

1.

(b) What is the key difference between software design and software architecture?

2.00

Answer:

The software designer role would be responsible for outlining a software solution to a specific problem by designing the details of individual components and their responsibilities.

A software architect role would be responsible for looking at the entire system and choosing appropriate frameworks, data storage, solutions and determining how components interact with each other.

That brings us to the primary difference between software design and software architecture.

Software design looks at the lower-level aspects of a system.

Software architecture looks at the higher-level aspects of a system.

1 c)

CRC Card Basic Example

Below is a CRC card collection example. Click the image and download the original template for any uses.

Class	Sales
Responsibility	Collaboration
<ul style="list-style-type: none">• Knowledge• Behaviour• Operation• Promotion...	<ul style="list-style-type: none">• Partner• Clients...

Class	Transaction
Responsibility	Collaboration
<ul style="list-style-type: none">• Money Transfer• Auditing...	<ul style="list-style-type: none">• Card reader• Clients...

Class	Order
Responsibility	Collaboration
<ul style="list-style-type: none">• Price• Stock• Valid Payment...	<ul style="list-style-type: none">• Customers• Order line...

Class	Delivery
Responsibility	Collaboration
<ul style="list-style-type: none">• Item identity• Check Receiver• Order No.• Total Qty...	<ul style="list-style-type: none">• Partner• Clients...

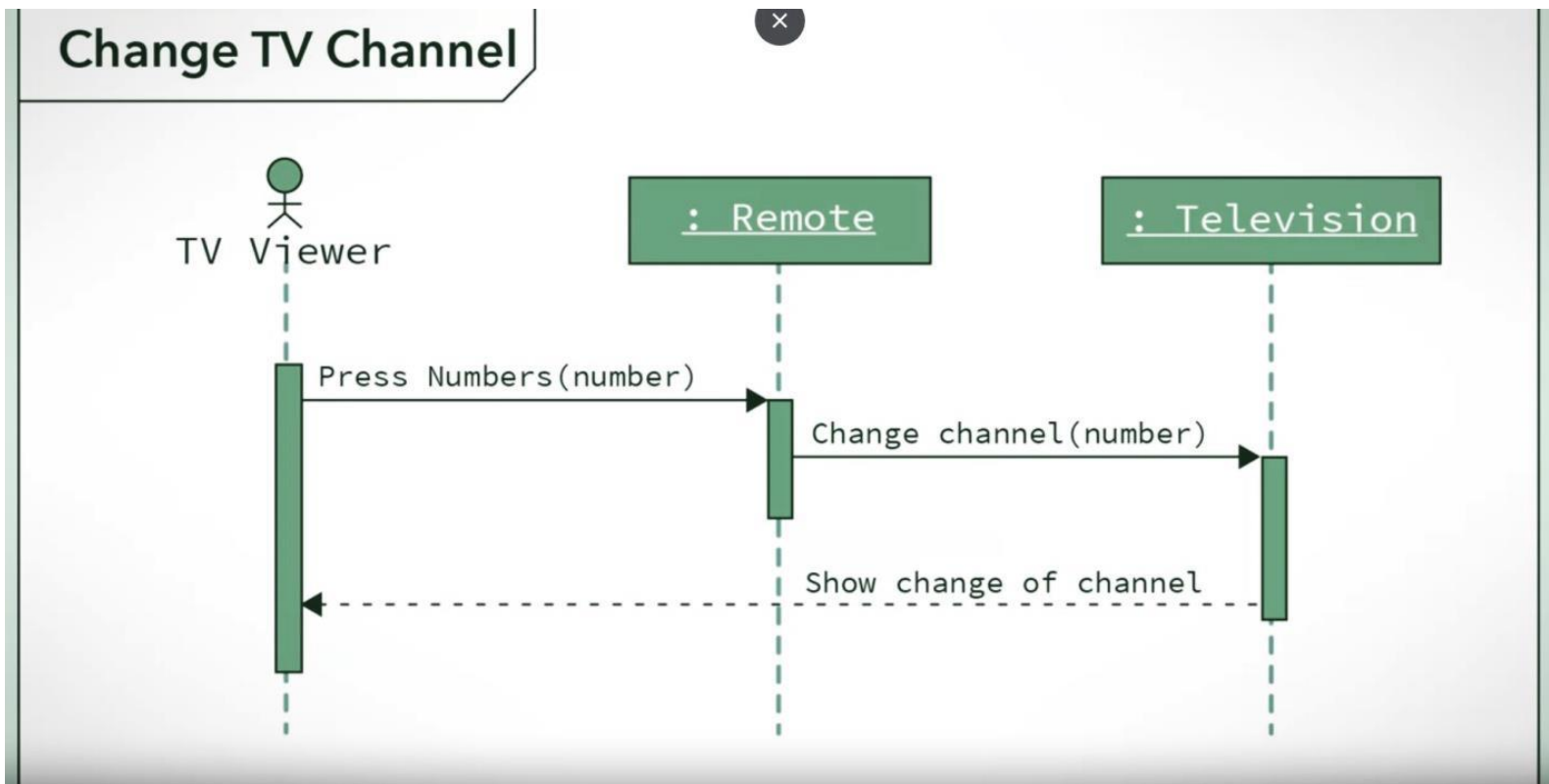
2.

(b) Consider an activity of “**changing TV channel using remote**”.

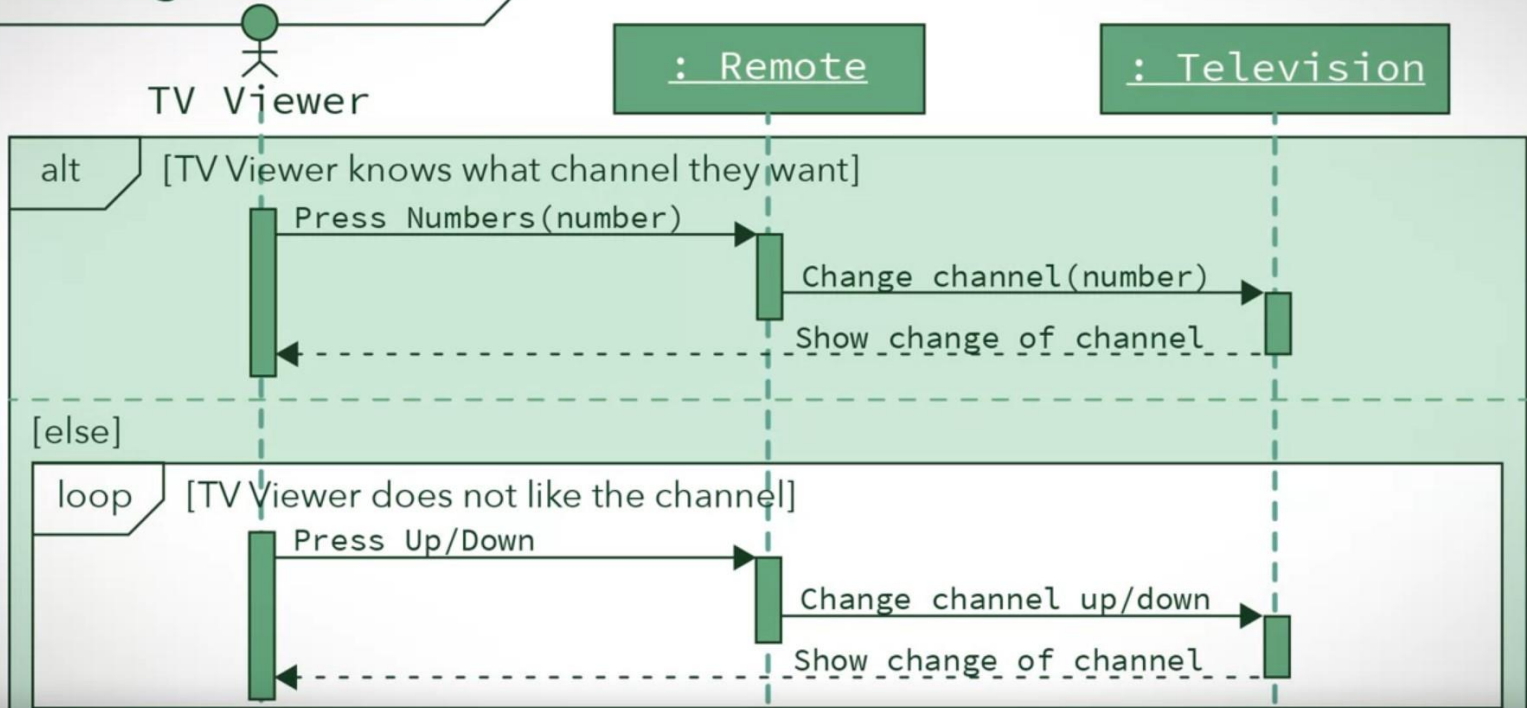
3.75

- When a viewer presses a number on the tv remote, the television changes the channel that the tv viewer can see on the tv screen.
- The viewer can also press the up/down button on the remote to change the channel

Use the above description to draw a UML sequence diagram to model the interaction between the user and TV through the remote.



Change TV Channel



4

- (a) What are “software design patterns”, and why should you use them? 3.50
- (b) Consider the java class below. Modify the class to convert it to a singleton class. 2-11
- Also, show how you can call the *executeSql* method of this singleton class using a demo program.

```

class Database {
    private String dbName, user, password;

    public Database(String dbName, String user, String password) {
        this.dbName = dbName;
        this.user = user;
        this.password = password;
    }

    public void executeSql(String sql){
        System.out.println("Executed query: "+sql);
    }
}
  
```

3.00

Answer:

```
public class Database {  
  
    private String dbName, user, password;  
    private static Database instance;  
  
    private Database(String dbName, String user, String password) {  
        this.dbName = dbName;  
        this.user = user;  
        this.password = password;  
    }  
  
    public static Database getInstance(String dbName, String user, String password) {  
        if (instance == null) {  
            instance = new Database(dbName, user, password);  
        }  
        return instance;  
    }  
  
    public void executeSql(String sql) {  
        System.out.println("Executed query: " + sql);  
    }  
  
    public static void main(String[] args) {  
        // Create an instance of the singleton Database  
        Database database = Database.getInstance("mydb", "myuser", "mypassword");  
        // Call the executeSql method  
        database.executeSql("SELECT * FROM mytable");  
    }  
}
```