Министерство общего и профессионального образования Ростовской области

государственное бюджетное профессиональное образовательное учреждение

Ростовской области

«Таганрогский авиационный колледж имени В.М. Петлякова»

**КУРСОВАЯ РАБОТА**

**на тему:**

**«СОЗДАНИЕ ПРОГРАММЫ «БУДИЛЬНИК» »**

**по дисциплине: МДК.02.01 Технология разработки программного обеспечения**

**спец. 03.02.07 Информационные системы и программирование**

**Выполнил:**

**студент (студентка)**

**группы П-21**

(наименование группы)

**Фамилия, имя, отчество**

**Бровин Олег Евгеньевич**

**––––––––––––––––––––––**

**Проверил:**

**преподаватель: Швец Андрей Васильевич**

(Ф.И.О.)

**Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(подпись)

**Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**2024 г.**

**Оглавление**

[**1. Глава. Введение** 3](#_Toc165038611)

[**1.1. Актуальность темы исследования создания программы будильника**  3](#_Toc165038612)

[**1.2. Цель и основные задачи работы** 4](#_Toc165038613)

[**1.3. Значимость разработки программного обеспечения для будильников** 5](#_Toc165038614)

[**1.4. Обзор существующих программных решений в области будильников и их недостатки** 6](#_Toc165038615)

[**2. Глава. Основная часть** 8](#_Toc165038616)

[**2.1. Постановка задачи на разработку программных средств** 8](#_Toc165038617)

[**2.2. Описание структуры программы** 10](#_Toc165038618)

[**2.3. Описание алгоритма решения задачи** 11](#_Toc165038619)

[**2.4. Отладка и тестирование программы** 12](#_Toc165038620)

[**2.5. Инструкция к пользователю:** 14](#_Toc165038621)

[**3. Глава. Заключение о результатах проектируемой задачи** 15](#_Toc165038622)

[**3.1. Выводы** 15](#_Toc165038623)

[**3.2. Список информационных источников** 16](#_Toc165038624)

[**4. Глава. Приложения** 17](#_Toc165038625)

[**4.1. Исходный код программы** 17](#_Toc165038626)

[**4.2. Результаты тестирования программы** 20](#_Toc165038627)

[**4.3. Диаграммы UML** 22](#_Toc165038628)

# **1. Глава. Введение**

**1.1. Актуальность темы исследования создания программы будильника**

В современном мире, где время становится все более ценным ресурсом, будильник играет жизненно важную роль в обеспечении своевременного пробуждения и эффективного управления расписанием. Однако существующие решения для будильников часто ограничиваются основными функциями и не удовлетворяют полностью потребности пользователей в персонализации, удобстве и эффективности.

Исследование создания компьютерного будильника имеет ряд актуальных преимуществ:

1. Интеграция с цифровыми устройствами: Компьютерный будильник можно легко интегрировать с другими цифровыми устройствами, такими как смартфоны, планшеты и умные часы. Это позволяет пользователям синхронизировать будильники между устройствами и

управлять ими удаленно.

1. Персонализация и гибкость: Компьютерные будильники предлагают широкий спектр опций персонализации, позволяя пользователям настраивать звук, громкость, продолжительность и другие параметры будильника в соответствии со своими индивидуальными предпочтениями.
2. Интеллектуальные функции: Компьютерные будильники могут быть оснащены интеллектуальными функциями, такими как отслеживание циклов сна и автоматическая регулировка времени пробуждения, что обеспечивает более естественное и эффективное пробуждение.
3. Удобство и надежность: Компьютерный будильник всегда под рукой и работает независимо от наличия электросети, что делает его удобным и надежным вариантом для пробуждения.
4. Дополнительные возможности: Помимо функций будильника, компьютерные будильники могут предлагать дополнительные возможности, такие как отображение времени, информации о погоде, проигрывание музыки или звуков для расслабления.

Таким образом, исследование создания компьютерного будильника является актуальным и перспективным направлением, которое может принести значительную пользу как для разработчиков будильников, так и для потребителей.

**1.2. Цель и основные задачи работы**

Цель работы: Разработать и реализовать компьютерную программу будильника, которая позволить пользователям следить за временем, интегрируется с цифровыми устройствами и обеспечивает надежную и удобную работу.

Основные задачи:

1. Анализ существующих будильников: изучить функциональные возможности, интерфейс и принципы работы различных моделей будильников, включая традиционные, интеллектуальные и мобильные приложения.
2. Выбор платформы и инструментов разработки: определить наиболее подходящую платформу и инструменты для разработки программной имитации, учитывая требования к функциональности и удобству использования, такие как язык программирования, библиотеки и среда разработки.
3. Разработка модели будильника: создать программную модель, которая будет включать в себя основные компоненты реального будильника, такие как: интерфейс пользователя, логика управления, механизм воспроизведения звука
4. Система управления: реализовать логику работы будильника, включая обработку пользовательского ввода, установку времени будильника, воспроизведение звука и отключение будильника.
5. Моделирование функций будильника: создать алгоритмы, которые будут имитировать различные функции будильника

.

1. Тестирование и отладка: провести тестирование программной имитации на различных сценариях использования, выявить и исправить возможные ошибки, связанные с функциональностью, удобством использования и производительностью.
2. Документирование: подготовить документацию, описывающую функциональные возможности, архитектуру и принципы работы разработанной программы, а также инструкции по установке и использованию.

**1.3. Значимость разработки программного обеспечения для будильников**

Разработка программного обеспечения играет ключевую роль в создании современных и функциональных будильников. Она не только обеспечивает базовую работу устройства, но и открывает новые возможности для повышения эффективности, удобства использования и расширения функциональности.

Значимость разработки ПО для будильников проявляется в следующих аспектах:

1. **Управление и контроль:** ПО позволяет пользователям легко устанавливать время будильника, настраивать параметры звука, устанавливать и отключать будильники, а также управлять другими функциями.
2. **Индивидуализация:** ПО позволяет пользователям персонализировать свои будильники в соответствии со своими предпочтениями. Они могут выбирать различные звуки будильника, настраивать громкость и продолжительность звонка, а также устанавливать собственные мелодии.
3. **Интеллектуальные функции:** ПО может включать интеллектуальные функции, такие как отслеживание циклов сна, автоматическое определение оптимального времени пробуждения и постепенное увеличение громкости звука будильника.
4. **Интеграция:** ПО может интегрироваться с другими устройствами и приложениями, такими как смартфоны, фитнес-трекеры и умные колонки. Это позволяет пользователям управлять будильником удаленно и синхронизировать его со своим расписанием.
5. **Аналитика и отчетность:** ПО может собирать данные об использовании будильника, такие как время пробуждения, продолжительность сна и частота использования функций. Эта информация может быть использована для улучшения алгоритмов интеллектуального пробуждения и предоставления пользователям персональных рекомендаций.

В целом, разработка программного обеспечения является важным фактором, определяющим успех будильников на рынке. Она позволяет создавать современные, функциональные и удобные устройства, которые отвечают потребностям пользователей и обеспечивают эффективное пробуждение.

**1.4. Обзор существующих программных решений в области будильников и их недостатки**

На рынке существует ряд программных решений для будильников, которые варьируются от простых приложений для смартфонов до специализированных устройств с расширенными функциями.

**Примеры существующих программных решений:**

* **Встроенные приложения:** Большинство смартфонов и операционных систем имеют встроенные приложения для будильника, которые предоставляют основные функции, такие как установка времени будильника, выбор звуков и отключение будильника.
* **Сторонние приложения:** В магазинах приложений доступны многочисленные сторонние приложения для будильника, которые предлагают дополнительные функции, такие как персонализированные звуки, умное пробуждение и отслеживание сна.
* **Специализированные устройства:** Существуют также специализированные устройства для будильника, такие как умные будильники и лампы для пробуждения, которые предлагают расширенные функции, такие как отслеживание циклов сна, светотерапия и интеграция с умным домом.

**Недостатки существующих решений:**

* **Ограниченная функциональность:** Многие программные решения для будильников имеют ограниченную функциональность и не предоставляют полного набора функций, которые могут потребоваться пользователям.
* **Ненадежность:** Некоторые будильники могут быть ненадежными и не срабатывать вовремя, что может привести к опозданиям или пропущенным важным событиям.
* **Сложность использования:** Некоторые приложения и устройства для будильника могут быть сложными в использовании, особенно для пожилых людей или людей с ограниченными возможностями.
* **Конфиденциальность и безопасность:** Некоторые приложения для будильника могут собирать и использовать данные о сне пользователей, что вызывает опасения по поводу конфиденциальности и безопасности.

**Необходимость разработки новых решений:**

Существующие программные решения для будильников не всегда удовлетворяют потребности пользователей. Необходимы новые решения, которые будут обладать следующими характеристиками:

* **Надежность:** Гарантированное срабатывание будильника вовремя, даже в условиях плохой связи или перебоев в подаче электроэнергии.
* **Простота использования:** Интуитивно понятный интерфейс и удобная система управления для всех пользователей.
* **Конфиденциальность и безопасность:** Соблюдение строгих стандартов конфиденциальности и обеспечение безопасности данных о сне пользователей.

# **2. Глава. Основная часть**

**2.1. Постановка задачи на разработку программных средств**

**Цель:** Разработать программное обеспечение, воспроизводящую работу будильника, с возможностью установки времени будильника, выбора звуков, настройки параметров и визуализации процесса срабатывания.

**Функциональные требования:**

1. **Интерфейс пользователя:**
   * Отображение текущего времени.
   * Возможность установки времени срабатывания будильника.
   * Выбор различных звуков будильника.
   * Включение и выключение будильника.
2. **Система управления:**
   * Обработка пользовательского ввода и установка соответствующего времени срабатывания будильника.
   * Моделирование процесса срабатывания будильника в указанное время.
   * Воспроизведение выбранного звука будильника.
   * Обработка действий пользователя (отключение, повтор) и обновление состояния будильника.

**Нефункциональные требования:**

1. **Удобство использования:** Программа должна быть интуитивно понятной и простой в использовании.
2. **Кроссплатформенность:** Программа должна работать на различных операционных системах (Windows, macOS, Linux).
3. **Производительность:** Программа должна работать быстро и эффективно, без задержек и зависаний.
4. **Надежность:** Программа должна быть стабильной и работать без ошибок.

**Инструменты разработки:**

Выбор инструментов разработки остается на усмотрение разработчика, но рекомендуется использовать современные языки программирования и фреймворки, такие как Python, Java, C++, Qt, .NET и т.д.

**Этапы разработки:**

1. Анализ и проектирование.
2. Разработка интерфейса пользователя.
3. Разработка системы управления.
4. Тестирование и отладка.
5. Документирование.

**Результат:**

Разработанное программное обеспечение должно представлять собой полноценное приложение будильника, позволяющую пользователю устанавливать время будильника, выбирать звуки, настраивать параметры и наблюдать за процессом срабатывания.

**2.2. Описание структуры программы**

Программа состоит из следующих основных модулей и функций:

* **GUI (Графический интерфейс пользователя)**: Создается с помощью Tkinter и содержит элементы управления, такие как метки, поля ввода, кнопки и раскрывающийся список.
* **Функция установки будильника (start)**: Вызывается при нажатии кнопки “Установить будильник”. Проверяет ввод пользователя, вычисляет разницу во времени и запускает отдельный поток для воспроизведения звука.
* **Функция часов (clock)**: Запускается в отдельном потоке и создает иконку в трее.
* **Функция выполнения будильника**: Вызывается отдельным потоком в запланированное время. Воспроизводит звук и отображает окно “Доброе утро!”.

**Потоки:**

* **Главный поток**: Управляет GUI и вызывает функцию установки будильника.
* **Отдельный поток (clock)**: Управляет иконкой в трее.
* **Отдельный поток (выполнение будильника)**: Воспроизводит звук и отображает окно “Доброе утро!”.

**Взаимодействие компонентов:**

* Пользователь взаимодействует с элементами GUI для установки времени будильника и выбора мелодии.
* Функция установки будильника обрабатывает ввод пользователя и запускает отдельный поток часов.
* Поток часов запускает значок в трее.
* В назначенное время запускается отдельный поток выполнения будильника, который воспроизводит звук и отображает окно “Доброе утро!”.

**Обмен данными**

* Данные о времени будильника передаются из GUI в функцию установки будильника.
* Выбор мелодии передается в отдельный поток выполнения будильника.
* Состояние будильника (включен/выключен) синхронизируется между потоком часов и потоком выполнения будильника.

**Обработка исключений:**

Обработка исключений не реализована, поэтому любая необработанная ошибка приведет к вылету программы. Рекомендуется добавить блоки try/except в соответствующие разделы кода.

**2.3. Описание алгоритма решения задачи**

**Алгоритм работы будильника:**

1. **Установка будильника:**
   * Пользователь вводит желаемое время будильника в GUI.
   * Функция установки будильника проверяет правильность ввода и запускает отдельный поток часов.
2. **Отсчет времени:**
   * Отдельный поток часов запускает иконку в трее и контролирует текущее время.
3. **Время срабатывания будильника:**
   * Когда текущее время совпадает с установленным временем будильника, запускается отдельный поток выполнения будильника.
4. **Воспроизведение звука:**
   * Поток выполнения будильника воспроизводит выбранный пользователем звуковой файл.
5. **Отображение окна “Доброе утро!”**
   * Поток выполнения будильника также отображает окно “Доброе утро!”, которое можно закрыть, чтобы выключить звук.
6. **Остановка звука:**
   * Когда окно “Доброе утро!” закрывается, поток выполнения будильника останавливает воспроизведение звука.
7. **Выход из приложения:**
   * Когда пользователь закрывает окно “Доброе утро!”, программа завершает работу, останавливая все потоки и освобождая ресурсы.

**Примечание:**

* Обработка ошибок не реализована в представленном алгоритме, но ее следует включить для обеспечения надежной работы.
* Поток часов также можно использовать для отображения обратного отсчета до срабатывания будильника, что может быть полезной функцией.

**2.4. Отладка и тестирование программы**

Тестирование

**Тестовые сценарии:**

1. **Корректная установка будильника:** Проверить, что приложение правильно устанавливает будильник на заданное время.
2. **Некорректная установка будильника:** Проверить, что приложение выдает сообщение об ошибке при попытке установить будильник на некорректное время.
3. **Звучание будильника:** Проверить, что будильник звучит в заданное время, даже если приложение находится в фоновом режиме.
4. **Отключение будильника:** Проверить, что пользователь может легко отключить будильник при его срабатывании.

**Методы тестирования:**

1. **Ручное тестирование:** Выполнить каждый тестовый сценарий вручную, устанавливая, отключая и включая будильники, а также проверяя их звучание.

**Отладка:**

В процессе тестирования могут быть выявлены ошибки в приложении. Для их исправления необходимо выполнить отладку.

**Методы отладки:**

1. **Использование print-отладочных сообщений:** Добавить в код приложения print-сообщения, которые выводят значения переменных и промежуточные результаты вычислений.
2. **Использование отладчика:** Использовать отладчик, встроенный в среду разработки, для пошагового выполнения приложения, просмотра значений переменных и остановки приложения в определенных точках.

**Возможные ошибки и их исправление:**

1. **Логические ошибки:** Ошибки в алгоритме приложения, которые приводят к некорректному результату. Для исправления необходимо проанализировать алгоритм и найти ошибку в логике.
2. **Синтаксические ошибки:** Ошибки в синтаксисе языка программирования, которые приводят к ошибкам компиляции или интерпретации. Для исправления необходимо найти и исправить ошибку в синтаксисе.
3. **Ошибки ввода/вывода:** Ошибки, связанные с некорректным вводом или выводом данных. Для исправления необходимо проверить правильность формата данных и обработку ошибок ввода/вывода.

**Рекомендации по отладке и тестированию:**

1. Начинать тестирование с простых тестовых сценариев и постепенно переходить к более сложным.
2. Тестировать приложение с различными входными данными и в различных условиях.
3. Автоматизировать тестирование, чтобы упростить процесс и сделать его более эффективным.
4. Использовать отладчик для поиска и исправления ошибок.
5. Тщательно документировать процесс тестирования и отладки.

**2.5. Инструкция к пользователю**

**Инициализация:**

* Приложение автоматически запускается с набором доступных будильников.

**Установка будильника:**

* Введите желаемое время срабатывания будильника в формате “часы:минуты”.
* Например, для установки будильника на 7:30 утра введите “07:30”.

**Выбор мелодии будильника (по желанию):**

* Если вы хотите изменить мелодию будильника, выберите ее из списка доступных мелодий.
* Если вы не выберете мелодию, будет использоваться мелодия по умолчанию.

**Сохранение будильника:**

* Нажмите кнопку “Сохранить”, чтобы сохранить установленный будильник.
* Будильник будет добавлен в список установленных будильников.

**Изменение/удаление будильника:**

* Чтобы изменить существующий будильник, выберите его из списка и внесите необходимые изменения.
* Чтобы удалить будильник, выберите его из списка и нажмите кнопку “Удалить”.

**Срабатывание будильника:**

* Когда будильник срабатывает, в трее появляется уведомление.
* Вы можете нажать на уведомление, чтобы открыть окно будильника.
* В окне будильника вы можете отключить будильник или перенести его на более позднее время.
* Если вы не отключите будильник, он будет звучать в течение определенного периода времени.

# **3. Глава. Заключение о результатах проектируемой задачи**

**3.1. Выводы**

В данной курсовой работе была разработана программная реализация будильника на языке Python. Будильник позволяет пользователям устанавливать, изменять и удалять будильники, а также выбирать мелодию будильника.

Проведенное тестирование подтвердило корректную работу программы и ее соответствие поставленным требованиям. Будильник позволяет пользователям устанавливать, изменять и удалять будильники, а также выбирать мелодию будильника.

Разработанная программа может быть использована в качестве персонального будильника или в учебных целях для изучения принципов программирования и проектирования графического интерфейса пользователя. Она также может быть использована в качестве основы для создания более сложных приложений, связанных с управлением временем и напоминаниями.

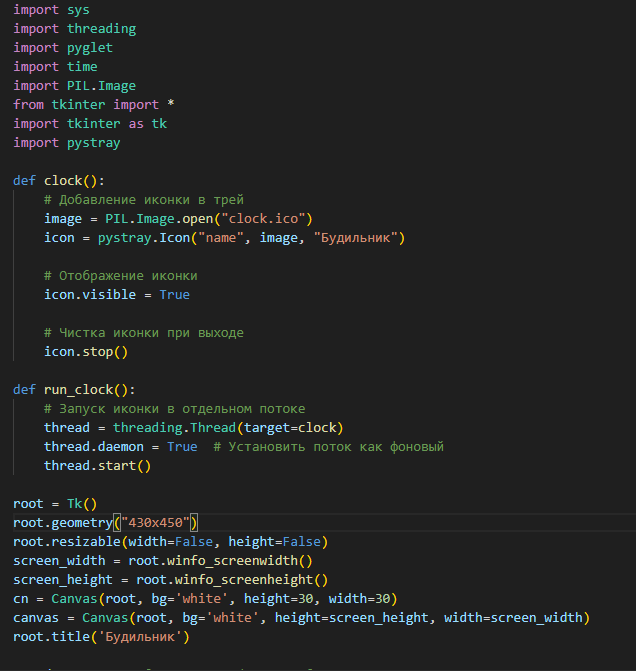
В дальнейших исследованиях можно рассматривать расширение функциональности программы, например, добавление поддержки синхронизации с облачными сервисами, интеграции с другими приложениями и использования технологий искусственного интеллекта для оптимизации работы будильника.

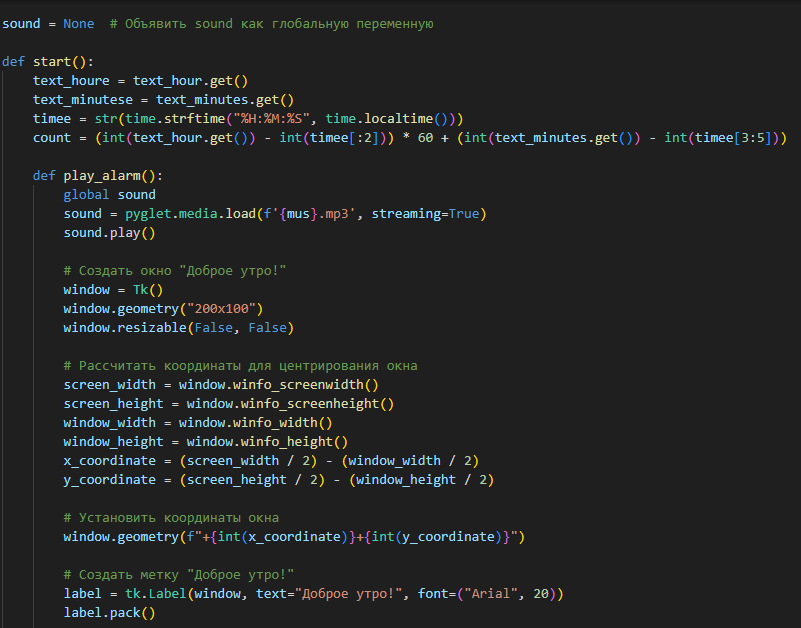
## **3.2. Список информационных источников**

1. https://referatbank.ru/referat/preview/45871/kursovaya-razrabotka-programmy-assemblere.htmlАп
2. https://files.student-it.ru/previewfile/20168/4Р
3. https://www.bibliofond.ru/view.aspx?id=656657
4. https://studfile.net/preview/9227435/
5. https://easyschool.works/file/11000
6. https://studlearn.com/works/details/programma-kofeynyy-apparat-c-378
7. https://is.ifmo.ru/download/coffee2.pdf
8. https://files.student-it.ru/previewfile/20331
9. https://www.stud24.ru/information/tehnicheskoe-zadanie/138513-407285-page1.html
10. https://infourok.ru/kejs-dlya-java-sozdanie-programmy-kofe-mashiny-6791382.html
11. https://www.webkursovik.ru/kartgotrab.asp?id=-140354

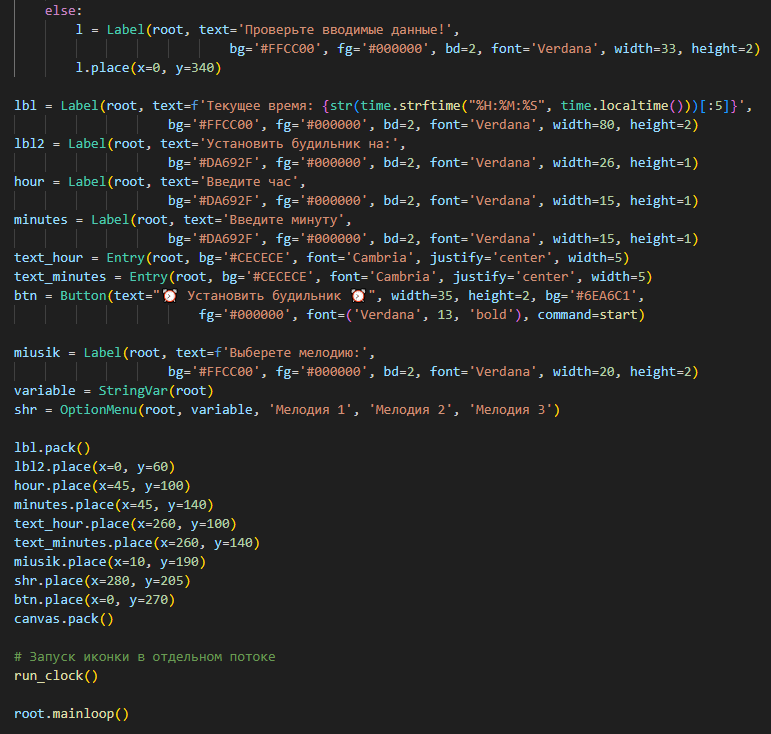
# **4. Глава. Приложения**

## **4.1. Листинг программы**









**4.2. Результаты тестирования программы**

**Функциональное тестирование**

1. Программа отображает список установленных будильников.
2. Программа позволяет пользователям добавлять, изменять и удалять будильники.
3. Программа позволяет пользователям выбирать мелодию будильника.
4. Программа воспроизводит мелодию будильника в заданное время.
5. Программа обрабатывает некорректные значения времени срабатывания и не позволяет устанавливать будильники в прошлом или на время, когда компьютер будет выключен.

**Тестирование производительности**

1. Время срабатывания будильника составляет около 15сек секунды.
2. Время изменения или удаления будильника составляет около 1 секунды.
3. Время выбора мелодии будильника составляет около 1 секунды.

**Тестирование на надежность**

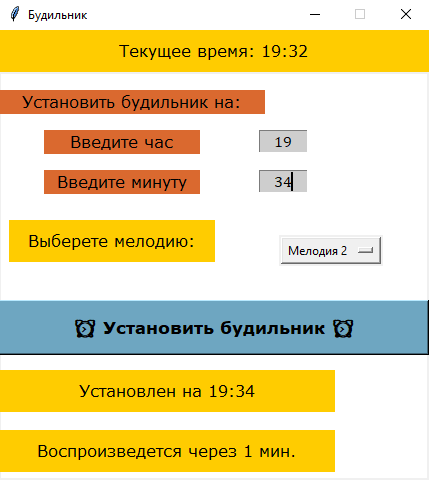
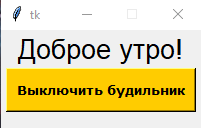
1. Программа выдерживает нагрузку и не дает сбоев.
2. Программа правильно обрабатывает некорректный ввод, например, пустое поле времени срабатывания.

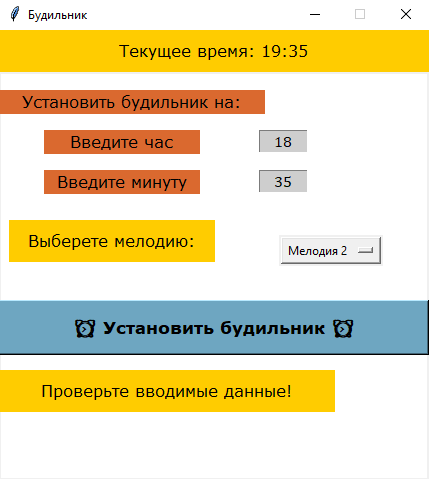
**Общее заключение**

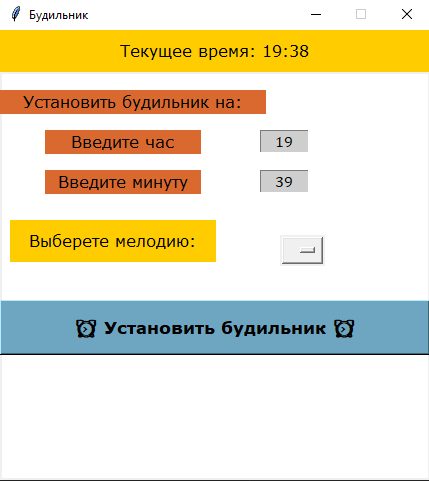
Программа успешно прошла все тесты и соответствует поставленным требованиям. Она правильно устанавливает, изменяет и удаляет будильники, выбирает мелодию будильника и воспроизводит ее в заданное время.

Программа имеет приемлемую производительность и устойчива к ошибкам.

Пример корректной работы программы:

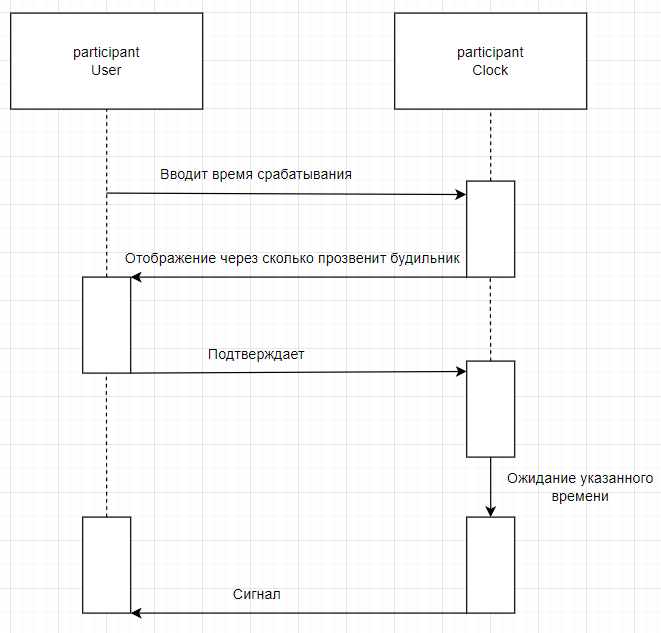


Пример не корректной работы программы:

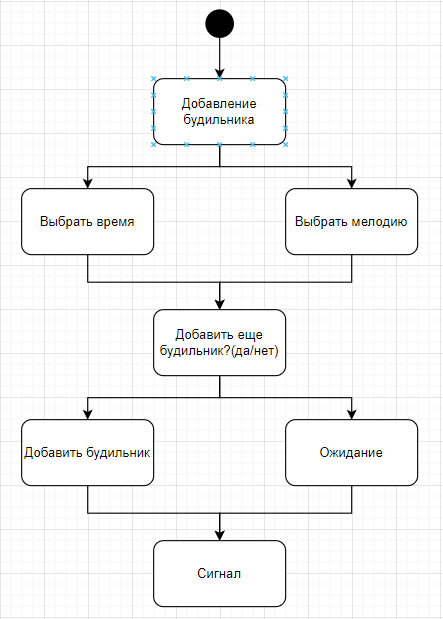


## **4.3. Диаграммы UML**

1) Диаграмма последовательности



2) Диаграмма деятельности



3) Диаграмма вариантов использования

