Simple Alarm Clock:

Input: Set the alarm time and optional message.
Output: Alarm triggered at the set time with a notification.

Example:

- Input: "Set alarm for 7:00 AM"
- Output: "Alarm set for 7:00 AM."
  At 7:00 AM: "Wake up! It's 7:00 AM."

Solution 1: Simple Alarm Clock using Timer and TimerTask

Code:

```
import java.util.Timer;
import java.util.TimerTask;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

public class SimpleAlarmClock {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Input time for the alarm
        System.out.println("Set alarm time in format HH:mm (24-hour format)
        String alarmTime = scanner.nextLine();

        // Input optional alarm message
        System.out.println("Enter an optional alarm message (or leave blank
        String message = scanner.nextLine();
        if (message.isEmpty()) {
            message = "Wake up!";
        }

        // Schedule the alarm
        scheduleAlarm(alarmTime, message);
        scanner.close();
    }
```

```
        // Method to schedule alarm using Timer
        public static void scheduleAlarm(String alarmTime, String message) {
            Timer timer = new Timer();

            // TimerTask to trigger alarm at set time
            TimerTask task = new TimerTask() {
                public void run() {
                    System.out.println(message);
                    timer.cancel(); // Stop the timer after the alarm
                }
            };

            try {
                // Parse the user input time
                SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm");
                Date alarmDate = dateFormat.parse(alarmTime);

                System.out.println("Alarm set for " + alarmTime);

                // Schedule task at the specified alarm time
                timer.schedule(task, alarmDate);
            } catch (Exception e) {
                System.out.println("Invalid time format. Please use HH:mm.");
            }
        }
    }
```

Output:

```
Set alarm time in format HH:mm (24-hour format):
 15:53
Enter an optional alarm message (or leave blank):

Alarm set for 15:53
Wake up!
```

Explanation :

- Imports Timer and TimerTask for scheduling tasks.
- Takes user input for alarm time (in 24-hour format) and an optional message.

- Parses the input time using SimpleDateFormat and sets a TimerTask to print the message at the specified time.
- Once the task is triggered, it cancels the timer to stop further execution.

Solution 2: Simple Alarm Clock using ScheduledExecutorService

Code:

```
import java.time.LocalTime;
import java.time.temporal.ChronoUnit;
import java.util.Scanner;
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.TimeUnit;

public class AlarmClock {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Input time for the alarm
        System.out.println("Set alarm time in format HH:mm (24-hour format)
        String alarmTime = scanner.nextLine();

        // Input optional alarm message
        System.out.println("Enter an optional alarm message (or leave blank
        String message = scanner.nextLine();
        if (message.isEmpty()) {
            message = "Wake up!";
        }

        // Schedule the alarm
        scheduleAlarm(alarmTime, message);
        scanner.close();
    }

    // Method to schedule alarm using ScheduledExecutorService
    public static void scheduleAlarm(String alarmTime, String message) {
        ScheduledExecutorService scheduler = Executors.newScheduledThreadPo

        try {
```

```
                // Parse input time to LocalTime
                LocalTime alarmLocalTime = LocalTime.parse(alarmTime);
                LocalTime currentTime = LocalTime.now();

                // Calculate delay in seconds
                long delay = ChronoUnit.SECONDS.between(currentTime, alarmLocal
                if (delay < 0) {
                    System.out.println("Alarm time is in the past. Please set a
                    return;
                }

                System.out.println("Alarm set for " + alarmTime);

                // Schedule the alarm task
                scheduler.schedule(() -> {
                    System.out.println(message);
                    scheduler.shutdown(); // Shut down the scheduler after the
                }, delay, TimeUnit.SECONDS);
            } catch (Exception e) {
                System.out.println("Invalid time format. Please use HH:mm.");
            }
        }
    }
```

Output:

```
Set alarm time in format HH:mm (24-hour format):
 15:45
Enter an optional alarm message (or leave blank):
 Good Morning!
Alarm set for 15:45
```

Explanation:

- Uses ScheduledExecutorService for more robust task scheduling.
- Takes user input for alarm time and optional message.
- Converts the alarm time to LocalTime, calculates the delay between the current time and alarm time.
- Schedules the alarm task to run after the calculated delay, using TimeUnit.SECONDS.

Java Code Editor: