Object Oriented Analysis and Design with Java

UE19CS353

Prof. Sindhu R Pai
Department of Computer Science and Engineering

Acknowledgements: Significant portions of the information in the slide sets presented through the course in the class, are extracted from the prescribed text books, information from the Internet. Since these are only intended for presentation for teaching within PESU, there was no explicit permission solicited. We would like to sincerely thank and acknowledge that the credit/rights remain with the original authors/creators only

UE19CS353: Object Oriented Analysis and Design with Java

Architectural patterns

Prof. Sindhu R Pai

Department of Computer Science and Engineering

Architectural Patterns - Agenda

- Introduction – Architectural patterns
- Model view Controller
- How MVC Works?
- MVC Architecture in JAVA

Introduction to Architectural patterns

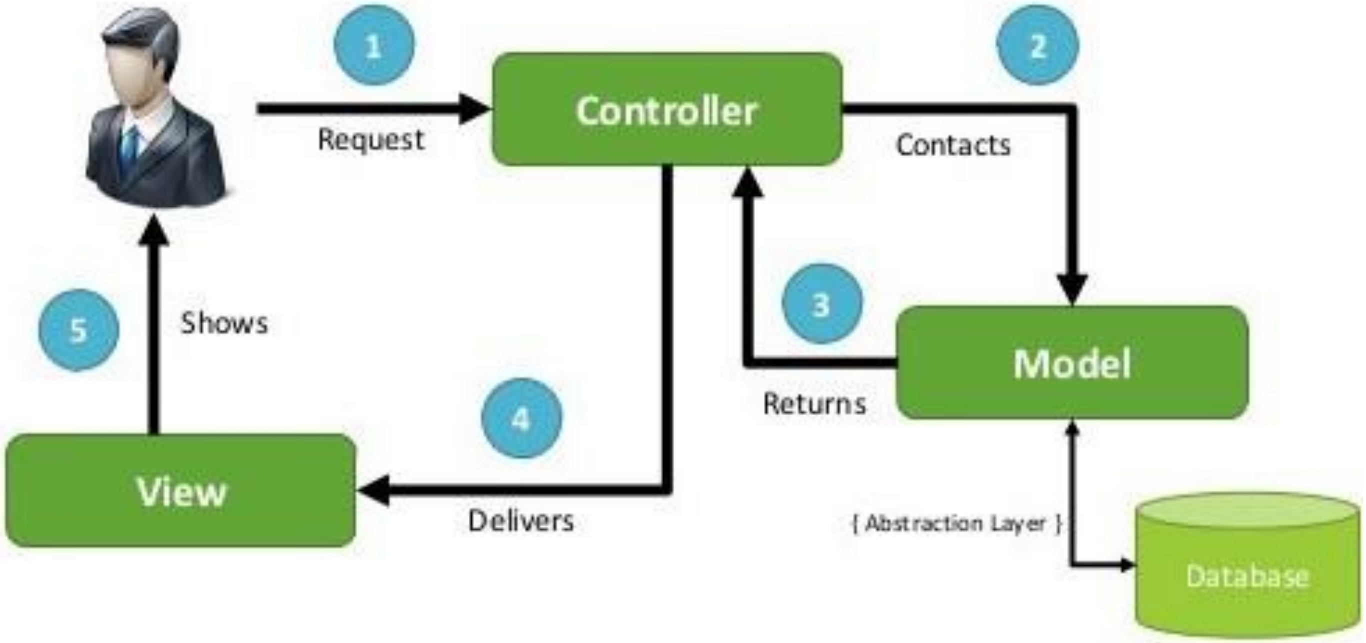
- Help define the basic characteristics and behavior of an application
- Some architecture patterns naturally lend themselves toward highly scalable applications, whereas other architecture patterns naturally lend themselves toward applications that are highly agile.
- Knowing the characteristics, strengths, and weaknesses of each architecture pattern is necessary in order to choose the one that meets the specific business needs and goals.
- Address various issues in software engineering, such as **computer hardware performance limitations, high availability and minimization of a business risk**
- Layered, Event driven, Microkernel, microservices, space based and MVC, etc..

Model View Controller

- Architectural pattern in Software Engineering.
- A way of designing and building applications and separates the application logic from presentation
- Division of application into three main logical components: **model**, **view**, and **controller.**
- A design pattern for computer software considered to distinguish between **the data model, processing control and the user interface.**
- It neatly separates the **graphical interface displayed to the user** from the **code that manages the user actions.**
- Completely separates the calculations and interface from each other

How MVC Works?

- Whenever the controller receives a request from the user (either directly or via the view), it puts the model to work. And when the model delivers the data requested in the right format, the controller forwards it to the view



Model View Controller architecture in Java

- Well-known design pattern in the web development field.
- It is way to organize our code.
- Consists of **Data model, presentation information and control information.**
  - **Model:** Represents the business layer of application. It is an object to carry the data that can contain the logic to update controller if data is changed.
  - **View:** Represents the presentation layer of application. It is used to visualize the data that the model contains.
  - **Controller:** It works on both the model and view. It is used to manage the flow of application, i.e. data flow in the model object and to update the view whenever data is changed.
- The MVC pattern needs all these components to be separated as **different objects.**

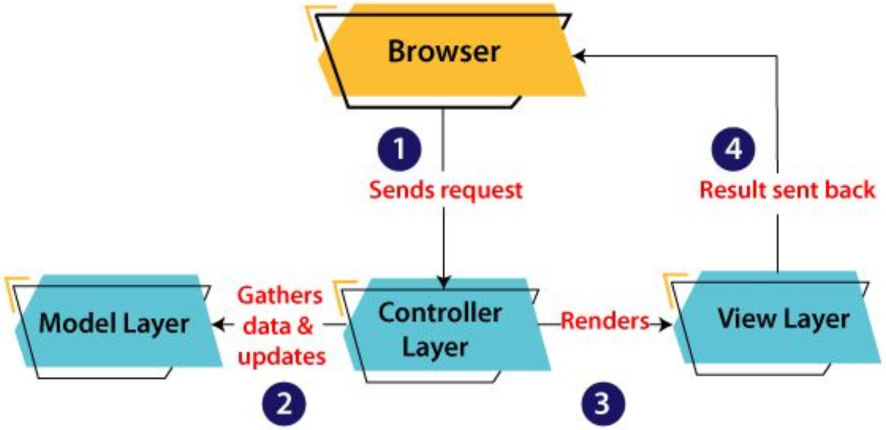
Model View Controller architecture in Java contd..

- In Java Programming,

  The Model contains the simple Java classes,

  The View used to display the data

  The Controller contains the servlets.



- Due to this separation the user requests are processed as follows:
  - A client (browser) sends a request to the controller on the server side, for a page.
  - The controller then calls the model. It gathers the requested data.
  - Then the controller transfers the data retrieved to the view layer.
  - Now the result is sent back to the browser (client) by the view

Implementation of MVC in Java

- **Employee Class, will act as model layer:**

  It represents the business logic for application and also the state of application. The model object fetch and store the model state in the database. Using the model layer, rules are applied to the data that represents the concepts of application.

- **EmployeeView Class, will act as a view layer:**

  View represents the visualization of data received from the model. The view layer consists of output of application or user interface. It sends the requested data to the client, that is fetched from model layer by controller

- **EmployeeContoller Class, will act a controller layer:**

  The controller layer gets the user requests from the view layer and processes them, with the necessary validations. It acts as an interface between Model and View. The requests are then sent to model for data processing. Once they are processed, the data is sent back to the controller and then displayed on the view.

References

- Difference Between Architectural Style, Architectural Patterns and Design Patterns – GeeksforGeeks
- Software Architecture Patterns (oreilly.com)
- MVC Architecture in Java - Javatpoint

THANK YOU

Prof. Sindhu R Pai

Department of Computer Science and Engineering

sindhurpai@pes.edu
+91 8277606459