

Java Course

Lecture 8 - Enums, MVC



IT Learning &
Outsourcing Center

www.pragmatic.bg



Enums

- Used when have finite enumerable constants
- For example: days of week (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday), directions (Left, Right, Up, Down), etc.
- Better than other representations (like int, String, etc.) when have finite enumerable value because each constant in the enum has same human readable meaning/name



Enums

```
public enum Day {  
    MONDAY,  
    TUESDAY,  
    WEDNESDAY,  
    THURSDAY,  
    FRIDAY,  
    SATURDAY,  
    SUNDAY  
}
```

- Defined the same way as classes but with keyword *enum* instead of *class*
- Define a new type
- Use *BIG_LETTERS* when defining the different constants (with *_* between the words)



Enums

```
public enum Day {  
    MONDAY ("Monday", 1),  
    TUESDAY ("Tuesday", 2);  
  
    private String name;  
    private int order;  
    Day(String name, int order) { this.name = name; this.order = order; }  
    String getName() { return name; }  
    int getOrder() { return order; }  
}
```

- Can have constructor and methods
- Constructor and methods appear bellow the constants



Enums

- `values()` returns all values as array []

```
for(Day day : Day.values()) {  
    System.out.println(day.getText() + " has order " + d.getOrder());  
}
```

- Useful links: <https://docs.oracle.com/javase/tutorial/java/javaOO/enum.html>



- MVC stands for (Model-View-Controller)
- MVC is a software architectural pattern for implementing user interfaces. That divides a given software application into three interconnected parts (Wikipedia)

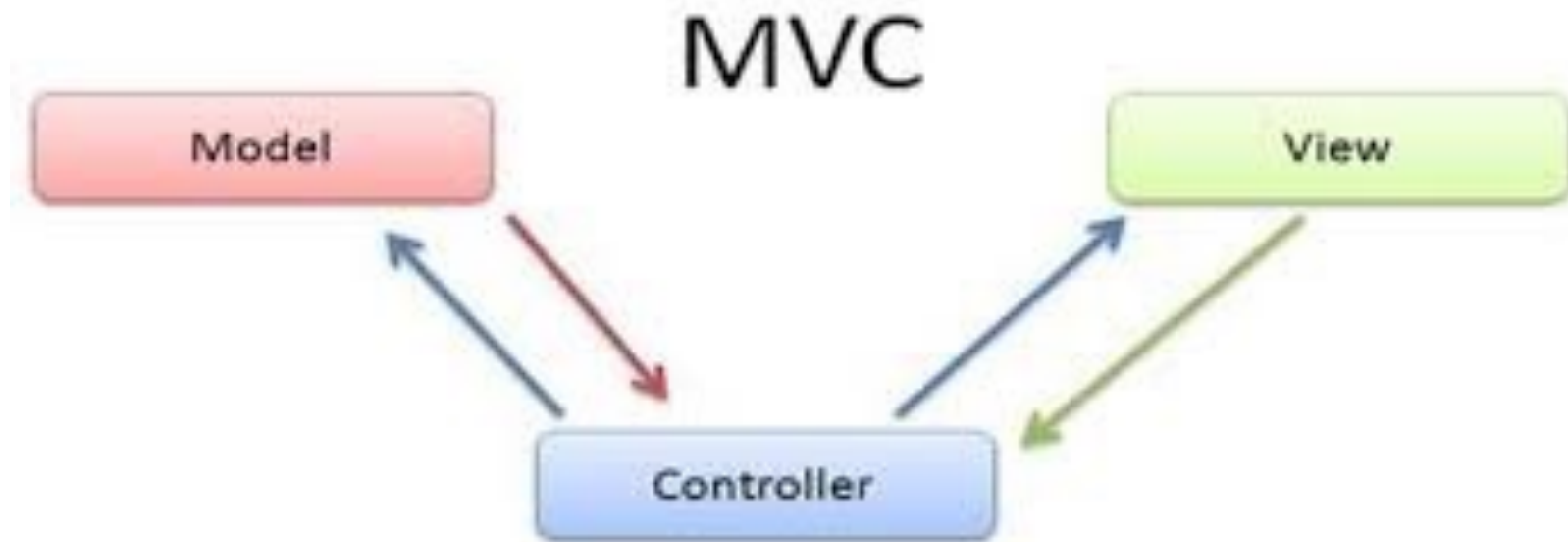


- Model - directly manages the data, logic and rules of the application. Captures the behaviour of the application
- View - any output representation of information, such as a chart or a diagram
- Controller - accepts input and converts it to commands for the model or view. Middleman that “understands” both model and views and controls them.

MVC



- Modern MVC does not allow communication between the Model and the View. Some architectures however allow this interaction





■ Useful links

- <https://www.youtube.com/watch?v=dTVVa2gfht8>
- http://www.tutorialspoint.com/design_pattern/mvc_pattern.htm
- <http://www.oracle.com/technetwork/articles/javase/index-142890.html>