Hello everyone.

So this week we’re asked to discuss the MVC (Model-View-Controller) of Java. In particular, we’re asked to discuss the design pattern behind it.

Let’s start with what the MVC really is. The MVC, or Model-View-Controller is a design pattern used by software engineers to separate their applications concerns. The Model represents a Java object, the View represents a visualization of the Model data, and the Controller is the brain that controls the flow of data between the model and the view. More specifically, when ever the Model object receives new data, the View is updated and that’s where the controller comes in. The controller keeps those to classes or actions separate in the sense that they do not know about each other. They only know they have to preform some task when they get instructions to do so (tutorialspoint, 2021).

Firstly, let’s have a close look at the Model, what is it exactly. The model is the place where all the hardcore calculations or operations take place. What I mean by that is, if you want to perform a calculation of 1 + 2, the method that will perform that would be coded in the Model. Basically, all the raw operations you want to perform on data should be coded in the model. Furthermore, lets look at the View. The View is a place mainly for the user. What I mean by that is everything you want your user to see should be coded in the View class. It’s the place where you can be creative in the sense of what your app looks like and what they user will be asked to to do etc. In a swing application this will be where most of your swing components will be constructed. Moreover, it should also have a method that will connect it to the controller in the form of action listeners etc. Lastly, the Controller, is place where we connect the Model class and the View class. The controller is responsible for the magic between the Model and the view, therefore, you will most probably have a actionPerformed in there if it’s a swing application. The actionPerformed method is the method that will be responsible for all the operations that takes place between the Model and the View.

In conclusion, the Model is used for operation on data, the View is used for operations on the users interface (what they will see and what they will be asked to do), and the Controller is the brain that controls the flow of data between the model and the view. The MVC pattern helps us have a nice structure in our application which in turn makes them more readable and maintainable. I am pretty sure you can apply this pattern to web development or even embedded and systems development.

Words 471

**References**

tutorialspoint. (2021). *tutorials simply easy learning.* https://www.tutorialspoint.com/design\_pattern/mvc\_pattern.htm#:~:text=MVC%20Pattern%20stands%20for%20Model,the%20data%20that%20model%20contains.