TOPDOWN APPROACH

Journey to your best program 2024

Topdown Approach

What is topdown approach ?

A method of analysis or decision-making that starts with the general or big picture and then moves to the specific or details.

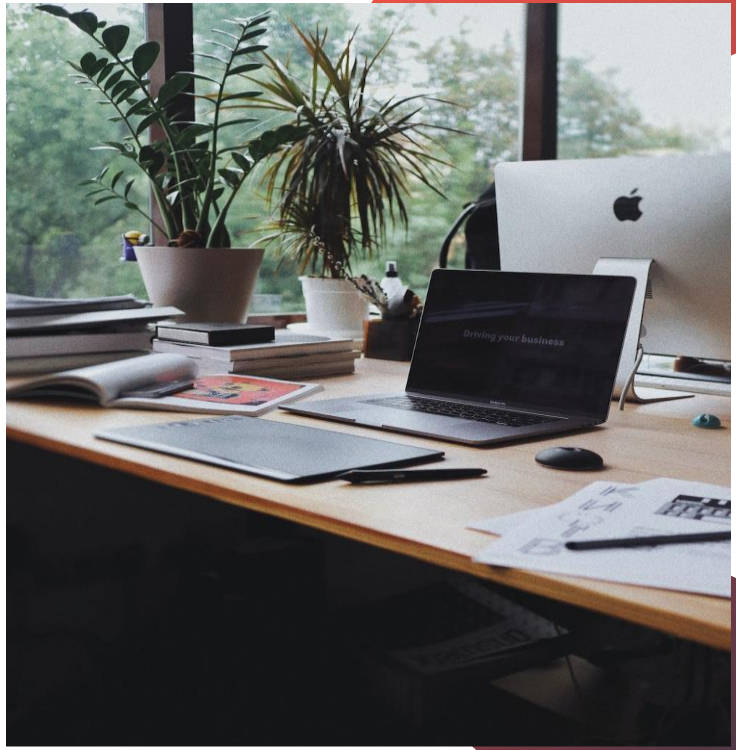
Example

Imagine you have a big, complex project in front of you, like building a house. The top-down approach is like starting at the roof and working your way down, floor by floor, wall by wall, until you reach the foundation. It's a method of problem-solving or decision-making that starts with the big picture and then breaks it down into smaller, more manageable pieces.

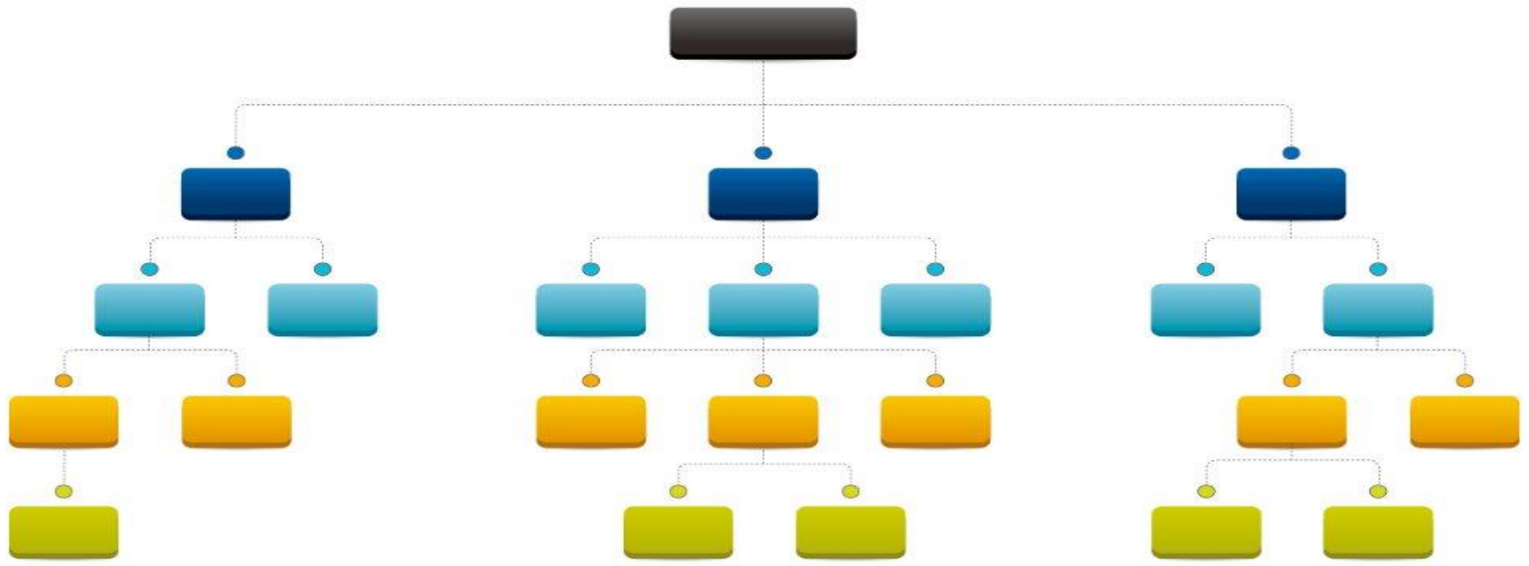
Topdown approach principles

1. Define the Big Picture
2. Breaking Down the Goal
3. Developing Plans for Each Sub-Goal
4. Implementation and Delegation
5. Monitoring and Adjusting

How the Topdown approach works

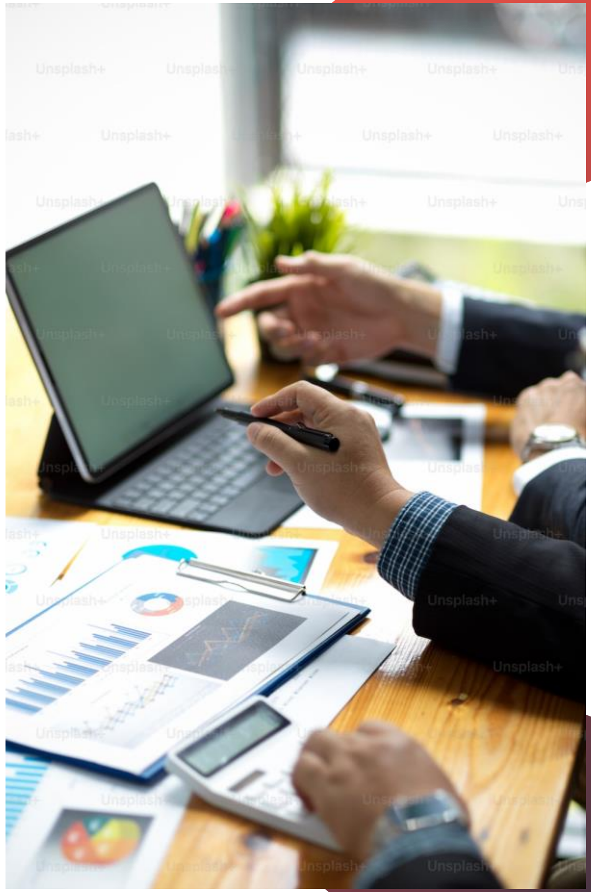


A top-down approach in a hierarchy system refers to the organizational structure where decisions and directives flow from the top-level management down to the lower levels. This hierarchical structure typically consists of different levels or layers, each with its own set of responsibilities and authority.



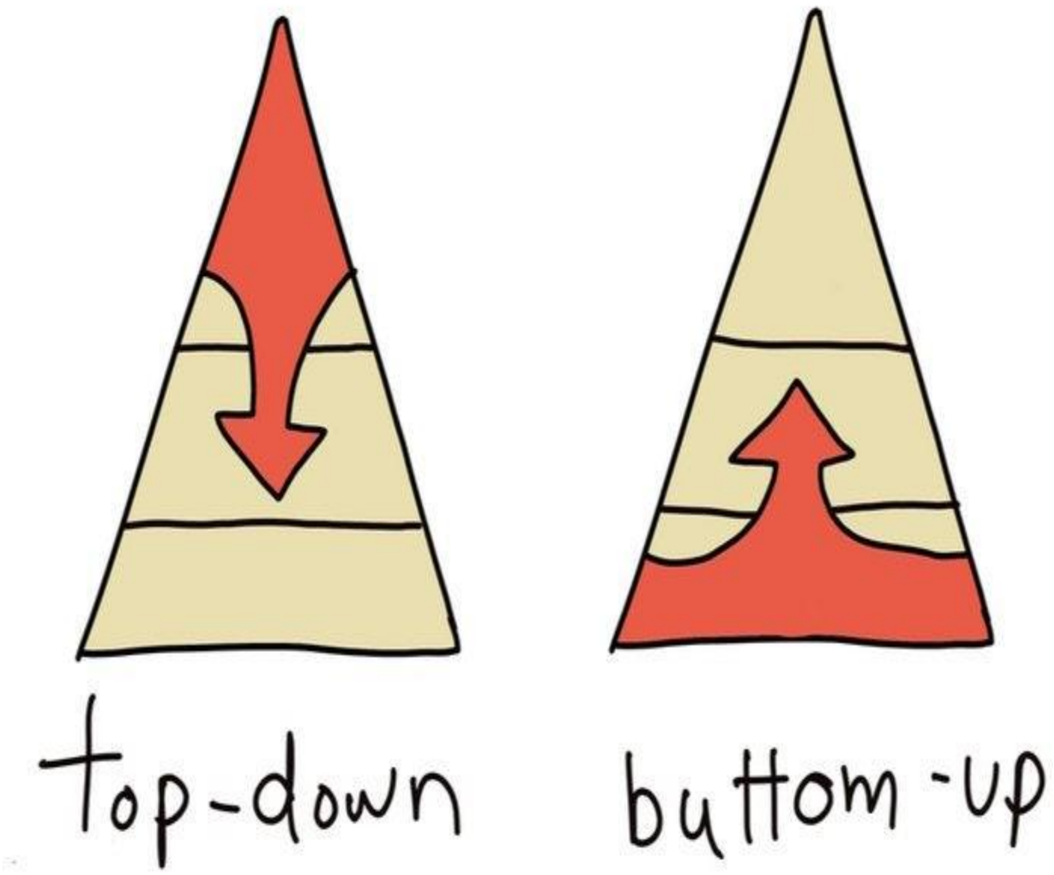
Hierarchy System image

Topdown approach vs Bottom up approach



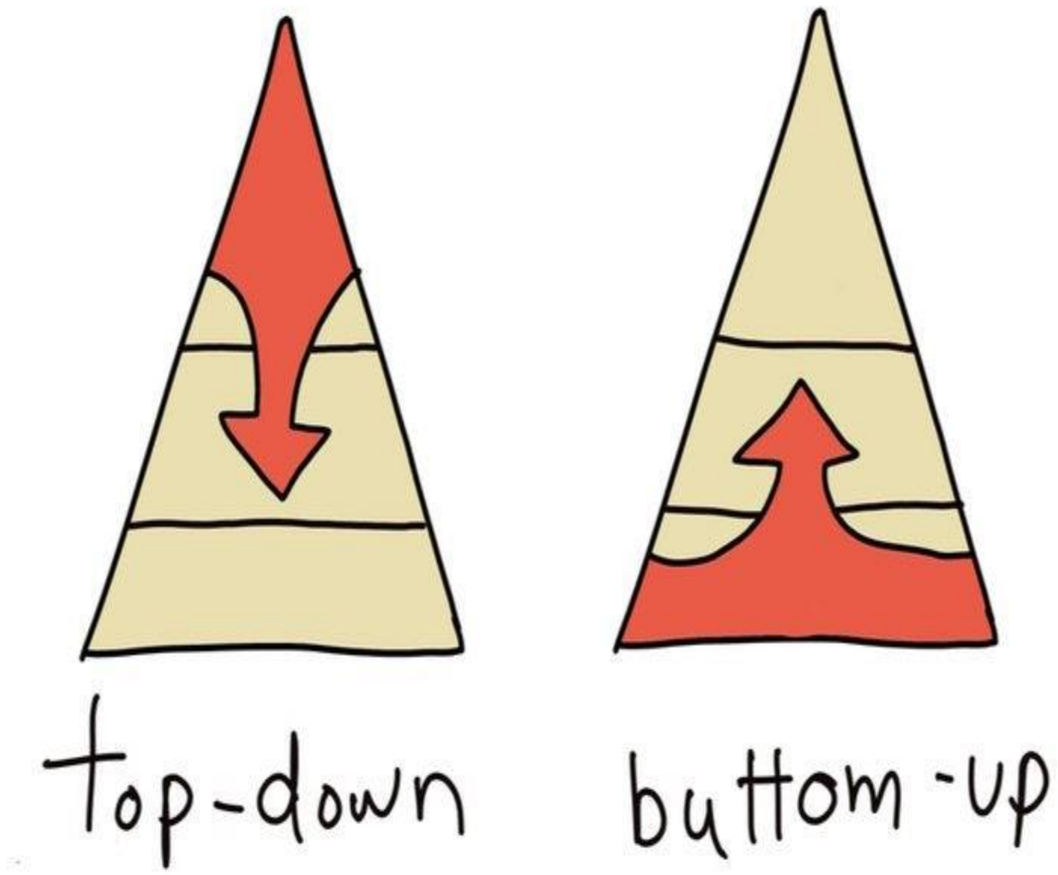
Topdown approach :

- **Start Point:** Begins with a broad, comprehensive system or idea.
- **Decomposition:** The system is broken down into smaller, more manageable parts.
- **Focus:** Emphasizes the final outcome or the overall goal from the beginning.
- **Implementation:** Components are developed in sequence, typically after the overall design or plan is completed.
- **Integration:** Individual components are integrated into the larger system, often after they have been developed.



Bottom up approach :

- **Start Point:** Begins with specific, small-scale components or elements.
- **Aggregation:** These components are developed and then combined to form larger systems or understandings.
- **Focus:** Emphasizes individual components and their functionality before considering the overall system.
- **Implementation:** Components are often developed simultaneously and independently
- **Integration:** The integration of these components gradually reveals the bigger picture or final outcome.



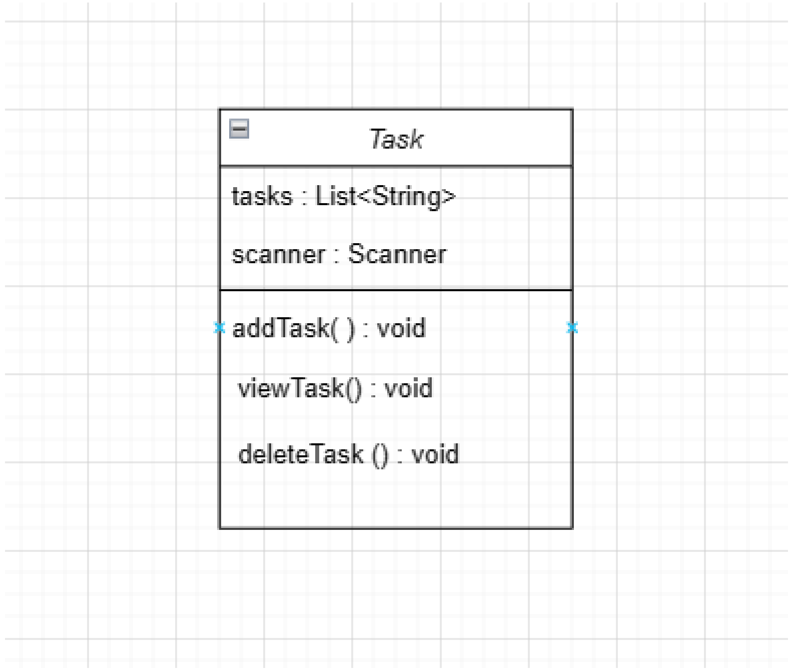
How to apply Topdown approach in programming

```
let dat
let subjectAverage;
query(
  "SELECT * FROM marks WHERE subject_ID=" + subject
  function (datasetsWithSubject) {
    if (datasetsWithSubject.length > 0) {
      subjectAverage = 0;
      datasetsWithSubjectLength = datasetsWithSu
      datasetsWithSubject.forEach((dataset) => {
        subjectAverage += parseFloat(dataset["
      });
      subjectAverage =
        subjectAverage / datas
    } else {
```

Problem

The challenge is to design and implement a to-do list application that incorporates CRUD functionalities (Create, Read, Update, Delete) for managing user tasks. The application should enable users to seamlessly add new tasks, view their existing tasks, update task details when necessary, and efficiently delete tasks. This project requires a solution that ensures a user-friendly and effective task management experience.

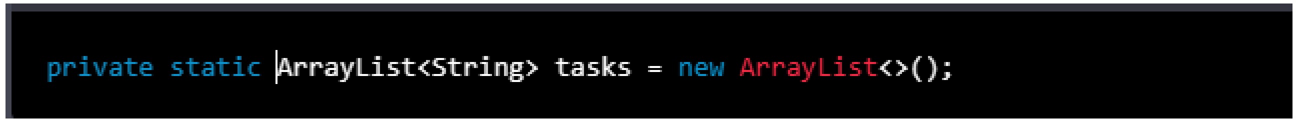
1. Define Overall Functionality



Goal: Create a console-based application to manage a to-do list.

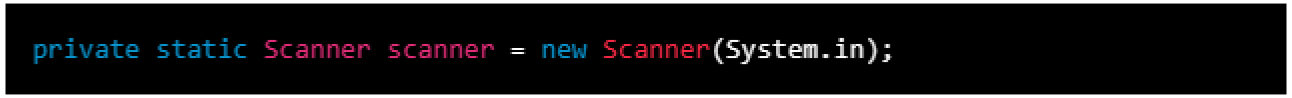
2. Design Major Components

- **Data Structure for Tasks:**



```
private static ArrayList<String> tasks = new ArrayList<>();
```

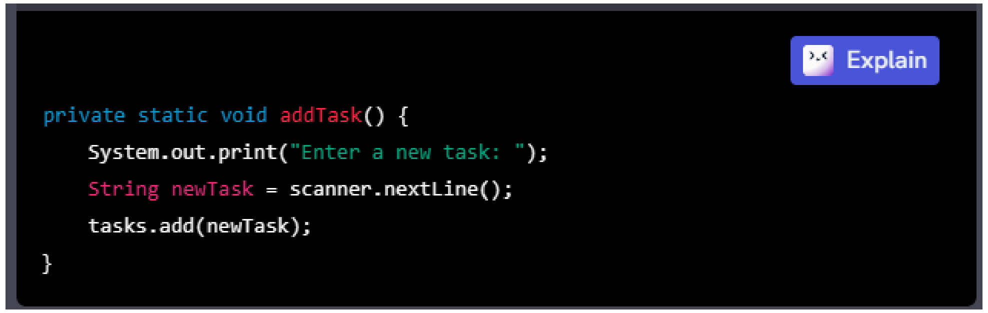
- **Input Handling:**



```
private static Scanner scanner = new Scanner(System.in);
```

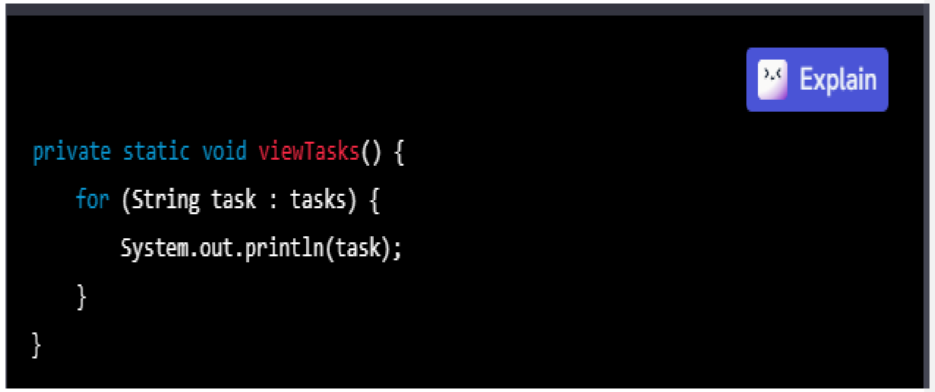
3. Break downComponents

- **Add task :**



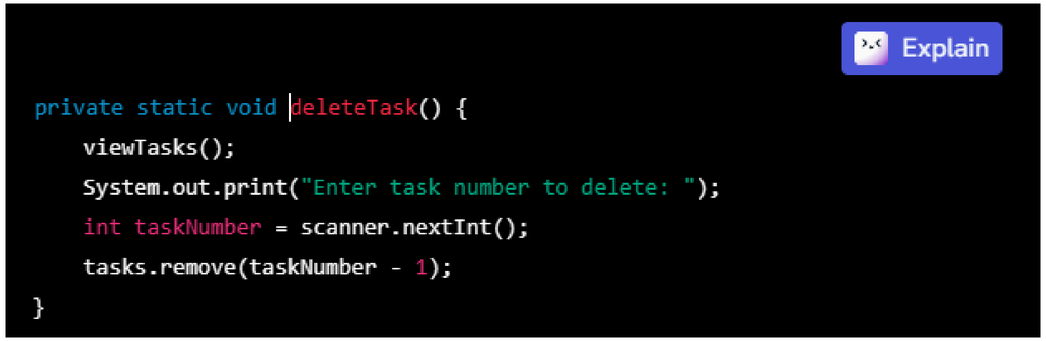
```
private static void addTask() {
    System.out.print("Enter a new task: ");
    String newTask = scanner.nextLine();
    tasks.add(newTask);
}
```

- **View task :**



```
private static void viewTasks() {
    for (String task : tasks) {
        System.out.println(task);
    }
}
```

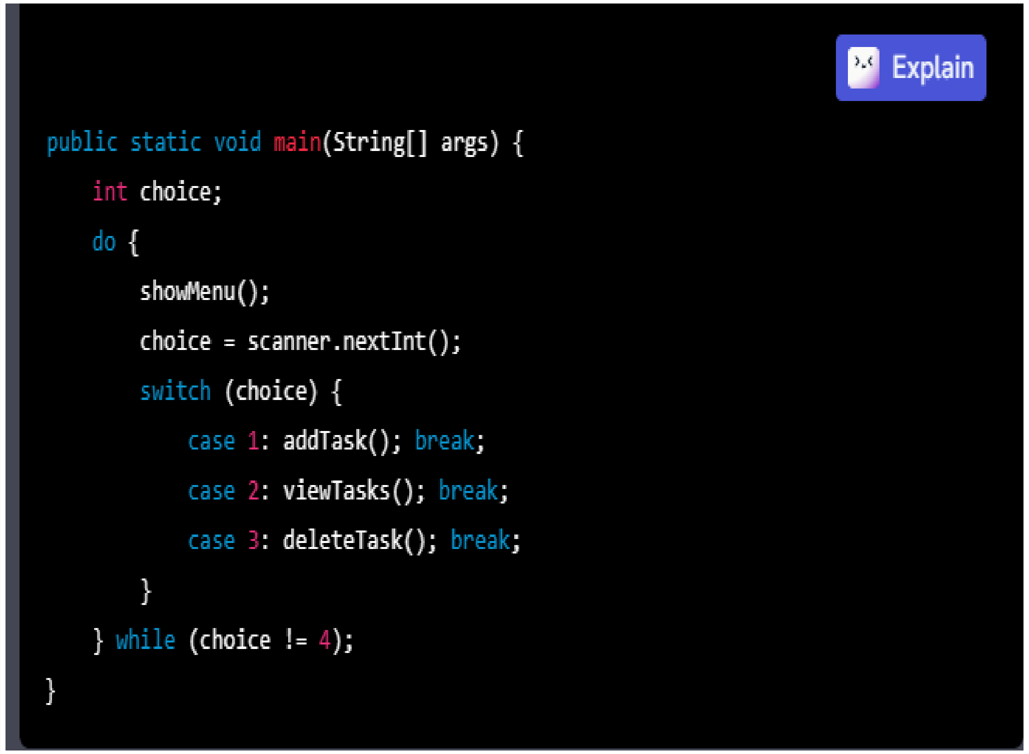
- **Delete task :**



```
private static void deleteTask() {
    viewTasks();
    System.out.print("Enter task number to delete: ");
    int taskNumber = scanner.nextInt();
    tasks.remove(taskNumber - 1);
}
```

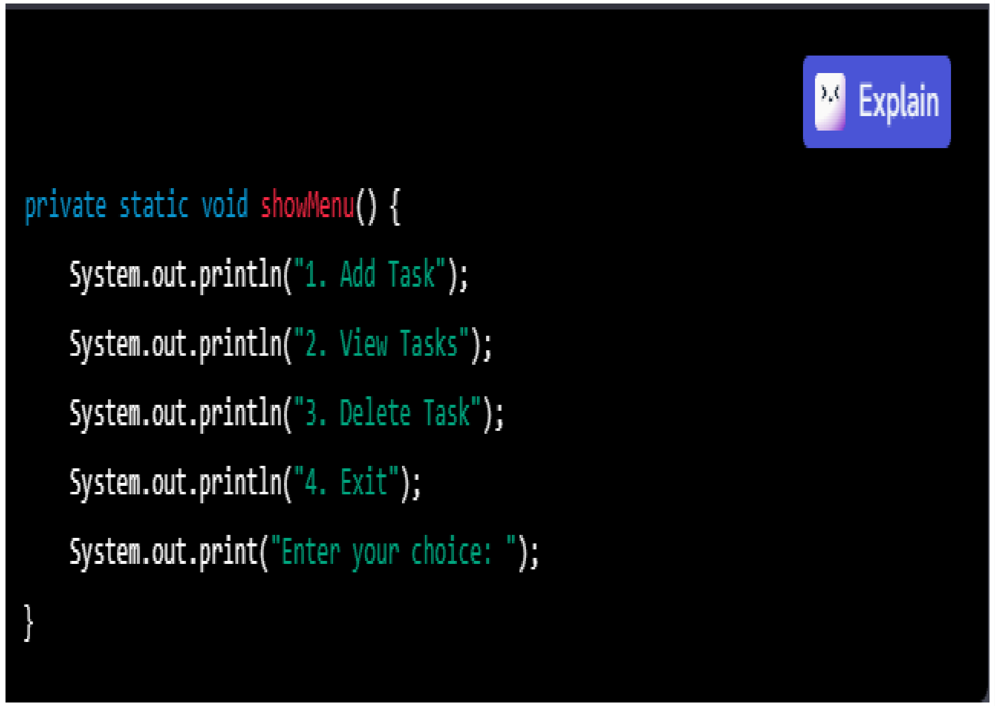
4. Implement Method

- **Main method :**



```java
public static void main(String[] args) {
    int choice;
    do {
        showMenu();
        choice = scanner.nextInt();
        switch (choice) {
            case 1: addTask(); break;
            case 2: viewTasks(); break;
            case 3: deleteTask(); break;
        }
    } while (choice != 4);
}
```

- **Show menu :**



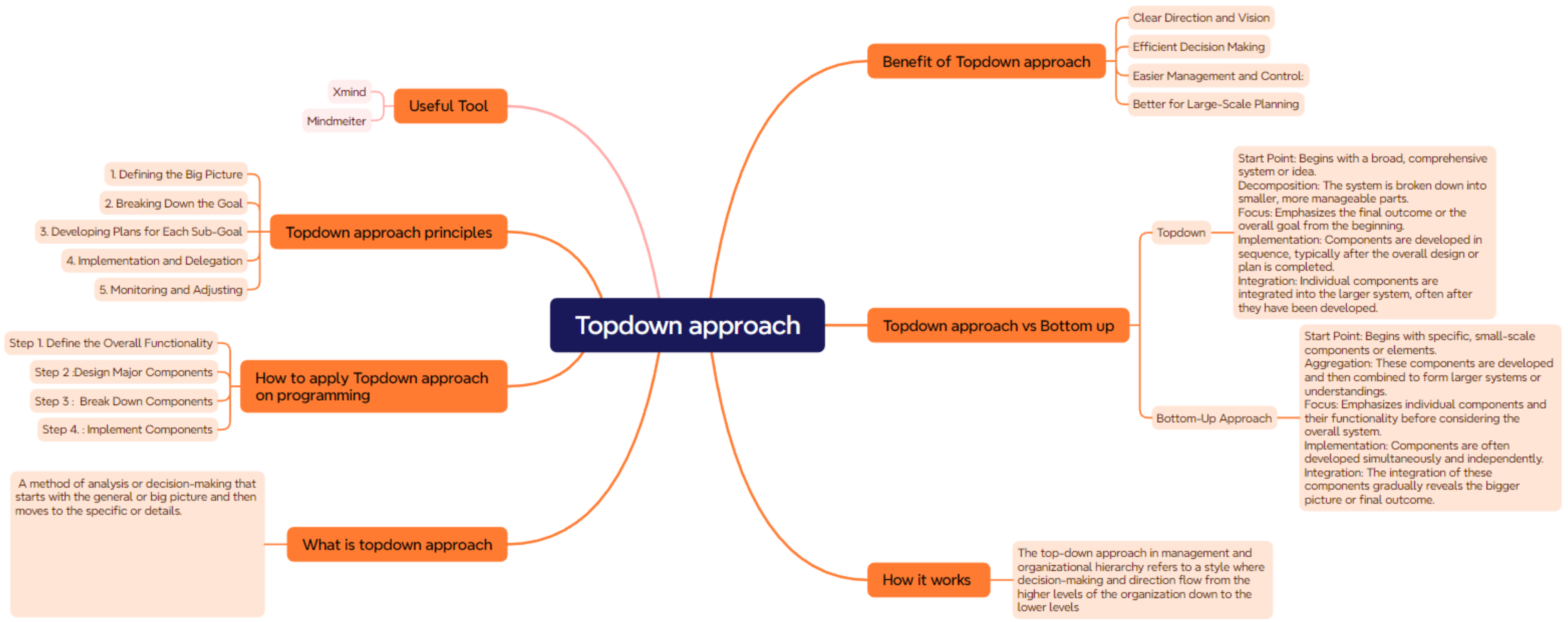
```java
private static void showMenu() {
    System.out.println("1. Add Task");
    System.out.println("2. View Tasks");
    System.out.println("3. Delete Task");
    System.out.println("4. Exit");
    System.out.print("Enter your choice: ");
}
```

Useful tools for Topdown approach

XMind

mind meister

Summary by mindmap



- Topdown approach
  - Useful Tool
    - Xmind
    - Mindmeiter
  - Topdown approach principles
    - 1. Defining the Big Picture
    - 2. Breaking Down the Goal
    - 3. Developing Plans for Each Sub-Goal
    - 4. Implementation and Delegation
    - 5. Monitoring and Adjusting
  - How to apply Topdown approach on programming
    - Step 1. Define the Overall Functionality
    - Step 2 :Design Major Components
    - Step 3 : Break Down Components
    - Step 4. : Implement Components
  - What is topdown approach
    - A method of analysis or decision-making that starts with the general or big picture and then moves to the specific or details.
  - Benefit of Topdown approach
    - Clear Direction and Vision
    - Efficient Decision Making
    - Easier Management and Control:
    - Better for Large-Scale Planning
  - Topdown approach vs Bottom up
    - Topdown
      - Start Point: Begins with a broad, comprehensive system or idea.
        Decomposition: The system is broken down into smaller, more manageable parts.
        Focus: Emphasizes the final outcome or the overall goal from the beginning.
        Implementation: Components are developed in sequence, typically after the overall design or plan is completed.
        Integration: Individual components are integrated into the larger system, often after they have been developed.
    - Bottom-Up Approach
      - Start Point: Begins with specific, small-scale components or elements.
        Aggregation: These components are developed and then combined to form larger systems or understandings.
        Focus: Emphasizes individual components and their functionality before considering the overall system.
        Implementation: Components are often developed simultaneously and independently.
        Integration: The integration of these components gradually reveals the bigger picture or final outcome.
  - How it works
    - The top-down approach in management and organizational hierarchy refers to a style where decision-making and direction flow from the higher levels of the organization down to the lower levels

THANKS

Any questions for my presentation & topdown approach