

# БУДИЛЬНИК

## ЦЕЛЬ

### 01 Создать программу "Будильник", которая будет:

- ✦ Позволять пользователю установить время будильника
- ✦ Опрашивать пользователя о его предпочтениях для будильника (например, звук)
- ✦ Определять, сработал ли будильник в заданное время
- ✦ При срабатывании будильника выводить сообщение и проигрывать звук или выводить текстовое сообщение

### 02 Реализовывать индивидуальные предпочтения. Например, будильник по дням недели или специальное сообщение

- ✦ Вам представлен выбор из заданий 2-х уровней сложности. Чем выше уровень сложности, тем более функциональное приложение получится
- ✦ Вам обязательно нужно выполнить все задания **базового уровня сложности** и опционально задания повышенного уровня сложности

## ЧТО НУЖНО СДЕЛАТЬ

### БАЗОВЫЙ УРОВЕНЬ СЛОЖНОСТИ

#### 01 Ввод данных

Пользователь должен ввести время для будильника в формате "ЧЧ:ММ". Также у него есть возможность выбрать звуковое уведомление или текстовое сообщение

#### 02 Цикл работы программы

Программа должна постоянно проверять текущее время и сравнивать его с установленным временем будильника. Если время совпало, будильник срабатывает

### 03 Вывод данных

При срабатывании будильника программа должна вывести сообщение "Время вставать!" или проиграть звук

### 04 Дополнительные возможности

Возможность установить несколько будильников (для этого использовать список/словарь для хранения времени будильников). Будильник должен работать до тех пор, пока не будет остановлен пользователем

## ПОВЫШЕННЫЙ УРОВЕНЬ СЛОЖНОСТИ

- 01 Реализовать возможность задания будильников на несколько дней, включая день недели
- 02 Добавить возможность выбора звука будильника из предустановленных (например, использование файлов звуков в формате .mp3 или .wav)
- 03 Сделать возможность настройки повтора будильника (например, через каждые 5 минут)

## РЕКОМЕНДАЦИИ ДЛЯ ВЫПОЛНЕНИЯ

- ✦ Используйте модуль `time` для получения текущего времени и опроса состояния будильника
- ✦ Применяйте условные операторы для сравнения введенного времени с текущим временем
- ✦ Для звукового уведомления используйте библиотеку `pygame` или стандартный модуль `winsound` (в зависимости от операционной системы)
- ✦ Используйте циклы и задержки для того, чтобы программа не перегружала процессор в постоянном ожидании

# АЛГОРИТМ ВЫПОЛНЕНИЯ ЗАДАЧИ

- 01 Разработать программу с использованием функции, которая будет:
  - ✦ Запрашивать у пользователя время будильника
  - ✦ Постоянно проверять текущее время
  - ✦ Сравнивать его с временем будильника
  - ✦ Если времена совпали, выводить сообщение или проигрывать звук
- 02 Реализовать структуру хранения нескольких будильников (например, через список или словарь)
- 03 Добавить обработку ошибок (например, неправильный формат времени)
- 04 Включить в программу возможности для дополнительных настроек (выбор звука, повторение)

## Вспомогательные материалы к домашнему заданию

Модуль time в Python, реальное время ОС ⚡

Модуль time ⚡

Pygame для начинающих программистов ⚡

## РЕЗУЛЬТАТ

ссылка на код приложения "Копилка" на Python в Github

# КРИТЕРИИ ОЦЕНИВАНИЯ

**К1** Корректная реализация основного функционала будильника (вывод сообщения, опрос времени)

19 баллов

**К2** Реализован цикл, который отслеживает время и срабатывает по установленному времени

3 балла

**К3** Возможность установки нескольких будильников с корректной проверкой времени

3 балла

**К4** Обработка ошибок ввода времени (неправильный формат) и корректное завершение программы

4 баллов

**К5** Выполнения задания повышенной сложности (звуковой файл, возможность повторного будильника, день недели)

6 баллов

35 баллов

**Максимальное  
количество баллов**

19 баллов

**Минимальное количество баллов, чтобы  
преподаватель смог зачесть вашу работу**