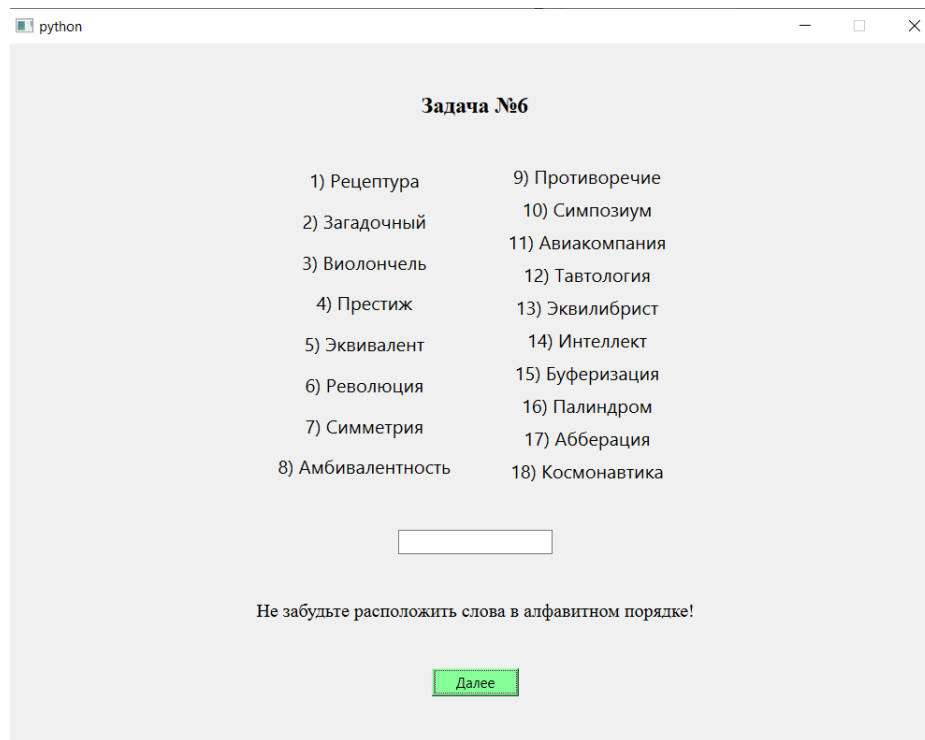


Рис. 4.4: Окно с задачей.