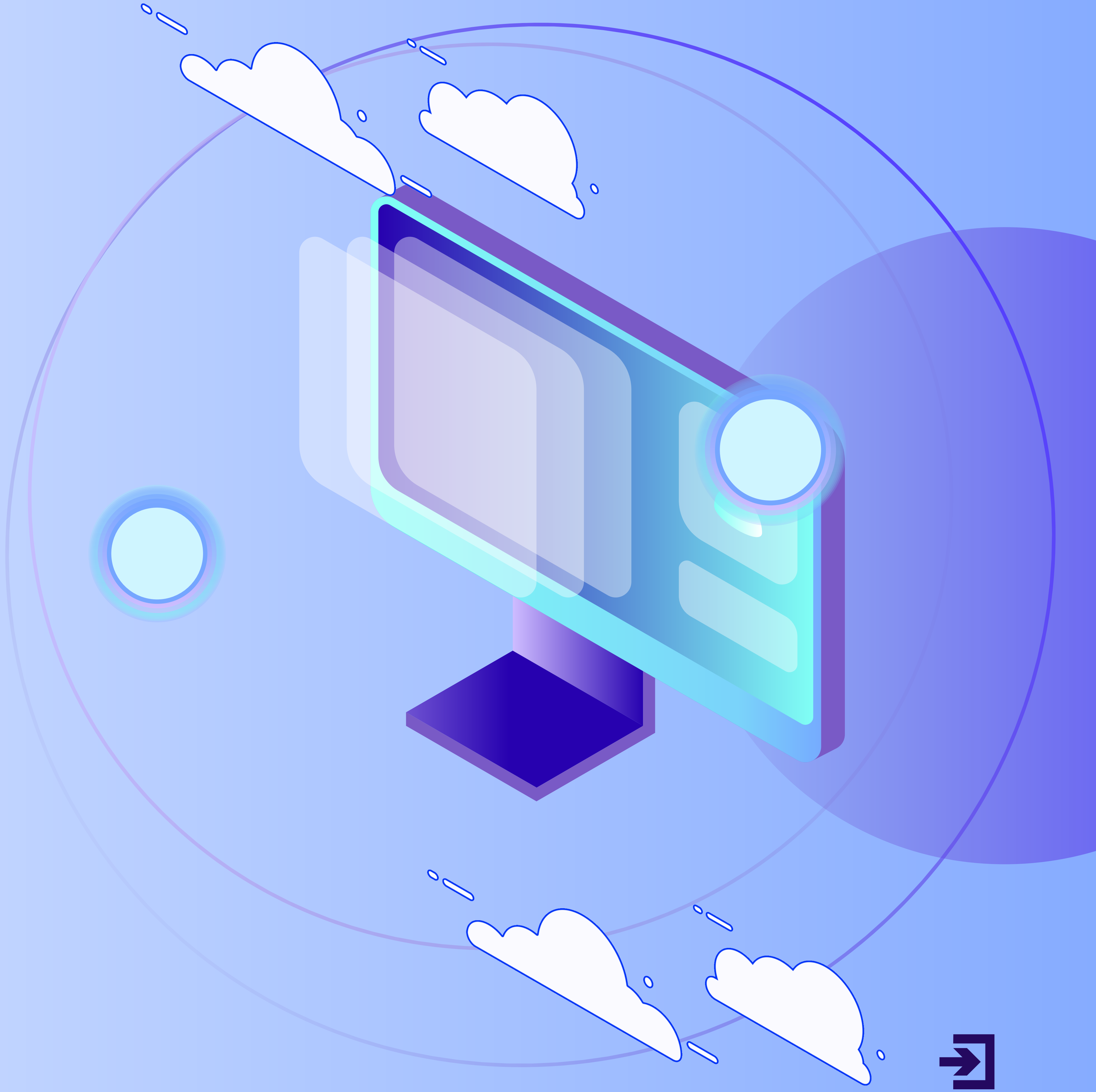
TO-DO LIST APPLICATION IN JAVA

BY USING ARRAYLIST

By Harsh Tiwari



INTRODUCTION TO TO-DO LISTS

What is a To-Do List?

- A To-Do List is a simple tool used to organize and prioritize tasks, helping users manage their daily activities efficiently.

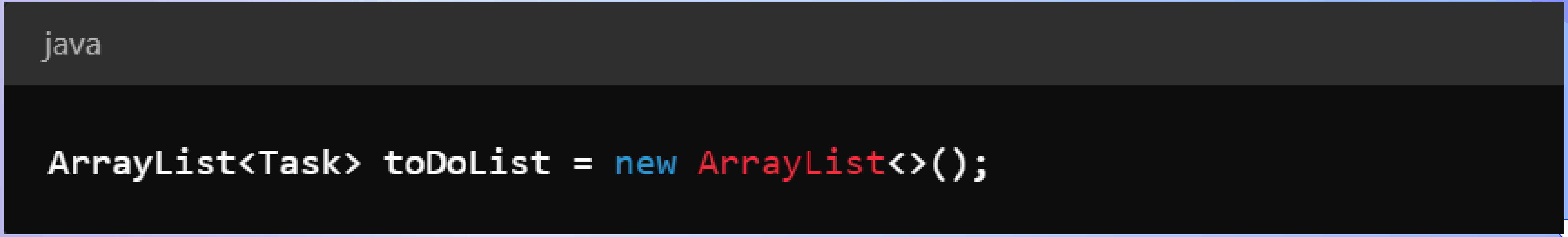
Objective of the Application:

- To create a console-based Java application that allows users to manage their tasks dynamically by adding, removing, viewing, and marking them as completed.

.

DYNAMIC TASK MANAGEMENT WITH ARRAYLIST

- **Dynamic Resizing:** ArrayList automatically adjusts its size when tasks are added or removed, which is crucial for an application where the number of tasks can frequently change.
- **Ease of Use:** Provides built-in methods like add(), remove(), get(), and size() to manage tasks efficiently.
- **Memory Efficiency:** Only consumes memory for the elements it contains, with an option to expand as needed.


java

```java
ArrayList<Task> toDoList = new ArrayList<>();
```

Core Features of the Application

Key Features:

- Add Tasks: Users can add new tasks to their list.
- Remove Tasks: Tasks can be removed when completed or no longer needed.
- View All Tasks: Users can see all current tasks in the list.
- Mark Tasks as Completed: Tasks can be marked as completed, helping users track progress.

TASK CLASS STRUCTURE

Class Structure:

Fields:

- "String description "- A brief description of the task.
- "boolean isCompleted" - A flag indicating whether the task is completed.

Key Methods:

- Constructor: Initializes the task with a description.
- Getters and Setters: Access and modify task properties.
- "markAsCompleted()": Marks the task as completed.

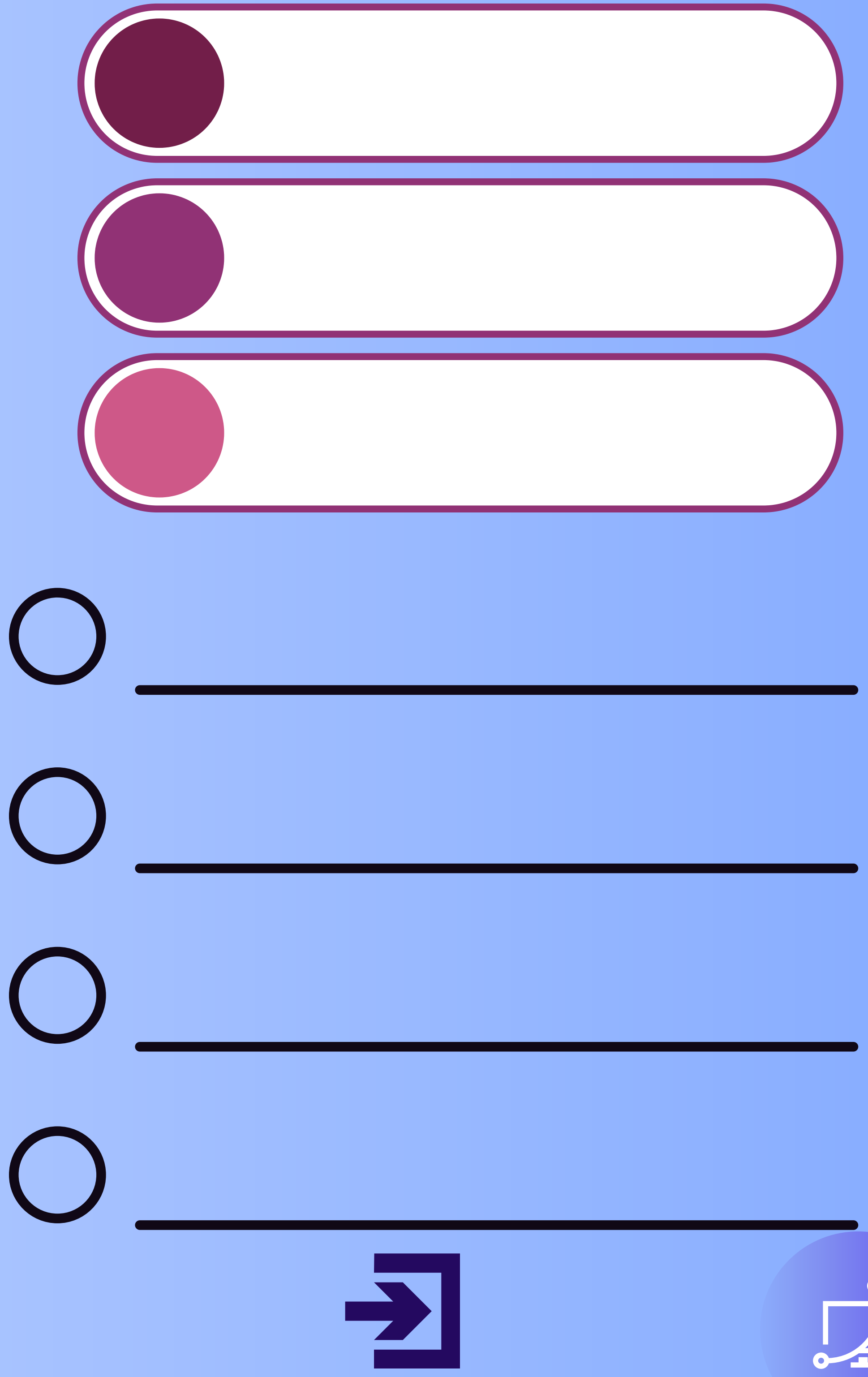
OVERVIEW OF JAVA CODE IMPLEMENTATION

Main Class Overview:

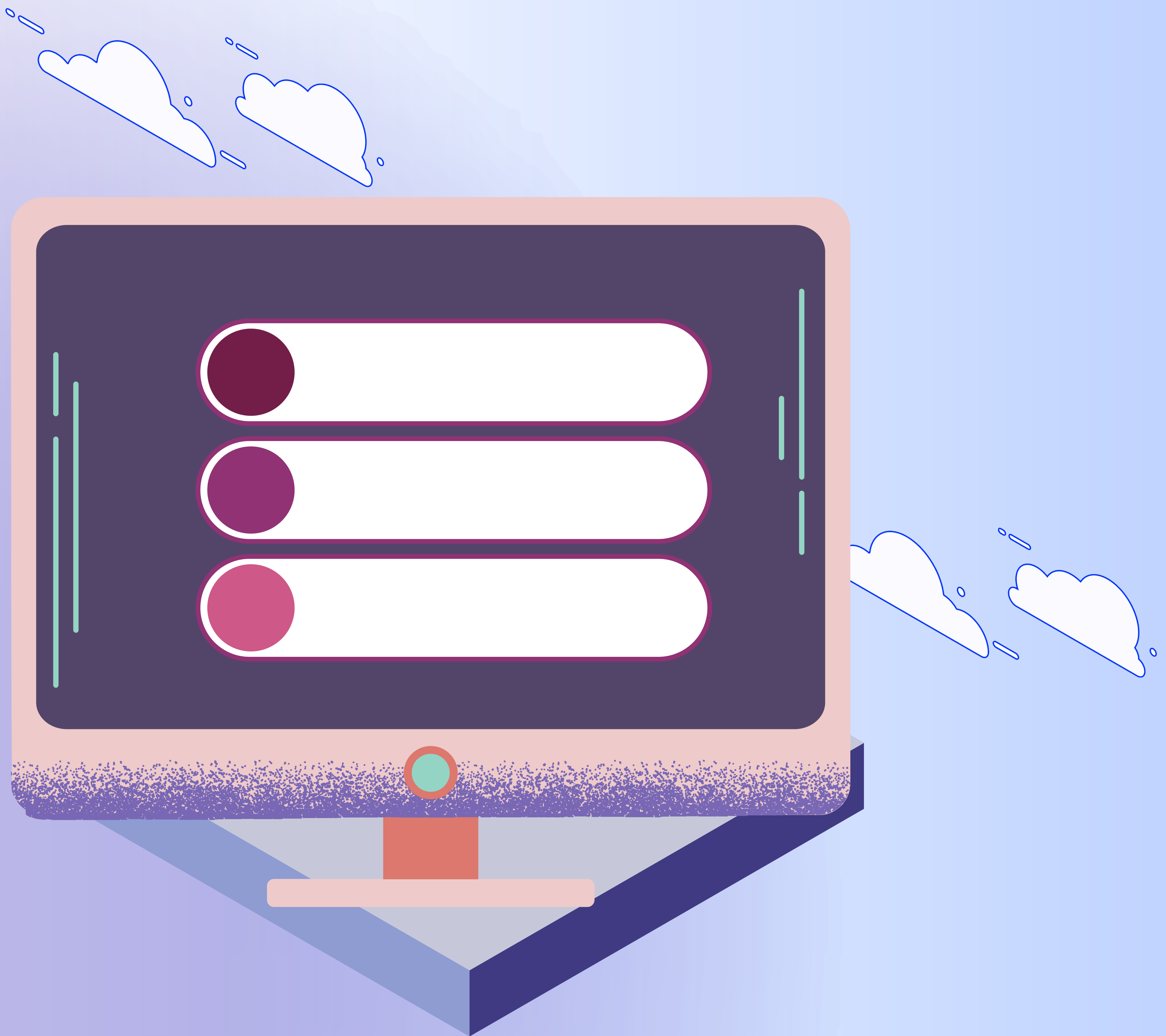
- User Interaction: Uses a loop to provide a menu-driven interface, allowing users to repeatedly choose actions until they exit.
- **Methods in Main Class:**
  - addTask(): Adds a task to the ArrayList.
  - removeTask(): Removes a task from the list based on user input.
  - viewTasks(): Iterates through the ArrayList and displays all tasks.
  - markTaskAsCompleted(): Marks a specified task as completed.

BENEFITS OF THIS APPROACH

- **Advantages of Using Java and ArrayList:**
  - Lightweight and Efficient: Simple console application with minimal resource requirements.
  - Cross-Platform: Java runs on any OS with JVM.
  - Easily Extendable: Code structure allows for future enhancements.
  - Clear and Structured Code: Object-Oriented approach keeps the code modular and maintainable.
- **Learning Opportunities:**
  - Understand Java basics, ArrayList usage, and handling user input/output.
  - Practice creating simple applications with practical use cases.



CONCLUSION



- **Summary of Key Points:**
  - Developed a basic To-Do List application in Java using ArrayList for dynamic task management.
  - Simple, intuitive console-based user interface that covers all basic functionalities.
  - Highlights the importance of using collections and object-oriented programming in real-world applications.
- **Call to Action:**
  - Explore the code, modify it, and experiment with new features.
  - Share feedback or contribute to an open-source version of the application.

THANK YOU!

Harsh Tiwari

twrharsh2064@gmail.com