DIGITAL CLOCK

A PROJECT REPORT

Submitted by

ASHMA BANU M (2303811710422014)

in partial fulfillment of requirements for the award of the course

CGB1201 - JAVA PROGRAMMING

In

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

NOVEMBER- 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **" DIGITAL CLOCK"** is the bonafide work of **ASHMA BANU M (2303811710422014)** who carriedout the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE

Mrs.A.Delphin Carolina Rani, M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

SIGNATURE

Mr. M. Saravanan, M.E.,

SUPERVISOR

ASSISTANTP ROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on 02.12.2024

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **"DIGITAL CLOCK"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201- JAVA PROGRAMMING.**

Signature

ASHMA BANU M

Place: Samayapuram

Date: 02.12.2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr.S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr A. DELPHIN CAROLINA RANI, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project guide **Mr.M. SARAVANAN, M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING** ,for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards

MISSION OF THE INSTITUTION

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- Be an institute with world class research facilities
- Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

MISSION OF DEPARTMENT

M1: Industry Specific: To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

M2: Research: To prepare students for research-oriented activities.

M3: Society: To empower students with the required skills to solve complex technological problems of society.

PROGRAM EDUCATIONAL OBJECTIVES

1. PEO1: Domain Knowledge

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

2. PEO2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

3. PEO3: Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: Domain Knowledge

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

PSO 2: Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications.

PSO 3: Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The Java Digital Clock program is a graphical user interface (GUI) application designed to display the current time in real-time, updating every second. Developed using the Java Swing framework, the program demonstrates key programming concepts such as event-driven programming, multithreading, and time manipulation. The clock is implemented using a Timer to ensure smooth and periodic updates without blocking the main thread. The time is formatted using the SimpleDateFormat class, providing an intuitive and user-friendly display. This lightweight and responsive application serves as a practical example of real-time Java programming and can be further enhanced with features like date display, custom themes, or alarm functionality.The Java Digital Clock program effectively showcases the use of Java's graphical capabilities to create a dynamic and real-time application. By employing Swing components like JFrame and JLabel, the program provides a clear and aesthetically pleasing interface to display the current time in the format HH:mm:ss.This project highlights the practical application of Java's GUI tools and serves as a foundation for further exploration into advanced features such as time zone support, date display, and user customization. It is a versatile and educational example for both beginner and intermediate Java developers.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
The Digital Clock project is a Java-based application designed to display the current system time in real-time, offering an engaging way to explore Java's core programming concepts. The project utilizes essential Java features such as Object-Oriented Programming (OOP) principles, multithreading**,** GUI development, and event handling to create an accurate and user-friendly clock.	**PO1 -3** **PO2 -3** **PO3 -3** **PO4 -3** **PO5 -3** **PO6 -3** **PO7 -3** **PO8 -3** **PO9 -3** **PO10 -3** **PO11-3** **PO12 -3**	**PSO1 -3** **PSO2 -3** **PSO3 -3**

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Objective

The primary objective of this Java program is to design and implement a functional digital clock that dynamically displays the current time in the format HH:mm:ss and updates in real-time every second. The program demonstrates the use of core Java features, including Swing GUI for creating a user-friendly graphical interface, multithreading for ensuring smooth and continuous updates without freezing, and time management using SimpleDateFormat and Date for formatting and fetching the current time. It incorporates event handling to manage periodic updates efficiently and provides an alternative console-based implementation for environments without graphical support. This program serves as a practical example of integrating essential Java concepts such as GUI development, threading, and event-driven programming, making it both functional and educational for developers learning Java fundamentals.

1.2 Overview

The digital clock program provides a simple yet practical implementation of a clock that displays the current time in real-time using Java. It offers both a Graphical User Interface (GUI) version, created with the Swing framework, and a console-based version for environments without GUI support. The GUI version features a user-friendly window displaying the time in HH:mm:ss format, updated every second using the Timer class and event handling mechanisms. The console-based version achieves the same functionality by continuously updating the time in the terminal using Thread.sleep() for periodic delays. This program integrates key Java concepts, including time formatting with SimpleDateFormat, multithreading for real-time updates, and exception handling to manage potential interruptions. It serves as an excellent demonstration of Java's versatility in creating dynamic, real-time applications, highlighting its ability to manage graphical components, time operations, and system resources effectively.

1.3 Java Programming Concepts

The basic concepts of Object-Oriented Programming (OOP) are:

- **Class and Object:** A class is a blueprint, and an object is an instance of the class.
- **Encapsulation:** Bundles data and methods into a single unit (class) while restricting direct access to data.
- **Inheritance:** Enables a class (child) to inherit properties and methods from another class (parent), promoting code reuse.
- **Polymorphism:** Allows methods to perform differently based on the object context (e.g., method overloading and overriding).
- **Abstraction:** Hides implementation details and exposes only essential features, simplifying system design.

Project related concepts

1. Classes and Objects

- java.util.Date Class : The Date class represents a specific moment in time, with millisecond precision. It encapsulates the current date and time or a date and time set explicitly by the programmer
- java.text.SimpleDateFormat Class : The SimpleDateFormat class is used to format and parse dates and times according to a specific pattern..

2. Polymorphism

- Polymorphism refers to the ability of an object to take many forms, allowing one interface to be used for different types. In the Digital Clock program, polymorphism is evident in the way Java handles event listeners and method overriding.

3. Multithreading

- In the Digital Clock program, multithreading is used to ensure the clock updates every second without freezing or blocking other operations in the application. Multithreading is a fundamental concept in Java that allows multiple tasks (threads) to run concurrently, improving the efficiency and responsiveness of an application.

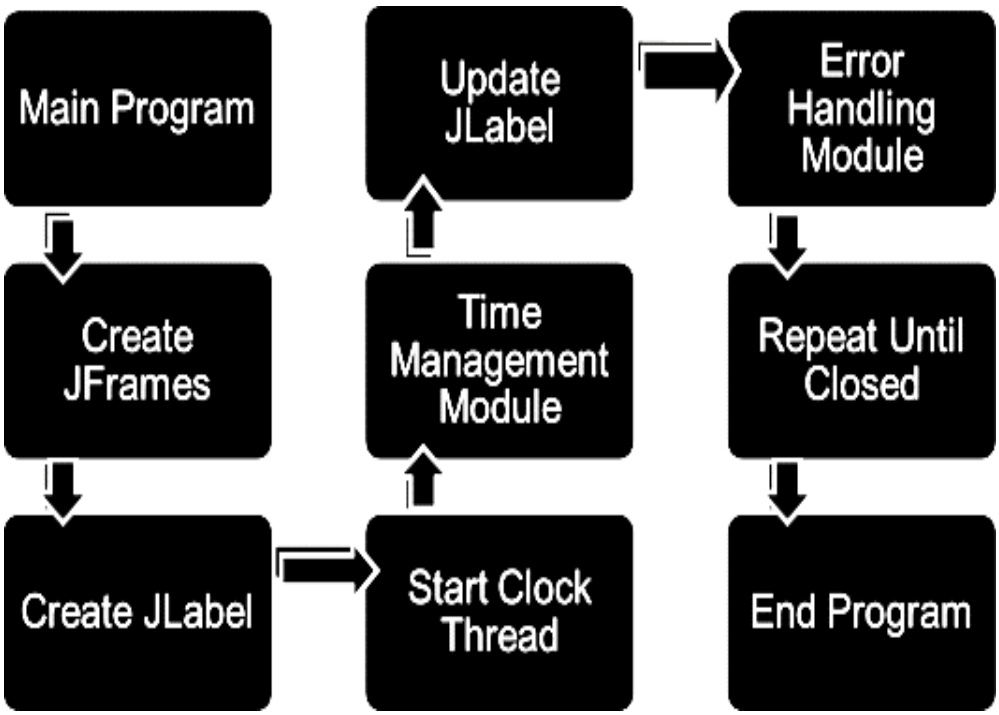
CHAPTER 2

PROJECT METHODOLOGY

2.1 Proposed Work

The proposed work focuses on developing a reliable, efficient, and user-friendly Digital Clock application using Java. This application will display the current system time in HH:mm:ss format, updating in real-time. Two approaches are proposed to cater to different environments: a console-based digital clock and a GUI-based digital clock. The console-based implementation is lightweight, leveraging core Java libraries such as SimpleDateFormat, Date, and Thread to format the time and update it continuously, making it ideal for headless environments like servers or remote systems.

2.2 Block Diagram



CHAPTER 3

MODULE DESCRIPTION

3.1 Threading Module

The Threading Module in the provided program refers to the use of Java's multithreading capabilities to ensure the digital clock updates at regular intervals. In the given console-based program, Thread.sleep() is used within an infinite loop to pause the execution for 1 second, allowing the time to be updated in real-time.

3.2 Time Management Module

The Time Management Module in the provided program handles retrieving, formatting, and displaying the current system time in real-time. It ensures that the digital clock accurately reflects the current time and updates at regular intervals. This module ensures the clock maintains synchronization with the system time and provides a seamless real-time experience for the user

3.3 Error Handling Module

The Error Handling Module in the provided program ensures that the program gracefully manages exceptions, particularly those that might occur during execution, such as interruptions to the thread's sleep state. By incorporating proper exception handling, the program remains robust and user-friendly, even in the presence of runtime anomalies.

3.4 Event Handling Module

The Event Handling Module in the context of the provided digital clock program is used primarily in GUI-based versions (e.g., using Swing or AWT). Event handling allows the program to respond to specific user actions or system-generated events, such as updating the clock display every second or closing the application window.

CHAPTER 4

CONCLUSION & FUTURESCOPE

4.1 CONCLUSION

The Digital Clock project successfully demonstrates the use of core Java concepts to create a simple yet functional real-time clock application. By utilizing key features such as multithreading**,** time management**,** and event handling, the program efficiently updates and displays the current time. The use of Java's SimpleDateFormat class allows for proper time formatting, while the Thread.sleep() method ensures periodic updates in console-based implementations.When extended with a Swing-based GUI**,** the project highlights the flexibility of Java in creating interactive applications. It also demonstrates the importance of handling exceptions and managing events in user interface development, providing a smooth and responsive experience.The project serves as a solid foundation for learning Java's basic concepts like Object-Oriented Programming (OOP)**,** GUI development, and time handling**,** while also offering practical experience with handling real-time updates and user interactions. Future enhancements could include adding features like customizable time formats, alarms, or integrating it with system notifications.

4.2 FUTURE SCOPE

The Digital Clock program has significant potential for future enhancements, allowing it to evolve into a more versatile and feature-rich application. Customizable time formats, such as toggling between 12-hour and 24-hour displays, and the inclusion of date display options can provide users with greater flexibility. Features like alarms and countdown timers can add practical utility, while support for world clocks can make the application useful for tracking time across multiple time zones. Improving the graphical user interface (GUI) with themes, interactive buttons, and dynamic backgrounds can enhance the user experience

REFERENCES

Java Books:

1. **Herbert Schildt**, Java: **The Complete Reference, 11th Edition, McGraw Hill Education:**
    - A comprehensive guide covering all aspects of Java programming, including GUI development, multithreading, and event handling.
2. **Cay S. Horstmann and Gary Cornell**, Core Java Volume I - Fundamentals, 11th Edition, Pearson:
    - Focuses on fundamental Java concepts, including object-oriented programming, Swing, and threading.

Websites:

1. **Oracle Java Documentation :**
    - Official Java tutorials covering a wide range of topics, including GUI programming with Swing, multithreading, and time handling.
2. **GeeksforGeeks:**
    - A comprehensive resource for Java concepts, including practical examples and detailed explanations of multithreading, Swing, and event handling.

YouTube Links:

1. **Java for Beginners - Java Brains**

- URL: https://www.youtube.com/user/koushks
- Offers Java tutorials from the basics to advanced concepts. The channel provides detailed guides on Java programming, including working with objects and classes, which are crucial for building an EPMS

APPENDIX A (SOURCE CODE)

```
import
java.text.SimpleDateFormat;
import java.util.Date;

public class Main {
   public static void main(String[] args) {
      // Infinite loop to update time every second
      while (true) {
         // Get the current time
         String time = new SimpleDateFormat("HH:mm:ss").format(new Date());
         // Clear the console screen and display the time
         System.out.print("\r" + time);
         try {
            // Wait for 1 second before updating
            Thread.sleep(1000);
         } catch (InterruptedException e) {
            e.printStackTrace();
         }
      }
   }
}
```

APPENDIX B (SCREENSHOTS)