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:**

**«НАПИСАНИЕ ПРОГРАММЫ, КОТОРАЯ НА ПОВЕРХНОСТЬ ФОРМЫ ВЫВОДИТ ИЗОБРАЖЕНИЕ ИДУЩИХ ЧАСОВ**

**С ЧАСОВОЙ, МИНУТНОЙ И СЕКУНДНОЙ СТРЕЛКАМИ»**

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

**спец. 09.02.07 Информационные системы и программирование**

**(базовая подготовка)**

**Выполнил:**

**студент (студентка)**

**группы П-21**

(наименование группы)

**Фамилия, имя, отчество**

**Дёшин Максим Сергеевич**

**Проверил:**

**преподаватель: Швец Андрей Васильевич**

(Ф.И.О.)

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

(подпись)

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

**2024 г.**

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc165023415)

[ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ОТОБРАЖЕНИЯ ВРЕМЕНИ НА АНАЛОГОВЫХ ЧАСАХ 5](#_Toc165023416)

[1.1 Принципы работы аналоговых часов 5](#_Toc165023417)

[1.2 Основные элементы часового механизма 5](#_Toc165023418)

[1.3 Математические вычисления для отображения времени 6](#_Toc165023419)

[ГЛАВА 2. РЕАЛИЗАЦИЯ ФУНКЦИОНАЛЬНОСТИ ЧАСОВ НА PYTHON 8](#_Toc165023420)

[2.1 UML-диаграммы и их роль в проекте: 8](#_Toc165023421)

[2.2 Разработка графического интерфейса и часовой механики 13](#_Toc165023422)

[ЗАКЛЮЧЕНИЕ 16](#_Toc165023423)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 17](#_Toc165023424)

[ПРИЛОЖЕНИЕ 1 18](#_Toc165023425)

[ПРИЛОЖЕНИЕ 2 19](#_Toc165023426)

# **ВВЕДЕНИЕ**

В современном информационном обществе, где каждый день приносит новые вызовы и возможности, умение эффективно управлять временем становится ключевым навыком. С каждым проходящим мгновением нам требуется быть в курсе текущего времени, будь то для планирования рабочего дня, организации встреч, контроля времени тренировок или отдыха. В такой динамичной и интенсивной среде создание программ, способных точно и наглядно отображать время, играет крайне важную роль.

Аналоговые часы, вопреки всему цифровому разнообразию, по-прежнему остаются одним из самых простых и понятных инструментов визуализации времени. Их преимущества неоспоримы: они предоставляют точную информацию о времени, но также позволяют воспринимать его в более естественной форме, что важно для нашего психологического комфорта. Аналоговые часы, будучи частью нашей повседневной жизни, создают атмосферу уюта и спокойствия, в отличие от цифровых аналогов, которые могут казаться более холодными и механическими.

В этом контексте создание программ, способных отображать время в удобной и эстетичной форме, становится не только актуальным, но и важным вкладом в повышение эффективности и комфорта нашей повседневной деятельности. Необходимость в программных решениях, которые помогут нам лучше ориентироваться во времени и управлять своими задачами, лишь растет с ростом информационной нагрузки и темпов жизни.

Таким образом, разработка программ, способных отображать время в формате аналоговых часов, имеет не только практическое значение для облегчения нашей повседневной жизни, но и символическое значение, напоминая нам о важности времени и его уникальной ценности.

**Цель работы:** разработать программу на языке Python для создания графического интерфейса, отображающего аналоговые часы с часовой, минутной и секундной стрелками.

**Задачи курсовой работы:**

1. Изучить основные принципы работы аналоговых часов и необходимые компоненты для их создания.

2. Разработать математические формулы для определения углов поворота стрелок в зависимости от текущего времени.

3. Провести анализ и разработать UML-диаграммы, необходимые для моделирования различных аспектов проекта.

4. Создать графический интерфейс приложения с использованием библиотеки Tkinter, учитывая принципы часовой механики.

5. Написать код для создания окна приложения с использованием библиотеки Tkinter и холста для реализации аналоговых часов, соответствующий UML-диаграммам.

6. Настроить обновление отображения времени каждую секунду с использованием модуля "time" для корректного отображения текущего времени.

# **ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ОТОБРАЖЕНИЯ ВРЕМЕНИ НА АНАЛОГОВЫХ ЧАСАХ**

## **1.1 Принципы работы аналоговых часов**

Аналоговые часы представляют собой механические устройства, использующие механические движители для отображения времени на циферблате. Они базируются на принципе использования движения стрелок для показа текущего времени. Основные компоненты аналоговых часов включают в себя циферблат, стрелки и часовой механизм.

Принцип работы аналоговых часов заключается в передаче движения механизма на стрелки, которые перемещаются по циферблату, указывая на текущее время. Каждая стрелка связана с определенным механизмом, который регулирует ее скорость и направление движения. Таким образом, в результате работы часового механизма стрелки плавно перемещаются по циферблату, показывая точное время.

Аналоговые часы обычно имеют три стрелки: часовую, минутную и секундную. Часовая стрелка показывает текущее количество часов, минутная - минут, а секундная - секунд. Эти стрелки располагаются на циферблате таким образом, чтобы обеспечить максимальную четкость и удобство чтения времени.

Использование аналоговых часов имеет свои преимущества, такие как более естественное восприятие времени и создание атмосферы уюта. Однако для корректного отображения времени на аналоговых часах необходимо понимание их принципов работы и взаимодействия компонентов.

## **1.2 Основные элементы часового механизма**

Часовой механизм аналоговых часов состоит из нескольких основных элементов, каждый из которых выполняет определенную функцию, необходимую для правильного отображения времени. Рассмотрим эти элементы более подробно:

Оси: Оси являются основными стержнями или валами, вокруг которых вращаются различные части механизма. Они обеспечивают стабильность и правильное распределение движения внутри часового механизма. Каждая стрелка аналоговых часов имеет свою ось, вокруг которой она вращается, показывая текущее время.

Шестерни: Шестерни представляют собой зубчатые колеса, которые передают движение от одной части механизма к другой. Они играют ключевую роль в механизме передачи движения, обеспечивая правильную скорость и направление вращения стрелок. Шестерни могут быть различных размеров и форм, в зависимости от конкретной функции, которую они выполняют в часовом механизме.

Пружины: Пружины являются основными источниками энергии для часового механизма. Они накручиваются вручную или автоматически и высвобождают свою энергию, обеспечивая движение стрелок. Принцип работы пружин заключается в их натяжении и расслаблении, что обеспечивает постоянное движение часового механизма.

Регуляторы: Регуляторы представляют собой устройства, контролирующие скорость и точность движения стрелок. Они могут быть в виде механических механизмов, регулирующих длину стрелок, или электронных устройств, подстраивающих частоту работы механизма. Регуляторы играют важную роль в обеспечении точности отображения времени на аналоговых часах.

Каждый из этих элементов взаимодействует между собой, обеспечивая точное и надежное отображение времени на аналоговых часах. Точность и надежность работы часового механизма зависят от правильной работы и взаимодействия всех его компонентов.

## **1.3 Математические вычисления для отображения времени**

Отображение времени на аналоговых часах требует точного позиционирования часовой, минутной и секундной стрелок в соответствии с текущим временем.

Цель этого раздела состоит в разработке и применении математических формул, которые позволят нам точно определить положение каждой стрелки на циферблате в зависимости от текущего времени. Такие вычисления обеспечат корректное отображение времени на аналоговых часах и поддержание их актуальности в течение дня.

1. Вычисление углов для часовой стрелки:

Для определения угла поворота часовой стрелки используется следующая формула:

**hour\_angle = math.radians((hours % 12) \* 30 + (minutes / 2))**

Эта формула состоит из двух частей. Первая часть ((hours % 12) \* 30) вычисляет угол для часовой стрелки в зависимости от текущего часа. Угол в 360 градусов делится на 12 часов, что дает 30 градусов на каждый час. Вторая часть (minutes / 2) добавляет дополнительное смещение в зависимости от минут, чтобы учесть плавное движение стрелки между часовыми делениями.

2. Вычисление углов для минутной стрелки:

Для определения угла поворота минутной стрелки используется следующая формула:

**minute\_angle = math.radians(minutes \* 6 + (seconds / 10))**

Здесь minutes \* 6 вычисляет угол поворота минутной стрелки в зависимости от текущего количества минут. Так как в часе 60 минут, угол поворота в 360 градусов делится на 60 минут, что дает 6 градусов на каждую минуту. Дополнительное смещение (seconds / 10) учитывает плавное движение стрелки в течение каждой минуты.

3. Вычисление углов для секундной стрелки:

Для определения угла поворота секундной стрелки используется следующая формула:

**second\_angle = math.radians(seconds \* 6)**

Здесь seconds \* 6 вычисляет угол поворота секундной стрелки в зависимости от текущего количества секунд. Поскольку в минуте 60 секунд, угол поворота в 360 градусов делится на 60 секунд, что дает 6 градусов на каждую секунду.

# **ГЛАВА 2. РЕАЛИЗАЦИЯ ФУНКЦИОНАЛЬНОСТИ ЧАСОВ НА PYTHON**

## **2.1 UML-диаграммы и их роль в проекте:**

В этом разделе представлены различные типы UML-диаграмм, которые используются для моделирования различных аспектов проекта "Написание программы, которая на поверхность формы выводит изображение идущих часов с часовой, минутной и секундной стрелками.".

Диаграмма классов:

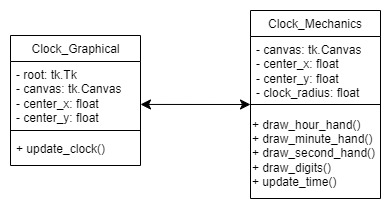


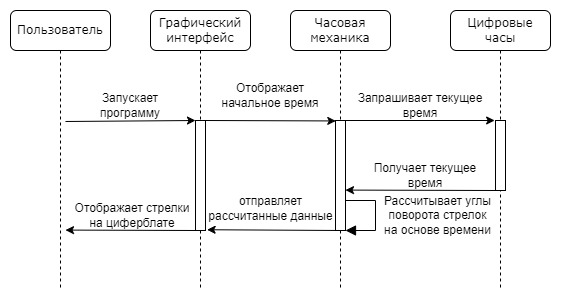
Диаграмма классов — это визуализация структуры классов и их взаимосвязей в проекте. В нашем случае, она позволяет увидеть классы Clock\_Graphical и Clock\_Mechanics, их атрибуты и методы, а также взаимосвязи между ними.

Clock\_Graphical — это класс, который отвечает за графический интерфейс приложения. У него есть атрибуты root (корневой элемент tkinter) и canvas (холст для отображения часов), а также метод update\_clock(), который обновляет время на часах.

Clock\_Mechanics — это класс, который представляет часы. У него есть атрибуты canvas, center\_x, center\_y и clock\_radius, определяющие параметры часов, а также методы для рисования стрелок и цифр (draw\_hour\_hand(), draw\_minute\_hand(), draw\_second\_hand(), draw\_digits()) и обновления времени (update\_time()).

Эта диаграмма помогает понять организацию кода, взаимосвязи между классами и их роли в проекте.

Диаграмма последовательности:

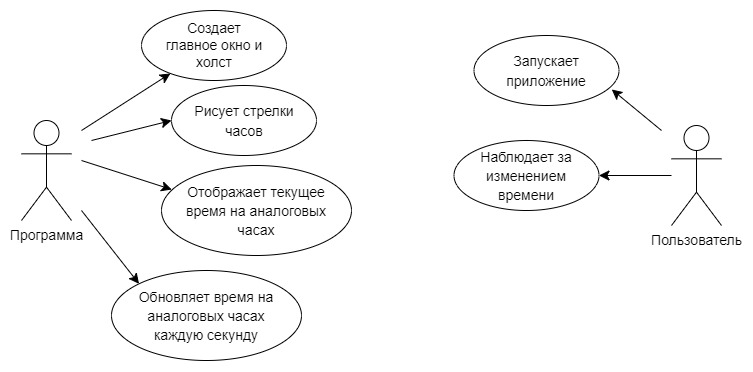


Эта диаграмма представляет последовательность действий и взаимодействия между пользователем и различными компонентами программы.

1. Пользователь: Запускает программу.
2. Графический интерфейс: Начинает процесс с отображения начального времени на экране.
3. Часовая механика: Запрашивает текущее время у цифровых часов.
4. Цифровые часы: Получает текущее время и предоставляет его часовой механике.
5. Часовая механика: Рассчитывает углы поворота стрелок на основе полученного времени.
6. Часовая механика: Отправляет рассчитанные данные графическому интерфейсу.
7. Графический интерфейс: Отображает стрелки на циферблате.

Эта диаграмма помогает понять взаимодействие между различными компонентами программы и последовательность действий, необходимых для отображения времени и отображения его на графическом интерфейсе.

Диаграмма прецедентов:



Акторы:

Программа: Отвечает за создание главного окна и холста, рисование стрелок часов, отображение текущего времени на аналоговых часах и обновление времени на них.

Пользователь: Запускает приложение и наблюдает за изменением времени.

Прецеденты:

Создание главного окна и холста: Программа создает главное окно и холст для отображения аналоговых часов.

Рисование стрелок часов: Программа рисует на холсте стрелки, соответствующие текущему времени.

Отображение текущего времени: Программа отображает на аналоговых часах текущее время.

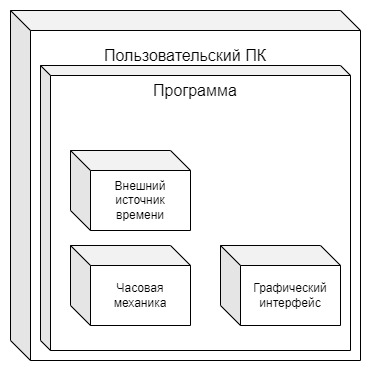
Обновление времени: Программа периодически обновляет время на аналоговых часах.

Запуск приложения: Пользователь запускает приложение "Аналоговые часы".

Наблюдение за изменением времени: Пользователь наблюдает за изменением времени на аналоговых часах.

Эта диаграмма помогает понять основные действия, которые выполняются программой и пользователем при использовании приложения "Аналоговые часы".

Диаграмма развёртывания:



На диаграмме приведены компоненты программы и их размещение на пользовательском ПК.

Пользовательский ПК: Компьютер пользователя, на котором запускается программа "Аналоговые часы".

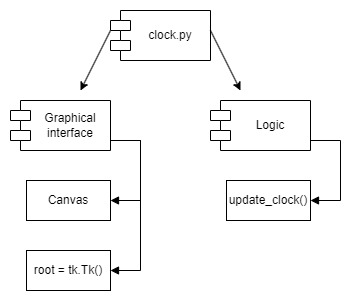
Графический интерфейс: Компонент, отвечающий за визуальное представление аналоговых часов на экране.

Часовая механика: Компонент, реализующий логику работы часов, включая расчет углов поворота стрелок и обновление времени.

Внешний источник времени: Компонент, который предоставляет актуальное время для отображения на аналоговых часах. Обычно это системное время операционной системы ПК.

Эта диаграмма позволяет понять, какие компоненты входят в систему и где они размещаются на пользовательском ПК в процессе развертывания приложения.

Диаграмма компонентов:



Компоненты:

Программа: Отвечает за создание главного окна и холста, обновление времени на аналоговых часах и взаимодействие с библиотекой Tkinter.

Холст: Предоставляет визуализацию аналоговых часов и инструменты для рисования стрелок и цифрового циферблата.

Логика: Включает в себя функцию update\_clock(), которая вычисляет текущее время, определяет положение стрелок часов и обновляет отображение на холсте.

Взаимодействие:

Создание главного окна и холста: Программа создает главное окно и холст для отображения аналоговых часов.

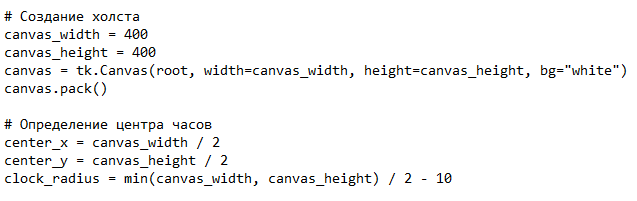
Обновление времени на аналоговых часах: Программа периодически обновляет отображение времени на аналоговых часах с учетом текущего времени.

Визуализация часов: Холст отображает стрелки и цифры на циферблате, соответствующие текущему времени.

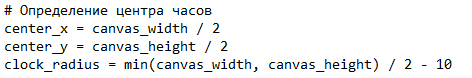
Данная диаграмма компонентов помогает понять взаимосвязь между различными компонентами программы, их функциональное назначение и взаимодействие друг с другом.

## **2.2 Разработка графического интерфейса и часовой механики**

Для начала, я создал холст с помощью библиотеки tkinter. Этот холст послужил базой для отображения аналоговых часов, предоставляя возможность нарисовать на нем стрелки и цифры.

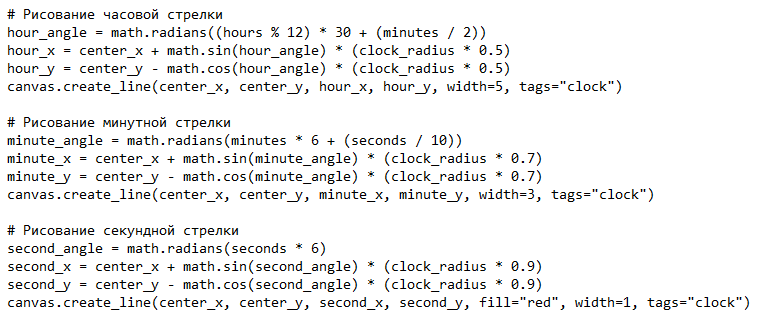


Для определения центра часов я использовал следующие вычисления:

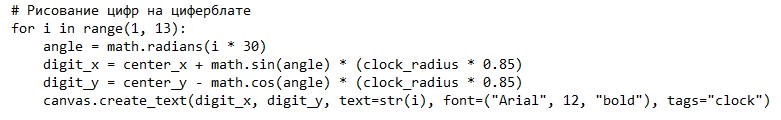


Эти строки кода помогли определить координаты центра часов на холсте, что позволило расположить стрелки и цифры в центре экрана, обеспечивая симметричное и аккуратное отображение аналоговых часов.

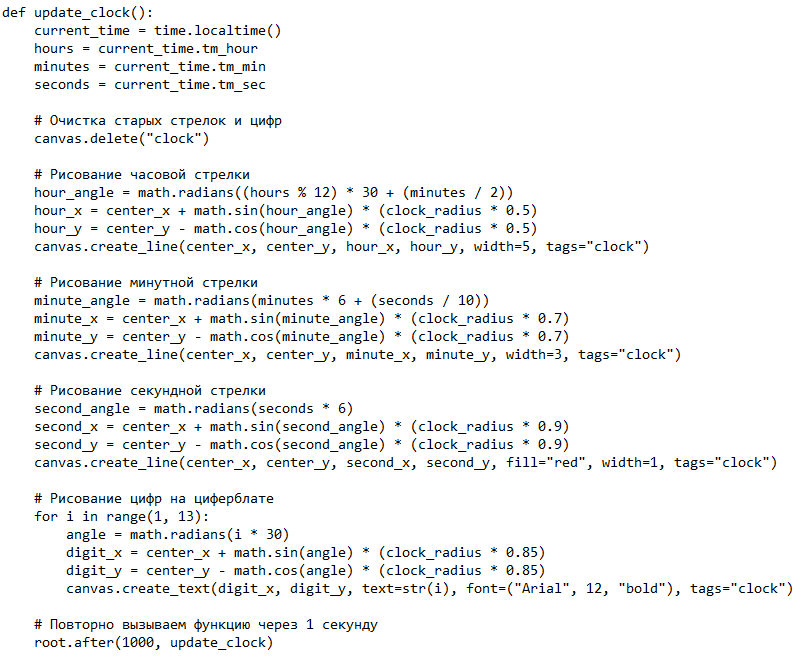
Далее я приступил к определению расположения стрелок на циферблате. Для этого я использовал математические вычисления, которые позволили мне точно определить координаты концов стрелок в зависимости от текущего времени.



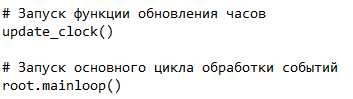
Кроме того, я реализовал отображение цифр на циферблате с помощью текстовых объектов.



Для обновления времени на часах я разработал функцию update\_clock(), которая вызывается периодически с определенным интервалом времени. Эта функция обновляет положение стрелок в соответствии с текущим временем, обеспечивая плавное изменение времени на аналоговом циферблате.



Теперь остаётся лишь вызвать функцию update\_clock() и запустить основной цикл обработки событий , используя метод root.mainloop(). После вызова функции, стрелки часов начинают двигаться и обновляться каждую секунду. А метод root.mainloop() помогает с обработкой всех действий пользователя и отображением самого окна программы.



Так же я составил блок-схему, чтобы лучше понять взаимосвязь различных частей программы и оценить ее общую логику.



# **ЗАКЛЮЧЕНИЕ**

Завершение проекта "Написание программы, которая на поверхность формы выводит изображение идущих часов с часовой, минутной и секундной стрелками." открывает множество перспективных возможностей для его применения на практике. Созданные часы не только отличаются стильным дизайном и удобством использования, но и обладают потенциалом для решения различных задач:

Во-первых, данное приложение может быть использовано как декоративный элемент в интерьере офиса, дома или других помещений, добавляя им уют и элегантность.

Во-вторых, оно может быть применено в образовательных целях для обучения основам работы с графическими интерфейсами и программирования на языке Python.

Также, аналоговые часы могут быть полезны в офисной среде для управления рабочим процессом и организации рабочего времени.

И, наконец, созданный проект может послужить вдохновением для разработки других графических приложений, адаптируя его функциональность и дизайн под различные потребности и задачи.

Возможности применения аналоговых часов на Python с использованием библиотеки tkinter ограничены только вашей фантазией и потребностями. Это универсальное приложение, которое может быть адаптировано под различные сферы деятельности, принося пользу и удовлетворение пользователям.

# **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. How Do Analog Clocks Work? // Официальный сайт Sciencing.com – URL: https://sciencing.com/analog-clocks-work-4912745.html (Дата обращения: 21.04.2024).

2. Документация по библиотеке Tkinter // Официальный сайт Docs.python.org – URL: https://docs.python.org/3/library/tkinter.html (Дата обращения: 22.04.2024).

3. Руководство по модулю time в Python // Официальный сайт Docs.python.org – URL: https://docs.python.org/3/library/time.html (Дата обращения: 22.04.2024).

4. Часовой механизм: как работает и какие части содержит? // Rlocman.ru. – URL: https://www.rlocman.ru/press-rel/rel.html?di=174 (Дата обращения: 21.04.2024).

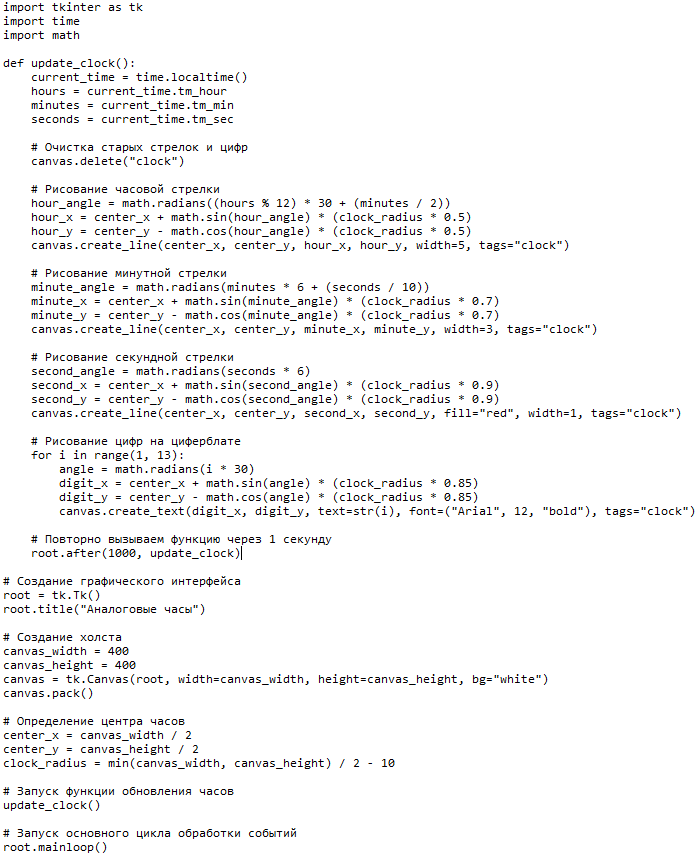
6. Лёгкий способ решать задачи о стрелках часов // Habr.com – URL: https://habr.com/ru/articles/676822/ (Дата обращения: 20.04.2024).

7. Что находится между идеей и кодом? Обзор 14 диаграмм UML // Habr.com – URL: https://habr.com/ru/articles/508710/ (Дата обращения: 23.04.2024).

8. Основы применения UML. Кто и как его использует // Systems.education. – URL: https://systems.education/who-uses-uml (Дата обращения: 23.04.2024).

# **ПРИЛОЖЕНИЕ 1**

**Листинг программы “Аналоговые часы”**



# **ПРИЛОЖЕНИЕ 2**

**Программа “Аналоговые часы”**

